



UNIVERSIDAD TÉCNICA DE AMBATO



**FACULTAD DE CIENCIAS E INGENIERÍA EN
ALIMENTOS Y BIOTECNOLOGÍA**

CARRERA DE BIOTECNOLOGÍA

Tema: Evaluación de un modelo de aprendizaje automático basado en redes neuronales para el diagnóstico temprano de la enfermedad de Alzheimer.

Informe Final del Trabajo de Titulación, Opción Proyecto de Investigación, previo la obtención del Título de Ingeniera Biotecnóloga, otorgado por la Universidad Técnica de Ambato, a través de la Facultad de Ciencia e Ingeniería en Alimentos y Biotecnología

Autor: Lilián Catalina Sarabia Ortiz

Tutor: Mg. Cristian Fernando Galarza Galarza

Ambato – Ecuador

Febrero – 2025

APROBACIÓN DEL TUTOR

Mg. Cristian Fernando Galarza Galarza

CERTIFICA:

Que el presente Informe Final del Trabajo de Titulación ha sido prolijamente revisado. Por lo tanto, autorizo la presentación de este Informe Final del Trabajo de Titulación, Opción Proyecto de Investigación, el mismo que responde a las normas establecidas en el reglamento de Títulos y Grados de la Facultad de Ciencia e Ingeniería en Alimentos y Biotecnología.

Ambato, 16 de Enero de 2025.



Firmado electrónicamente por:
**CRISTIAN FERNANDO
GALARZA GALARZA**

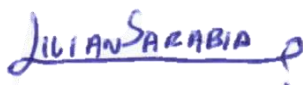
Mg. Cristian Fernando Galarza Galarza

1803160272

TUTOR

AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Lilián Catalina Sarabia Ortiz, manifiesto que los resultados obtenidos en el presente Informe Final del Trabajo de Titulación, Opción Proyecto de Investigación, previo a la obtención del título de Ingeniera Biotecnóloga, son absolutamente originales, auténticos y personales; a excepción de las citas bibliográficas.



Lilián Catalina Sarabia Ortiz

1850076918

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Informe Final del Trabajo de Titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los Derechos en línea patrimoniales de mi Informe Final del Trabajo de Titulación, con fines de difusión pública, además apruebo la reproducción de este, dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando los derechos de autor.



Lilián Catalina Sarabia Ortiz

1850076918

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

Los suscritos Profesores Calificadores, aprueban el presente Informe Final del Trabajo de Titulación, Opción Proyecto de Investigación, el mismo que ha sido elaborado de conformidad con las disposiciones emitidas por la Facultad de Ciencia e Ingeniería en Alimentos y Biotecnología de la Universidad Técnica de Ambato.

Para constancia firman:



Firmado electrónicamente por:
**LORENA DE LOS
ANGELES NUNEZ
VILLACIS**

Presidente del Tribunal



Firmado electrónicamente por:
**ALEX FABIAN
VALENCIA SILVA**

Mg. Alex Fabián Valencia Silva

1803121084



Firmado electrónicamente por:
**LILIANA PAULINA
LALALEO CORDOVA**

Dra. Liliana Paulina Lalaleo Córdova

1803601853

Ambato, 03 de Febrero de 2025

DEDICATORIA

“A mis padres, por luchar cada día de su vida para sacar adelante a sus hijas con amor, esfuerzo y dedicación a lo que hacen, por estar siempre alentando y apoyando mis decisiones y sueños.

A mi hermana, la cual ha estado siempre a mi lado apoyándome en todo y enseñándome la incondicionalidad.

A mi familia, por apoyarme en este largo camino y se han sentido orgullosos por lo que soy.

A Paulo, que desde un inicio de esta etapa estuvo conmigo apoyándome, escuchándome, alentándome, enseñándome y queriéndome a pesar de todos los obstáculos de la vida.

A mis amigos, por siempre ayudarme a seguir y compartir conmigo momentos inolvidables.

A mis gatos, Maty el cual siempre me acompañó en los desvelos y que desde el cielo me cuida y Pimienta que sigue conmigo haciéndome compañía todos los días.

A Dios, por darme la fortaleza, conocimiento y capacidad para poder llegar a cumplir este logro y avanzar en la vida a pesar de todos los problemas.

Y a mí, por creer, confiar, amar y hacer todo esto posible.

Con amor y agradecimiento”

Lilián

AGRADECIMIENTO

Quiero agradecer a Dios por darme la fortaleza y guiarme en este camino.

Mis padres Rodrigo y Marianela que siempre han estado conmigo ayudándome a cumplir mis sueños. A mi hermana Emily por ser mi compañera de vida y confidente. Mis abuelitos y tíos quienes han sido un pilar fundamental en mi vida, los cuales me han enseñado muchas cosas con amor y paciencia.

A Paulo agradezco profundamente por quererme incondicionalmente, apoyarme, escucharme y por brindarme un espacio en su vida. Gracias por los viajes, aventuras, amigos, fotos, conciertos, risas y paciencia. TQM <3

A Maty por haber aparecido repentinamente en mi vida y hacerme sentir un amor incondicional, te considero como mi hijo el cual nunca olvidare, gracias por escucharme y acompañarme en esos años, pronto nos volveremos a ver. Pimienta, mi viejita que siempre me ha calentado las piernas, espero tenerte muchos años junto a mí.

A mis amigos, Alex, Jae, Dami, Daya, Joel, Pame, Monse, Joel, Ángel y Patricio. Gracias por estar y crear momentos lindos en mi vida, espero que esta amistad siga por más años.

A mi tutor de tesis Cristian, el cual apoyó mi proyecto, me brindó su conocimiento y paciencia para poder terminar este trabajo. A mis maestros de la carrera, los cuales me han enseñado todo lo que sé y así amar la ciencia.

Al destino, el cual hizo que conozca a todas estas personas, las cuales marcaron mi vida en esta linda etapa.

Los amo a todos y gracias por confiar en mí, espero nunca defraudarlos.

ÍNDICE GENERAL DE CONTENIDOS

APROBACIÓN DEL TUTOR.....	ii
AUTORÍA DEL TRABAJO DE TITULACIÓN	iii
DERECHOS DE AUTOR.....	iv
APROBACIÓN DEL TRIBUNAL DE GRADO	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL DE CONTENIDOS	viii
ÍNDICE DE TABLAS	xii
ÍNDICE DE FIGURAS	xiii
ÍNDICE DE ANEXOS.....	xiv
RESUMEN EJECUTIVO	xv
ABSTRACT	xvi
CAPÍTULO I.- MARCO TEÓRICO	1
1.1 Antecedentes Investigativos	1
1.1.1 Introducción al problema de la enfermedad de Alzheimer.....	1
1.1.1.1 Definición y características de la enfermedad de Alzheimer.....	1
1.1.1.2 Diagnóstico de la enfermedad de Alzheimer.....	3
1.1.1.3 Tratamientos de la enfermedad de Alzheimer	4
1.1.2 Inteligencia Artificial y Aprendizaje Automático en Medicina	6
1.1.2.1 Conceptos generales del Aprendizaje Automático (Machine Learning, ML)	6
1.1.2.2 Modelos supervisados y no supervisados.....	6
1.1.3 Redes Neuronales y Aprendizaje Profundo (Deep Learning)	7
1.1.3.1 Conceptos básicos de Redes Neuronales y Aprendizaje Profundo	7
1.1.3.2 Aplicaciones de DL con imágenes médicas	8
1.1.3.3 Arquitecturas preentrenadas en Deep Learning	8

1.1.4	Evaluación y Validación de los Modelos	9
1.1.4.1	Métricas de evaluación de modelos	9
	<i>F1-Score</i>	11
1.1.5	Interpretabilidad de los Modelos con Mapas de Calor.....	13
1.1.5.1	Importancia de la interpretabilidad en modelos de IA.....	13
1.1.5.2	Mapas de calor (Grad-CAM)	14
1.1.6	Base de Datos	15
1.1.6.1	Descripción de la base de datos OASIS-1	15
1.1.6.2	Descripción de la base de datos ADNI.....	15
1.2	Objetivos	17
1.2.1	Objetivo General	17
1.2.2	Objetivos Específicos	17
CAPÍTULO II.- METODOLOGÍA.....		18
2.1	Materiales	18
2.2	Métodos.....	19
2.2.1	Construcción de un conjunto de datos de prueba a partir de información obtenida de la base de datos Oasis y ADNI de pacientes con y sin enfermedad del Alzheimer	19
2.2.1.1	Descarga y preparación de datos	19
2.2.1.2	Selección y Filtrado de Imágenes	19
2.2.1.3	Etiquetado y Balanceo de Datos	20
2.2.2	Creación de las funciones necesarias para preparar el modelo de Deep Learning	21
2.2.2.1	Importación de las librerías principales.....	21
2.2.2.2	División del conjunto de datos.....	23
2.2.2.3	Preprocesamiento y Generación de Datos.....	24
2.2.2.4	Configuración de la visualización de datos	25
2.2.2.5	Configuración de Callbacks para el Control del Entrenamiento.....	26
2.2.2.6	Creación de imágenes del historial y entrenamiento del modelo.....	27

2.2.2.7	Creación de la matriz de confusión	28
2.2.3	Estructura del modelo CNN	29
2.2.3.1	Preparación de los datos para el modelo de aprendizaje profundo	29
2.2.3.2	Definición de la arquitectura del modelo de red neuronal convucional (CNN).....	29
2.2.3.3	Entrenamiento del modelo CNN.....	31
2.2.4	Construcción del modelo EfficientNetB7 y ResNet50.....	33
2.2.4.1	Carga del modelo base	33
2.2.4.2	Congelación de capas.....	34
2.2.4.3	Ajuste fino	34
2.2.4.4	Adición de capas personalizadas.....	34
2.2.4.5	Compilación y entrenamiento	35
2.2.4.6	Descongelar algunas capas del modelo base	35
2.2.4.7	Recopilación del modelo.....	35
2.2.4.8	Entrenamiento con capas descongeladas.....	35
2.2.5	Evaluación de la precisión del modelo de aprendizaje automático sobre datos de pacientes con la enfermedad de Alzheimer.	36
2.2.5.1	Realizar las predicciones.....	36
2.2.5.2	Generación de curva ROC-AUC.....	37
2.2.5.3	Matriz de confusión y reporte de clasificación.....	37
2.2.5.4	Gráfico Grad-CAM.....	38
2.2.5.5	Prueba con datos nuevos de OASIS y ADNI	39
CAPÍTULO III.- RESULTADOS Y DISCUSIÓN		40
3.1	Análisis y discusión de los resultados.....	40
3.1.1	Entrenamiento de las arquitecturas EfficientNetB7 y ResNet50.....	40
3.1.2	Evaluación de rendimiento e interpretabilidad de las arquitecturas EfficientNet B7 y ResNet50.....	44
3.1.3	Evaluación de rendimiento de las arquitecturas EfficientNet B7 y ResNet50 frente a nuevos datos generados de ADNI y OASIS	52

CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES	60
4.1 Conclusiones	60
4.2 Recomendaciones.....	62
MATERIALES DE REFERENCIA	63
Referencias Bibliográficas.....	63
ANEXOS.....	69

ÍNDICE DE TABLAS

Tabla 1 Matriz de Confusión.....	10
Tabla 2. Recursos Bioinformáticos.....	18
Tabla 3. Equipos.....	18
Tabla 4. Desempeño de las arquitecturas basada en parámetros precisión, recall, f1-score y especificidad.	48
Tabla 5. Aplicación de los modelos EfficientNet B7 y ResNet50 en datos nuevos ADNI.....	52
Tabla 6. Aplicación de los modelos EfficientNet B7 y ResNet50 en datos nuevos OASIS	55

ÍNDICE DE FIGURAS

Figura 1: Comparación entre un cerebro saludable y un cerebro con Alzheimer grave	1
Figura 2. Representación de la curva ROC-AUC	13
Figura 3. Ejemplo de Imágenes de Resonancia Magnética (RM) de las Bases de Datos OASIS-1 y ADNI	21
Figura 4. Resumen de la arquitectura del modelo	31
Figura 5. Curvas de Pérdida y Precisión del Modelo Durante el Entrenamiento	33
Figura 6. Gráfica de pérdida y precisión de la arquitectura EfficientNetB7	41
Figura 7. Gráfica de pérdida y precisión de la arquitectura ResNet50	42
Figura 8. Curvas ROC-AUC comparativas de las arquitecturas EfficientNetB7 y ResNet50 para clasificación multiclase.	45
Figura 9. Matrices de confusión en la clasificación multiclase del alzheimer	46
Figura 10. Comparación de los mapas generados por las dos arquitecturas EfficientNet y ResNet50	50

ÍNDICE DE ANEXOS

Anexo 1. Script para preprocesar las imágenes de la base de datos OASIS	69
Anexo 2. Script para preprocesar las imágenes de la base de datos ADNI.....	80
Anexo 3. Enlace hacia el script con sus resultados de la arquitectura ResNet50	85
Anexo 4. Enlace hacia el script con sus resultados de la arquitectura EfficientNet B7	85

RESUMEN EJECUTIVO

La detección temprana del Alzheimer representa un desafío crítico, la intervención oportuna puede retrasar significativamente su progresión. Este estudio aborda dicha problemática mediante la evaluación de modelos de aprendizaje profundo basados en redes neuronales para clasificar imágenes de resonancia magnética en distintas etapas de la enfermedad, utilizando datos provenientes de las bases OASIS y ADNI. La relevancia del trabajo radica en la necesidad de desarrollar herramientas precisas y automatizadas que faciliten el diagnóstico temprano, particularmente en fases iniciales como el deterioro cognitivo leve. La metodología consistió en construir un conjunto de datos a partir de imágenes preprocesadas e implementar las arquitecturas EfficientNet B7 y ResNet50. Estas fueron entrenadas empleando técnicas, como el aumento de datos y validadas en varios escenarios, incluidos datos controlados, imágenes modificadas y datos no vistos previamente. Se analizaron métricas clave como precisión, sensibilidad, especificidad, F1-score y curva ROC-AUC. Los resultados mostraron que EfficientNet B7 superó a ResNet50 en precisión, sensibilidad y especificidad, destacándose especialmente en la clasificación de etapas tempranas del Alzheimer. Además, EfficientNet B7 mostró una mayor capacidad de generalización, alcanzando un alto nivel de precisión tanto con datos preprocesados como con imágenes nuevas. En contraste, ResNet50 presentó limitaciones al trabajar con datos heterogéneos. Este estudio subraya la importancia de emplear arquitecturas modernas para abordar problemas complejos, como la detección temprana del Alzheimer. Sin embargo, pone de manifiesto las limitaciones de utilizar bases de datos controladas, como OASIS y ADNI, dado que estas pueden restringir la aplicabilidad de los modelos en escenarios clínicos reales.

Palabras clave: Alzheimer, Deep Learning, EfficientNet, ResNet, OASIS

ABSTRACT

Early detection of Alzheimer's disease represents a critical challenge; timely intervention can significantly delay its progression. This study addresses this problem by evaluating deep learning models based on neural networks to classify magnetic resonance images at different stages of the disease, using data from the OASIS and ADNI databases. The relevance of the work lies in the need to develop accurate and automated tools to facilitate early diagnosis, particularly in early stages such as mild cognitive impairment. The methodology consisted of building a dataset from preprocessed images and implementing the EfficientNet B7 and ResNet50 architectures. These were trained using techniques such as data augmentation and validated in various scenarios, including controlled data, modified images and previously unseen data. Key metrics such as accuracy, sensitivity, specificity, F1-score and ROC-AUC curve were analyzed. The results showed that EfficientNet B7 outperformed ResNet50 in accuracy, sensitivity and specificity, especially excelling in the classification of early stages of Alzheimer's disease. In addition, EfficientNet B7 showed greater generalization capability, achieving a high level of accuracy with both preprocessed data and new images. In contrast, ResNet50 presented limitations when working with heterogeneous data. This study highlights the importance of employing modern architectures to address complex problems, such as the early detection of Alzheimer's disease. However, it highlights the limitations of using controlled databases, such as OASIS and ADNI, as these may restrict the applicability of the models in real clinical scenarios.

Keywords: Alzheimer, Deep Learning, EfficientNet, ResNet, OASIS

CAPÍTULO I.- MARCO TEÓRICO

1.1 Antecedentes Investigativos

1.1.1 Introducción al problema de la enfermedad de Alzheimer

1.1.1.1 Definición y características de la enfermedad de Alzheimer

La enfermedad del Alzheimer (EA) es el responsable del 60-80% de demencia más común en las personas, el cual llega a afectar principalmente en adultos mayores de 65 años, pero últimamente se ha llegado a diagnosticar casos en personas jóvenes (**Alzheimer's Association, 2022**).

La sobreproducción de las proteínas β -amiloide y tau son las responsables del desarrollo de esta enfermedad, ya que provoca la formación de placas amiloides y ovillos neurofibrilares (ovillos de tau) en el cerebro, lo que contribuye a la pérdida de la función neuronal.

Además, la desconexión entre neuronas causa el deterioro de las habilidades cognitivas y conductuales, afectando capacidades como pensar, razonar, hablar, comer, caminar, entre otras (**National Institute on Aging, 2023**).

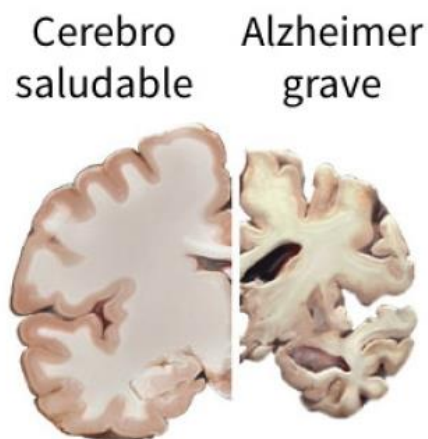


Figura 1: Comparación entre un cerebro saludable y un cerebro con Alzheimer grave

Nota: La imagen ilustra los efectos progresivos de la enfermedad de Alzheimer sobre el tejido cerebral. A la izquierda, un cerebro sano muestra una estructura completa y funcional, mientras que, a la derecha, el cerebro con Alzheimer avanzado presenta atrofia severa, pérdida de masa cerebral y ampliación de los

ventrículos, características típicas de esta enfermedad neurodegenerativa. Estas alteraciones estructurales están asociadas con el deterioro cognitivo y la pérdida de funciones cerebrales en los pacientes. Obtenido de **(National Institute on Aging, 2023)**.

En varias investigaciones se ha llegado a la conclusión que, en la mayoría de los casos, la EA puede llegar a una etapa de demencia después de unos 15 a 25 años desde la etapa preclínica, sin presentar ningún síntoma significativo hasta la etapa terminal. Algunas personas presentan deterioro cognitivo leve, caracterizado por dificultades de memoria mayores a lo esperado para su edad, pero esto no afecta su vida, también se puede incluir problemas de movimiento y olfato, aumentando así el riesgo de desarrollar Alzheimer, aunque no todos los afectados lo padecen. Dichos síntomas varían entre individuos, las dificultades cognitivas no relacionadas con la memoria, como problemas para encontrar palabras, orientación espacial, y razonamiento o juicio, pueden ser las primeras señales de la enfermedad **(Scheltens et al., 2021; National Institute on Aging, 2023)**.

Breijyeh & Karaman, (2020); National Institute on Aging, (2023) mencionan que existen 4 fases de la EA, primero se tiene a la etapa **preclínica o presintomática**, la cual puede durar varios años y se caracteriza por cambios tempranos en la corteza y el hipocampo, junto con una pérdida leve de memoria, cabe recalcar que en esta fase no existe deterioro en las actividades diarias ni signos clínicos evidentes de la enfermedad. En la segunda etapa **leve o temprana**, los síntomas comienzan a manifestarse, como dificultades en la vida diaria relacionadas con pérdida de memoria y concentración, desorientación, cambios en el estado de ánimo y posible desarrollo de depresión.

La tercera etapa es **moderada**, aquí la enfermedad afecta más áreas de la corteza cerebral, causando problemas graves de memoria, dificultad para reconocer a familiares y amigos, pérdida de control de impulsos y problemas para leer, escribir y hablar.

Por último, en la etapa **grave o tardía** la enfermedad se extiende por toda la corteza, con una acumulación masiva de placas y ovillos, lo que provoca un deterioro cognitivo y funcional severo. Los pacientes pueden perder completamente la capacidad de reconocer a sus seres queridos, quedar postrados en cama y tener dificultades para tragar y controlar esfínteres, lo que eventualmente puede llevar a la muerte.

Uno de los factores de riesgo de la enfermedad más importante es la edad, ser mayor a 65 años, pero esto no siempre puede ser un indicio de padecimiento; otro factor, es contar con un alelo *APOE ε4*, el cual sigue siendo investigado si este gen es hereditario, debido a que, estudios en gemelos han demostrado que entre

el 60-80% de la EA se da por herencia, este es un factor clave en investigaciones porque se ha descubierto genes protectores, resultados de una rara mutación que disminuye 2 veces el padecimiento de demencia; por otro lado, el sexo también tiene relación con la enfermedad, se ha demostrado que las mujeres tienen mayor probabilidad de poseer más carga de la proteína tau; de igual forma se dice que la mala alimentación y estilo de vida aumenta el deterioro cognitivo en las personas, al igual que lesiones en la cabeza y enfermedades cardiovasculares (Scheltens et al., 2021; Breijyeh & Karaman, 2020).

Cabe destacar que Khan et al., (2020) menciona otros mecanismo de la EA, los cuales son: degeneración neurofibrilar, disfunción sináptica y desequilibrio de neurotransmisores, neuroinflamación, alteración del microbioma intestinal, mutaciones genéticas, estrés oxidativo y autofagia. Es por ello por lo que, a pesar de los diferentes indicios de la enfermedad, no se sabe con exactitud que desata la patogénesis en cada individuo.

1.1.1.2 Diagnóstico de la enfermedad de Alzheimer

El diagnóstico de la EA se basa en la tomografía por emisión de positrones (PET) y el análisis de proteínas en el líquido cefalorraquídeo (LCR). La fosforilación de la P-tau18 es un biomarcador clave, este se deposita en el LCR y su resultado se muestra en la sangre. Las exploraciones PET, que emplean ligandos como florbetapir, florbetaben y flutemetamol muestran alta especificidad del 100% y 96% de sensibilidad, pero dicha técnica no es tan aplicada por su alto coste. Por otro lado, el análisis de LCR, que evalúa P-tau, A β 42 y tau total, es más económico pero invasivo porque se realiza por medio de una punción lumbar, a pesar de lo tardío del proceso y las pocas instalaciones existentes, dicha técnica muestra una precisión del 85-90% (Khan et al., 2020). Además, se ha demostrado que los cambios patológicos pueden detectarse hasta dos décadas antes de la aparición de los síntomas clínicos, subrayando la importancia de la detección temprana para un manejo eficaz de la enfermedad (Bateman et al., 2012).

En la recopilación de información de Khan et al., (2020) menciona que existen diferentes biomarcadores para la detección temprana del Alzheimer según el método de análisis, los marcadores principales y ya establecidos son los A β 42 y A β 40, estos se pueden detectar en la sangre, orina y LCR, según el método u experimento que se presente. En la fase lipídica del cerebro, los niveles bajos de esfingolípidos presentes en la sangre se muestran afectado en la EA. En la genética se tiene un factor de riesgo debido al alelo ϵ 4 del gen APOE el cual está presente en más del 40% de pacientes con la EA. Por otro lado, el polimorfismo Val66Met del factor neurotrófico derivado del cerebro (BDNF) empeora las condiciones de reducción del

volumen hipocampal en un lapso de 36 meses. De igual forma, en la EA se presenta la reducción de la proteína renal/cerebral KIBRA, la cual está asociada a la memoria de las personas.

El diagnóstico actual de la EA se basa en una combinación de pruebas neuropsicológicas, neuroimagen avanzada y análisis de biomarcadores. Las concentraciones de péptidos A β en proporción A β 1-42 : A β 1-40 y proteínas tau en el LCR son esenciales para confirmar el diagnóstico, pero su obtención es invasiva y no siempre bien tolerada. Métodos no invasivos como la PET con trazadores específicos y la MRI se utiliza para evaluar tanto la estructura como la función cerebral, permitiendo el análisis del volumen hipocampal, la conectividad neuronal y el flujo sanguíneo cerebral. Avances recientes incluyen el uso de nanopartículas, que actúan como agentes de contraste y permiten visualizar placas amiloides con mayor precisión. Nanopartículas de oro y óxido de hierro se han empleado para mejorar las imágenes de MRI, aprovechando sus propiedades para marcar y detectar fibrillas A β . Además, técnicas emergentes han desarrollado sistemas ultrasensibles para detectar oligómeros solubles de A β en el LCR, como códigos de barras biológicos basados en nanopartículas, que ofrecen una detección más temprana y precisa. Estos avances han mejorado la capacidad diagnóstica y abren la puerta a intervenciones más eficaces. Sin embargo, los tratamientos actuales solo modifican la progresión de la enfermedad, ofreciendo un beneficio limitado y subrayando la necesidad de seguir investigando enfoques terapéuticos más efectivos **(Breijyeh & Karaman, 2020; Khan et al., 2020; Tiwari et al., 2019)**.

1.1.1.3 Tratamientos de la enfermedad de Alzheimer

Tratamientos farmacológicos

Los tratamientos farmacológicos actuales para la EA son principalmente sintomáticos y se enfocan en limitar la progresión de síntomas cognitivos, conductuales y psicológicos. Los fármacos aprobados incluyen inhibidores de la anticolinesterasa en los cuales destacan el donepezilo, galantamina, rivastigmina y antagonistas de los receptores NMDA que es la memantina, administrados oralmente o por vía transdérmica **(Atri, 2019)**. Estos medicamentos aumentan los niveles de acetilcolina y regulan el glutamato para mejorar la transmisión neuronal y retrasar el deterioro cognitivo. Sin embargo, la eficacia es limitada, en parte, por la barrera hematoencefálica (BHE), la progresión avanzada de la EA y la necesidad de dosificación elevada, que puede causar efectos adversos **(Passeri et al., 2022)**.

La administración de estos tratamientos en etapas avanzadas de la EA ha mostrado beneficios temporales, pero no significativos en la reducción de la progresión de la enfermedad. Los ensayos clínicos con enfoques antiamiloides, aunque han logrado disminuir la carga de placas, no han mostrado mejorías significativas en la función cognitiva, probablemente debido al diagnóstico tardío. Esto enfatiza la importancia de un diagnóstico temprano que aproveche el uso de biomarcadores adicionales para identificar la EA en fases iniciales, donde las intervenciones podrían ser más efectivas. Además, existe la necesidad de investigar nuevos objetivos terapéuticos más allá de la formación de placas amiloides para desarrollar tratamientos que modifiquen la enfermedad y no solo sus síntomas (Cummings et al., 2019; Passeri et al., 2022).

Tratamientos no farmacológicos

En el año 2019, **World Health Organization, (2019)** publicó un libro de recomendaciones para reducir el riesgo de deterioro cognitivo y demencia basado en evidencia científica. Este menciona 12 aspectos importantes a considerar para prevenir y retrasar el paso de la enfermedad. Entre dichos puntos se tiene la actividad física, dejar de fumar y el alcohol, buena nutrición, actividad social, control de peso, manejo de la hipertensión, diabetes, depresión y pérdida auditiva. Las intervenciones pueden presentar limitaciones con diferentes grados de certeza, ya que el trabajo realizado no se encuentra controlado a largo plazo y en países de bajos recursos es más difícil encontrar especialistas.

Además, **Scheltens et al., (2021)** nos menciona una serie de estudios que se han realizado a lo largo de los años, donde señala que los fracasos de intervenciones aisladas indican la necesidad de un enfoque preventivo multimodal. El estudio Finandés FINGER demostró que una combinación de dieta saludable, ejercicio, entrenamiento cognitivo y gestión de riesgos puede reducir el deterioro cognitivo en pacientes. Aunque el ensayo Francés MAPT que implementó ácidos grasos omega-3 en la dieta y PreDIVA (Holandés) con farmacología vascular y metabólico, no lograron resultados primarios significativos, pero se presentaron beneficios cognitivos en subgrupos con mayor riesgo de demencia. Un subestudio de MAPT encontró que las intervenciones en el estilo de vida mejoraron la cognición en personas con β amiloide positivo, sugiriendo que los cambios en el estilo de vida pueden ser beneficiosos incluso en quienes ya presentan signos de la enfermedad.

1.1.2 Inteligencia Artificial y Aprendizaje Automático en Medicina

1.1.2.1 Conceptos generales del Aprendizaje Automático (Machine Learning, ML)

El aprendizaje automático (ML) es un conjunto de herramientas basadas en datos utilizadas para apoyar la toma de decisiones en muchos ámbitos científicos, especialmente en la asistencia sanitaria se ha demostrado potencial, debido a que emplean algoritmos que aprenden a predecir un resultado específico, como la incidencia de enfermedades o el futuro pronóstico del paciente **(Silva et al., 2022)**.

Un algoritmo de aprendizaje automático es una de las técnicas intelectuales artificiales para seleccionar el mejor modelo de un conjunto de alternativas para adaptarse a un conjunto de observaciones. Los algoritmos de aprendizaje automático tienen varios méritos, incluyendo la no linealidad, la tolerancia a la falla y la operación en tiempo real, haciéndolos adecuados para aplicaciones complejas **(Chang et al., 2021)**.

1.1.2.2 Modelos supervisados y no supervisados

Los tipos de algoritmos de ML se distinguen según su enfoque de aprendizaje, los principales son: aprendizaje supervisado, aprendizaje no supervisado, aprendizaje por refuerzo, entre otras.

El aprendizaje supervisado (SL) es utilizado para predecir resultados específicos a partir de datos etiquetados, entrenando un modelo para identificar patrones y clasificar nuevos datos; se puede utilizar para el diagnóstico de enfermedades, pronóstico y toma de decisiones clínicas. De igual forma el Aprendizaje no supervisado (UL) es aplicado cuando los datos no están etiquetados, buscando patrones y agrupando datos similares sin etiquetas predefinidas, el cual permite identificar subgrupos de pacientes, descubrir nuevas relaciones entre variables y explorar la estructura de datos complejos. Por otro lado, el aprendizaje por refuerzo (RL) entrena a un agente para aprender la mejor habilidad a partir de la interacción con su entorno, recibiendo recompensas y castigos por sus acciones gracias a la retroalimentación; es útil para optimizar tratamientos médicos personalizados, simular escenarios clínicos y desarrollar sistemas de apoyo a la toma de decisiones **(Woodman & Mangoni, 2023)**.

Los algoritmos de aprendizaje automático pueden dividirse en modelos supervisados en los que los datos estén etiquetados, algoritmos no supervisados en los que el objetivo sea separar un dato no etiquetado en grupos de casos relacionados, y algoritmos semisupervisados, incluyendo datos etiquetados y no etiquetados. Un algoritmo de aprendizaje automático es un procedimiento para seleccionar el mejor modelo de un conjunto de alternativas que se ajustan a un conjunto de observaciones. Los algoritmos de aprendizaje automático tienen varios méritos, incluyendo la no linealidad, la tolerancia a la falla y la operación en tiempo real, haciéndolos adecuados para aplicaciones complejas (**Chang et al., 2021**).

1.1.3 Redes Neuronales y Aprendizaje Profundo (Deep Learning)

1.1.3.1 Conceptos básicos de Redes Neuronales y Aprendizaje Profundo

Las redes neuronales profundas (Deep Neural Networks, DNN) son una subcategoría del aprendizaje automático inspirada en la estructura y funcionamiento del cerebro humano. Estas redes están compuestas por múltiples capas de nodos interconectados, conocidos como neuronas, que procesan información y transmiten los resultados a las capas siguientes. A diferencia de las redes neuronales tradicionales, las DNN cuentan con múltiples capas ocultas que permiten extraer representaciones complejas y abstractas de los datos (**Lin et al., 2021; Qiu et al., 2022**). En el diagnóstico de enfermedades como el Alzheimer, las DNN se han destacado por su capacidad para identificar patrones en datos complejos, incluyendo imágenes de resonancia magnética y biomarcadores moleculares, facilitando así la detección temprana de la enfermedad (**Warren & Moustafa, 2023**). Al analizar automáticamente grandes volúmenes de datos, las DNN son capaces de identificar anomalías que podrían pasar desapercibidas para los especialistas clínicos (**Q. Yang et al., 2022**).

El aprendizaje profundo ha demostrado ser particularmente eficaz en la interpretación de datos de neuroimagen, como resonancias magnéticas funcionales (fMRI) y tomografías por emisión de positrones (PET), gracias a su capacidad para manejar información de alta dimensionalidad y reducir el ruido presente en las imágenes (**Lin et al., 2021; Warren & Moustafa, 2023**). Modelos específicos, como las Redes Neuronales Convolucionales (CNN), se utilizan frecuentemente para clasificar imágenes cerebrales, ayudando a distinguir entre diferentes estados cognitivos, desde el deterioro cognitivo leve hasta la demencia avanzada (**Qiu et al., 2022**). Además, técnicas como las Redes Generativas Antagónicas (GAN) y los Autoencoders se han implementado para mejorar la calidad de las imágenes y generar datos sintéticos

que apoyen la investigación y el desarrollo de modelos más robustos (**Lin et al., 2021**). Estos avances han convertido a las redes neuronales profundas en herramientas fundamentales para la creación de sistemas automatizados de diagnóstico que complementen las prácticas clínicas tradicionales, mejorando la precisión y rapidez en la detección del Alzheimer (**Qiu et al., 2022; Q. Yang et al., 2022**).

1.1.3.2 Aplicaciones de DL con imágenes médicas

En el contexto del diagnóstico de Alzheimer, las redes neuronales convolucionales (CNNs) han sido ampliamente aplicadas en imágenes de resonancia magnética (MRI) para detectar patrones sutiles de atrofia cerebral que son difíciles de identificar por métodos convencionales. Estudios recientes han demostrado que los modelos de aprendizaje profundo pueden superar el rendimiento de los expertos clínicos en la identificación de la enfermedad, con precisiones que alcanzan valores cercanos al 90% en diversas cohortes, incluyendo la iniciativa ADNI y otros estudios multicéntricos. Además, técnicas avanzadas como las redes completamente convolucionales (FCN) han permitido la generación de mapas de probabilidad de enfermedad, facilitando la interpretación de los resultados y ayudando a los profesionales de la salud a visualizar las regiones cerebrales afectadas (**Qiu et al., 2022**).

Más allá de las aplicaciones en MRI, el aprendizaje profundo también se ha empleado para analizar otras modalidades de imágenes médicas, como por ejemplo las fotografías retinianas. Estudios recientes han explorado el uso de redes neuronales para detectar signos de Alzheimer mediante cambios en la retina, aprovechando que esta estructura comparte similitudes anatómicas y funcionales con el cerebro (**Yousefzadeh et al., 2023**). Estas aproximaciones tienen el potencial de transformar el diagnóstico al ofrecer métodos no invasivos y de bajo costo que se pueden implementar en entornos de atención primaria, ampliando así el acceso al diagnóstico temprano (**Qiu et al., 2022; Warren & Moustafa, 2023**).

1.1.3.3 Arquitecturas preentrenadas en Deep Learning

La red de aprendizaje residual profunda (Deep Residual Learning) es una innovadora arquitectura para redes neuronales que permite crear modelos significativamente más profundos, lo que mejora la precisión en tareas de reconocimiento de imágenes, dentro de esta rama se encuentran un sinfín de arquitecturas preentrenadas, las cuales ya están configuradas de diferente manera para que puedan ser eficientes en distintas aplicaciones. ResNet (Residual Network) es una arquitectura de red neuronal convolucional profunda que se caracteriza por utilizar conexiones residuales que permiten saltar una o más capas, facilitando el flujo de

información a través de la red. Esto permite entrenar redes muy profundas de más de 100 capas, lo cual es una manera efectiva al mitigar el problema de la degradación del gradiente, mejorando significativamente el rendimiento en tareas de visión por computador, especialmente en clasificación de imágenes **(He et al., 2015; Herberg, 2023; Targ et al., 2016)**.

Por otro lado, EfficientNet es una familia de arquitecturas de redes neuronales convolucionales (CNN) diseñadas para optimizar simultáneamente la precisión y la eficiencia computacional en tareas de visión por computadora. Su principal innovación es el escalado compuesto, una técnica que ajusta de manera balanceada la profundidad (número de capas), el ancho (número de canales por capa) y la resolución de entrada (tamaño de la imagen), para maximizar el rendimiento con menor costo computacional. Con aproximadamente 66 millones de parámetros, EfficientNet-B7 alcanza una precisión Top-1 del 84.4% en ImageNet, colocándose entre los modelos más precisos de su tiempo, mientras que mantiene una eficiencia superior en comparación con arquitecturas como ResNet o Inception. Es ideal para aplicaciones que requieren máxima precisión, aunque con un mayor costo computacional **(Tan & Le, 2020)**.

Las dos arquitecturas han sido frecuentemente utilizadas para la predicción de imágenes en el diagnóstico médico de diferentes enfermedades que necesitan un gran análisis de datos.

1.1.4 Evaluación y Validación de los Modelos

1.1.4.1 Métricas de evaluación de modelos

En el desarrollo de modelos de aprendizaje automático para aplicaciones médicas, las métricas de evaluación desempeñan un papel crítico, ya que permiten medir de manera objetiva el desempeño de los algoritmos y su capacidad para generalizar datos nuevos en donde los valores a estudiar se dan por medio de una matriz 2x2, esta contiene información sobre el resultado que se genera entre los Valores Actuales junto con los Valores Predichos.

Tabla 1

Matriz de Confusión

		Valores Actuales	
		Positivos	Negativos
Valores Predichos	Positivos	VP	FP
	Negativos	FN	VN

Donde:

- VP: Verdaderos positivos, casos correctamente clasificados como positivos.
- FP: Falsos positivos, casos negativos incorrectamente clasificados como positivos.
- FN: Falsos negativos, casos positivos incorrectamente clasificados como negativos.
- VN: Verdaderos negativos, casos correctamente clasificados como negativos.

Precisión

La precisión mide la proporción de predicciones correctas que realiza el modelo sobre el total de predicciones positivas observadas incluyendo las falsas positivas. Es una métrica útil para obtener una visión general del rendimiento del modelo, especialmente cuando las clases están equilibradas (**Jude Chukwura Obi, 2023**).

$$Precisión = \frac{VP}{VP + FP}$$

Sin embargo, en problemas médicos donde las clases suelen estar desbalanceadas, como en el diagnóstico temprano del Alzheimer, la precisión por sí sola puede ser engañosa. Por ejemplo, si solo un pequeño porcentaje de los pacientes está enfermo, un modelo que siempre prediga "sano" podría obtener una precisión alta, pero fallar completamente en identificar casos positivos.

Sensibilidad (Recall)

También conocida como *recall* o tasa de verdaderos positivos, es la habilidad del modelo para identificar correctamente los verdaderos positivos (se incluyen los verdaderos positivos más los verdaderos positivos).

erróneamente clasificados como negativos (FN)) (**Jude Chukwura Obi, 2023**). Es especialmente importante en contextos médicos donde los falsos negativos (personas enfermas clasificadas como sanas) tienen consecuencias graves.

$$\text{Sensibilidad} = \frac{VP}{VP + FN}$$

Una sensibilidad alta indica que el modelo tiene una baja tasa de falsos negativos, lo cual es crucial en el diagnóstico de enfermedades como el Alzheimer, donde un diagnóstico temprano puede mejorar significativamente la calidad de vida del paciente. Por ejemplo, modelos que analizan imágenes de resonancia magnética cerebral suelen priorizar la sensibilidad para no pasar por alto casos críticos (**Sidey-Gibbons & Sidey-Gibbons, 2019**).

Especificidad

La especificidad mide la capacidad del modelo para identificar correctamente los casos negativos, es decir, aquellos pacientes que no presentan la enfermedad. Un valor alto de especificidad indica que el modelo es bueno aislando los casos negativos de los positivos, este complementa la sensibilidad y es crucial para evitar falsos positivos (**Jude Chukwura Obi, 2023**).

$$\text{Especificidad} = \frac{VN}{VN + FP}$$

En el ámbito médico, una alta especificidad reduce la posibilidad de diagnósticos erróneos en pacientes sanos, evitando procedimientos innecesarios, ansiedad y costos adicionales. Por ejemplo, en el contexto del Alzheimer, una especificidad alta asegura que las personas sanas no sean sometidas a tratamientos o exámenes invasivos innecesarios (**Chen et al., 2019**). Sin embargo, es importante equilibrar esta métrica con la sensibilidad para lograr un diagnóstico confiable.

F1-Score

El F1-score combina la precisión y la sensibilidad en una sola métrica, lo que lo hace especialmente valioso en datasets desbalanceados, donde una alta precisión podría no reflejar un buen desempeño si la sensibilidad

es baja. Por esta razón, el F1-score tiene en cuenta tanto los falsos positivos como los falsos negativos (**Jude Chukwura Obi, 2023**).

$$F1 - Score = 2 * \frac{Precisión * Sensibilidad}{Precisión + Sensibilidad}$$

Un F1-score alto indica que el modelo tiene un buen equilibrio entre precisión y sensibilidad, evitando tanto falsos positivos como falsos negativos. En el diagnóstico de enfermedades como el Alzheimer, un alto F1-score asegura que el modelo sea confiable tanto en la detección de casos como en la exclusión de pacientes sanos.

Curva ROC-AUC

La curva ROC (*Receiver Operating Characteristic*) – AUC (*Area Under the ROC Curve*) es una representación gráfica del modelo de clasificación en todos los umbrales de clasificación. La curva consiste en el gráfico de la tasa de verdaderos positivos (TVP) frente a la tasa de falsos positivos (TFP) para todos los umbrales de clasificación. El área bajo la curva ROC (AUC) proporciona una medida cuantitativa de la capacidad del modelo para discriminar entre clases positivas y negativas (**Jude Chukwura Obi, 2023**).

$$TVP (Sensibilidad) = \frac{VP}{VP + FN}$$

y

$$TFP = \frac{FP}{FP + VN}$$

- Una AUC cercana a 1 indica un modelo con excelente capacidad discriminativa, capaz de distinguir entre pacientes sanos y enfermos con alta precisión.
- Una AUC de 0.5 indica un modelo sin capacidad discriminativa, equivalente a realizar predicciones al azar.

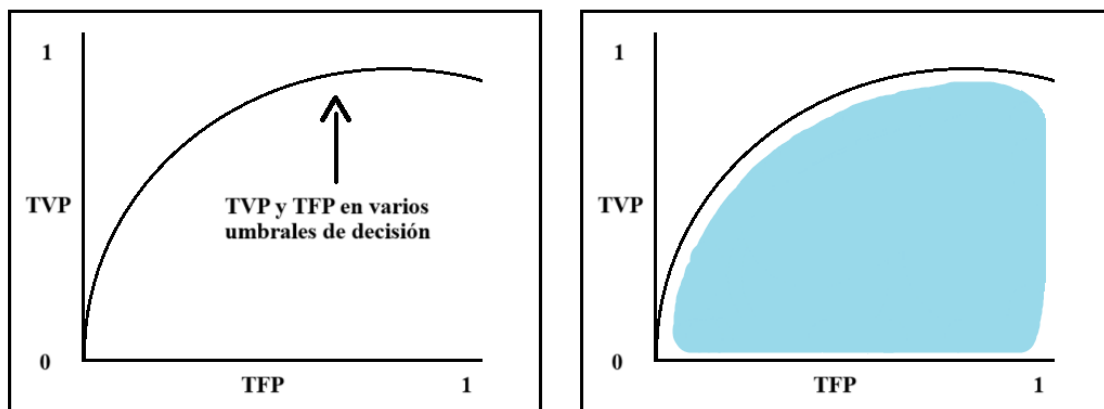


Figura 2. Representación de la curva ROC-AUC

Nota: En la figura A se puede observar una representación del TVP vs TFP en diferentes umbrales de clasificación indicados por las líneas de puntos. En la figura B se tiene el área debajo de la curva ROC. Modificado de **(Jude Chukwura Obi, 2023)**.

1.1.5 Interpretabilidad de los Modelos con Mapas de Calor

1.1.5.1 Importancia de la interpretabilidad en modelos de IA

La inteligencia artificial ha demostrado ser una herramienta poderosa en el análisis de imágenes médicas, especialmente en tareas complejas como el diagnóstico de enfermedades neurodegenerativas, entre las que destaca la enfermedad de Alzheimer. Sin embargo, uno de los principales desafíos en la implementación de modelos de IA en entornos clínicos es la falta de interpretabilidad. Esta limitación surge del hecho de que los modelos de aprendizaje profundo, como las redes neuronales convolucionales (CNN), funcionan como "cajas negras", generando predicciones sin ofrecer explicaciones claras de cómo llegan a sus conclusiones. En el contexto médico, esto puede dificultar la confianza y aceptación de estas herramientas por parte de los profesionales de la salud, además de limitar su utilidad en la validación clínica y el descubrimiento de nuevos patrones biomédicos **(Schutte et al., 2021; Zhang et al., 2021)**.

La interpretabilidad en IA es esencial por varias razones. En primer lugar, permite que los médicos comprendan y validen las decisiones tomadas por los modelos, asegurando que estas se basen en características relevantes desde un punto de vista clínico. Esto es especialmente importante en enfermedades como el Alzheimer, donde los cambios estructurales en el cerebro, como la atrofia del

hipocampo y la reducción del grosor cortical, son indicadores clave de la enfermedad **(Basheera & Ram, 2021; Puente-Castro et al., 2020)**. Además, la interpretabilidad facilita la detección de sesgos en los modelos, garantizando que las predicciones sean fiables y equitativas para diferentes grupos de pacientes **(S. Yang & Berdine, 2023)**. Finalmente, el cumplimiento normativo y ético en la medicina moderna exige que las herramientas de IA sean explicables, asegurando que los pacientes y médicos puedan confiar en sus resultados **(Saporta et al., 2022)**.

1.1.5.2 Mapas de calor (Grad-CAM)

Los mapas de calor generados mediante Grad-CAM (Gradient-weighted Class Activation Mapping) han emergido como una solución práctica para mejorar la interpretabilidad de los modelos de aprendizaje profundo aplicados al análisis de imágenes médicas. Grad-CAM permite identificar visualmente las áreas de una imagen que más contribuyen a las predicciones del modelo, generando mapas de activación que destacan regiones relevantes. Este enfoque es particularmente útil en el análisis de imágenes por resonancia magnética (MRI) en pacientes con Alzheimer, donde ayuda a identificar patrones anatómicos y estructurales relacionados con la enfermedad **(Fan et al., 2021)**.

El funcionamiento de Grad-CAM se basa en la extracción de las activaciones de las capas convolucionales de un modelo de IA, ponderadas según los gradientes calculados para una clase objetivo-específica. Esto genera un mapa visual que resalta las áreas que influyen en la decisión del modelo, como el hipocampo o la corteza entorrinal en el caso del Alzheimer. Por ejemplo, en estudios recientes que utilizaron datos de las bases ADNI y OASIS, Grad-CAM identificó consistentemente regiones clave como el hipocampo y los ventrículos laterales, esenciales para la clasificación de pacientes en etapas tempranas de la enfermedad **(Basheera & Ram, 2021; Puente-Castro et al., 2020)**. Además, estos mapas han validado predicciones y respaldado la identificación de biomarcadores críticos en imágenes estructurales de resonancia magnética **(Xiao et al., 2021)**.

Además, Grad-CAM ha demostrado ser una herramienta útil para identificar biomarcadores visuales en estudios longitudinales, permitiendo observar cómo cambian las regiones cerebrales afectadas a lo largo del tiempo. Por ejemplo, un modelo entrenado utilizando datos de ADNI aplicó Grad-CAM para analizar imágenes sagitales, identificando patrones de atrofia en áreas como la corteza entorrinal y el hipocampo, validando su utilidad en la predicción del Alzheimer **(Puente-Castro et al., 2020)**. Aunque Grad-CAM tiene ventajas claras, en imágenes de resonancia magnética de alta resolución, puede carecer de la precisión

necesaria para capturar detalles anatómicos finos. Esto ha llevado al desarrollo de variantes como Grad-CAM++, que mejora la localización de las regiones relevantes y proporciona un nivel de granularidad más detallado (Zhang et al., 2021).

1.1.6 Base de Datos

1.1.6.1 Descripción de la base de datos OASIS-1

La base de datos OASIS-1 (Open Access Series of Imaging Studies) es un recurso de acceso abierto diseñado para facilitar la investigación en neurociencia y aprendizaje automático mediante la provisión de imágenes de resonancia magnética (MRI) del cerebro humano. Esta base de datos contiene información de 416 sujetos con edades entre 18 y 96 años, incluyendo tanto individuos cognitivamente normales como pacientes con demencia leve (Mild Cognitive Impairment, MCI) o enfermedad de Alzheimer en sus etapas tempranas. Proporciona imágenes de resonancia magnética estructural (MRI) en alta resolución (T1-weighted) obtenidas con un escáner de 1.5T. Estas imágenes están diseñadas específicamente para estudios de neuroimagen y aprendizaje automático. La base de datos incluye tanto cortes axiales como reconstrucciones tridimensionales del cerebro, permitiendo realizar análisis volumétricos y morfométricos detallados. Adicionalmente, se ofrecen medidas clínicas como el puntaje del Mini-Mental State Examination (MMSE) y la Escala Clínica de Evaluación de Demencia (CDR). Estas características hacen de OASIS-1 un recurso versátil para estudiar biomarcadores relacionados con enfermedades neurodegenerativas. Una particularidad de los datos de OASIS-1 es la inclusión de participantes en diversas etapas de salud cognitiva, desde sujetos jóvenes y sanos hasta pacientes con deterioro cognitivo leve (MCI) o Alzheimer, lo que permite explorar diferencias a lo largo de un espectro continuo de envejecimiento y deterioro cognitivo (Marcus et al., 2007).

1.1.6.2 Descripción de la base de datos ADNI

La base de datos ADNI (Alzheimer's Disease Neuroimaging Initiative), establecida en 2004, se ha consolidado como uno de los recursos más completos y utilizados para el estudio del Alzheimer. Este proyecto tiene como objetivo principal desarrollar y validar biomarcadores para el diagnóstico temprano y seguimiento de la progresión de la enfermedad. ADNI incluye datos de miles de participantes clasificados en tres grupos según su estado cognitivo: cognitivamente normales (CN), pacientes con deterioro cognitivo

leve (MCI) y pacientes diagnosticados con Alzheimer (AD). Los datos disponibles incluyen resonancia magnética estructural (MRI) y tomografía por emisión de positrones (PET), empleando trazadores específicos para medir depósitos de beta-amiloide, metabolismo de glucosa y tau. Además, ADNI ofrece información complementaria como biomarcadores del líquido cefalorraquídeo (LCR), evaluaciones cognitivas extensas y datos genéticos, como el genotipo APOE asociado al riesgo de Alzheimer. Esta base de datos está disponible para investigadores registrados que cumplen con estrictos lineamientos éticos. La importancia de ADNI radica en su gran tamaño, la diversidad y calidad de sus datos, y su enfoque longitudinal, que permite estudiar la progresión de la enfermedad, lo cual es crítico en el desarrollo de algoritmos de aprendizaje automático para el diagnóstico temprano del Alzheimer **(Jack et al., 2008)**.

Por su parte, la base de datos ADNI-1 Annual 2Yr 3T ofrece imágenes adquiridas con un escáner de resonancia magnética de 3T, que proporciona una resolución más alta y una mayor sensibilidad para detectar cambios estructurales cerebrales asociados con el Alzheimer. Este subgrupo específico de la base ADNI incluye imágenes longitudinales, lo que significa que los participantes fueron evaluados en múltiples puntos temporales (anualmente durante dos años), permitiendo el análisis de la progresión de la enfermedad. Las imágenes incluyen datos en modalidades como T1-weighted para análisis estructural, T2-weighted para evaluar anomalías en la materia blanca, y mapas de densidad de protones. Además, estas imágenes están alineadas y corregidas para variaciones de intensidad, facilitando su uso en análisis automatizados como segmentación de estructuras cerebrales y detección de atrofia en regiones como el hipocampo y el córtex entorrinal **(Jack et al., 2008)**.

1.2 Objetivos

1.2.1 Objetivo General

Evaluar un modelo de aprendizaje automático basado en redes neuronales para el diagnóstico temprano de la enfermedad de Alzheimer.

1.2.2 Objetivos Específicos

- Construir un conjunto de datos de prueba a partir de información obtenida de la base de datos Oasis y ADNI de pacientes con y sin enfermedad del Alzheimer.
- Desarrollar el modelo de predicción basado en redes neuronales para la identificación temprana de la enfermedad de Alzheimer usando imágenes obtenidas de RM.
- Evaluar la precisión del modelo de aprendizaje automático sobre datos de pacientes con la enfermedad de Alzheimer.

CAPÍTULO II.- METODOLOGÍA

2.1 Materiales

El presente trabajo de titulación fue de tipo experimental “*in silico*”, el cual se llevó a cabo en las instalaciones de la Facultad de Ciencias e Ingeniería en Alimentos y Biotecnología de la Universidad Técnica de Ambato. En la Tabla 1 se detallan los programas y recursos bioinformáticos utilizados en el desarrollo de la investigación y en la Tabla 2 se especifican los equipos utilizados en la investigación.

Tabla 2.

Recursos Bioinformáticos

Recursos	Enlace
OASIS-1 (Open Access Series of Imaging Studies)	https://sites.wustl.edu/oasisbrains/
ADNI (Alzheimer’s Disease Neuroimaging Initiative)	https://adni.loni.usc.edu/
Python	https://www.python.org/
Spyder	https://www.spyder-ide.org/
Kaggle	https://www.kaggle.com/
cv2	https://opencv.org/
Numpy	https://numpy.org/
pandas	https://pandas.pydata.org/
seaborn	https://seaborn.pydata.org/
sklearn	https://scikit-learn.org/stable/
tensorflow	https://www.tensorflow.org/
matplotlib	https://matplotlib.org/

Tabla 3.

Equipos

Recursos	Cantidad
Computador personal	1
Servicio de internet con Router	1

2.2 Métodos

2.2.1 Construcción de un conjunto de datos de prueba a partir de información obtenida de la base de datos Oasis y ADNI de pacientes con y sin enfermedad del Alzheimer

2.2.1.1 Descarga y preparación de datos

Se inició descargando las carpetas del conjunto de datos OASIS-1, que contienen la información completa y detallada de los 416 pacientes. Además de las carpetas con imágenes, se descargó un archivo en formato Excel que proporcionaba información clínica y demográfica de todos los pacientes incluidos en el estudio, el cual fue fundamental para relacionar las imágenes con las características y diagnósticos específicos de cada paciente.

En una primera etapa, se realizó una depuración de las carpetas descargadas para reducir la complejidad del manejo de datos. De cada carpeta individual, se seleccionó exclusivamente el archivo brain.mgz, ya que este contiene la información de las imágenes de MRI correspondientes al cerebro completo. Esta selección permitió un manejo más eficiente de los datos, asegurando que cada archivo estuviera etiquetado correctamente según el identificador único de cada paciente. Posteriormente, se transformaron los archivos en formato .mgz a imágenes en formato .jpg, lo que facilitó la visualización y el procesamiento de las imágenes en etapas posteriores.

2.2.1.2 Selección y Filtrado de Imágenes

Para el análisis de las imágenes, se optó por realizar cortes axiales, ya que esta perspectiva permite una visualización completa y clara de las estructuras cerebrales. Este tipo de corte es adecuado, ya que permite analizar regiones críticas como el hipocampo y el córtex entorrinal. Sin embargo, para seleccionar imágenes representativas, se estableció un criterio de brillo mínimo de 980,000. Este criterio fue implementado para eliminar aquellas imágenes que no cumplían con los requisitos de calidad, lo cual es esencial para asegurar un entrenamiento preciso y fiable del modelo. Como resultado, se seleccionaron 35 cortes por paciente; en caso de que no se cumplieran estos criterios, las imágenes de dicho paciente fueron eliminadas. En total, 14 pacientes no cumplieron con los requisitos establecidos y fueron eliminados del conjunto de datos, ya

que conservar estas imágenes habría producido problemas de similitud redundante durante el entrenamiento del modelo.

2.2.1.3 Etiquetado y Balanceo de Datos

Una vez seleccionadas las imágenes representativas, se integraron con la información clínica contenida en el archivo Excel, de donde se descartaron aquellas imágenes que no tenían un diagnóstico registrado. Las imágenes restantes fueron etiquetadas de acuerdo con el diagnóstico del paciente, clasificándolas en las siguientes categorías: no dementia, mild dementia, moderate dementia, y severe dementia.

Posteriormente, las imágenes en formato BGR fueron convertidas a RGB y redimensionadas a un tamaño de 224x224 píxeles.

Tras el procesamiento y etiquetado de las imágenes, se realizó un análisis de la distribución del conjunto de datos, el cual indicó un desbalance significativo en la cantidad de imágenes por categoría. Las distribuciones iniciales fueron las siguientes: no dementia con 4,375 imágenes, mild dementia con 2,170 imágenes, moderate dementia con 595 imágenes, y severe dementia con únicamente 70 imágenes. Para abordar este desbalance, se implementó un proceso de data augmentation utilizando la biblioteca Keras. Este proceso generó imágenes adicionales para las clases con menor representación, asegurando que las categorías menos comunes tuvieran suficientes datos para entrenar el modelo de manera efectiva.

El data augmentation se realizó utilizando parámetros cuidadosamente seleccionados para evitar distorsiones significativas que pudieran alterar las características anatómicas críticas de las imágenes MRI. Los parámetros fueron los siguientes: rotación de 5°, desplazamiento horizontal y vertical de 0.02, y un rango de zoom de 0.02. Estos ajustes leves, posibilitaron aumentar el número de imágenes preservando la integridad de las estructuras cerebrales. Las nuevas cantidades de imágenes tras el aumento de datos fueron las siguientes: no dementia con 4,375 imágenes, mild dementia con 2,170 imágenes, moderate dementia con 2,345 imágenes, y severe dementia con 1,820 imágenes. Este balance en el conjunto de datos resultante es esencial para garantizar que el modelo de aprendizaje profundo pueda aprender de manera efectiva las diferencias entre las categorías.

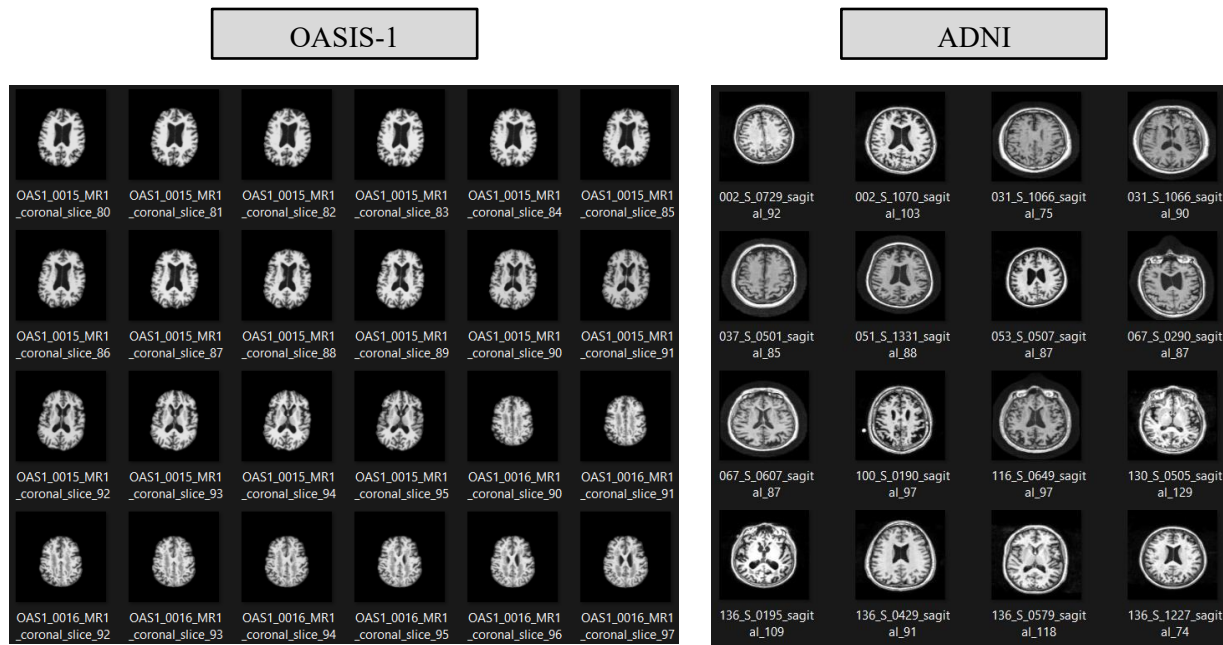


Figura 3. Ejemplo de Imágenes de Resonancia Magnética (RM) de las Bases de Datos OASIS-1 y ADNI

Nota. La figura muestra cortes axiales representativos de imágenes de resonancia magnética obtenidas de las bases de datos OASIS-1 y ADNI. Estas imágenes fueron empleadas para entrenar y evaluar modelos de aprendizaje automático orientados al diagnóstico temprano de la enfermedad de Alzheimer. Las diferentes imágenes se encuentran etiquetadas de acuerdo al estado de Alzheimer, en el caso de OASIS se tiene: no demencia, mild demencia, moderate demencia y severe demencia, por otro lado las imágenes de ADNI se etiquetaron como AD, CN y MCI.

2.2.2 Creación de las funciones necesarias para preparar el modelo de Deep Learning

2.2.2.1 Importación de las librerías principales

Para el desarrollo del modelo de aprendizaje profundo, se importaron diversas bibliotecas necesarias que permitieron gestionar archivos, manipular datos, crear visualizaciones, construir y evaluar el modelo.

Se utilizaron bibliotecas del sistema, tales como: os, time, shutil y pathlib, para la gestión y manipulación de archivos y directorios. Estas herramientas realizan operaciones como la carga, organización y reestructuración de los datos de entrada. Además, la biblioteca itertools se empleó para crear ciclos y repeticiones complejas de manera eficiente, especialmente para analizar datos y crear gráficos.

Para el manejo y procesamiento de datos, se utilizaron cv2 de OpenCV, numpy y pandas. OpenCV ayuda a realizar las operaciones sobre imágenes, incluyendo su lectura, transformación y preprocesamiento. Los paquetes numpy y pandas realizaron cálculos numéricos y gestionaron datos en estructuras tabulares como dataframes, para organizar y analizar los datos de entrada antes de ser ingresados al modelo.

En cuanto a la visualización de datos, se emplearon matplotlib.pyplot y seaborn. Estas bibliotecas crean gráficos para representar métricas y resultados obtenidos, como curvas de precisión y pérdida, matrices de confusión y análisis de clasificación. Además, se configuró el estilo visual darkgrid de Seaborn para garantizar una presentación uniforme y atractiva de las visualizaciones.

Para tareas relacionadas con aprendizaje automático y métricas de evaluación, se utilizó la biblioteca scikit-learn. Las cuales permitieron dividir los datos en conjuntos de entrenamiento y prueba mediante train_test_split. Además, sus funciones confusion_matrix y classification_report facilitaron la evaluación del desempeño del modelo, proporcionando métricas clave como la precisión, sensibilidad, especificidad, entre otras.

El desarrollo del modelo de aprendizaje profundo se basó en tensorflow y su API de alto nivel keras. Estas herramientas son útiles para la implementación de un modelo secuencial utilizando la clase Sequential. Se emplearon capas como Conv2D, MaxPooling2D, Flatten, Dense, Activation, Dropout y BatchNormalization, que ayudaron a construir una arquitectura que combinó extracción de características, activación de funciones, regularización y normalización de datos. Para la generación de datos, ImageDataGenerator permitió realizar aumentos dinámicos de imágenes, aumentando la robustez del modelo al exponerlo a variaciones de los datos de entrada. Además, se emplearon optimizadores como Adam y Adamax para ajustar los pesos durante el entrenamiento, y la métrica de pérdida categorical_crossentropy para evaluar el desempeño en problemas de clasificación multiclase. También se integraron regularizadores para evitar el sobreajuste del modelo mediante la penalización de pesos extremos.

Por último, se utilizó la biblioteca warnings para ignorar advertencias generadas durante la ejecución del código, asegurando que el flujo de trabajo se ejecute de manera limpia y sin interrupciones innecesarias. Para verificar la correcta carga de todos los módulos, se imprimió en la consola el mensaje: "modules loaded".

2.2.2.2 División del conjunto de datos

Para la división del conjunto de datos se implementó la función `define_paths(data_dir)`, la cual automatiza el proceso de extracción de rutas de archivos e identificación de etiquetas correspondientes. El directorio principal (`data_dir`) está estructurado en subcarpetas las cuales representan una clase específica. La función recorre cada una de estas subcarpetas y recopila las rutas completas de los archivos de imagen, asignando automáticamente las etiquetas en función del nombre de la carpeta. De este proceso se obtiene una lista que contiene las rutas de los archivos (`filepaths`) y otra lista que incluye las etiquetas (`labels`), asegurando que ambas se encuentren alineadas para su uso posterior.

Posteriormente, se agrega la función `define_df(files, classes)`, que convierte las listas obtenidas en un `DataFrame` de `pandas`. Este `DataFrame` organiza la información de manera estructurada en dos columnas: una columna almacena las rutas completas de los archivos (`filepaths`), mientras que la otra contiene las etiquetas asociadas a cada archivo (`labels`). La utilización de un `DataFrame` permite realizar análisis exploratorio, manipulaciones y particiones de manera eficiente, el cual se integra de forma nativa con bibliotecas que se utilizarán en el aprendizaje automático, como `sklearn` y `tensorflow`, lo que facilita la implementación del flujo de trabajo.

Posteriormente se utilizó la función `split_data(data_dir)` para dividir el conjunto de datos en tres subconjuntos: entrenamiento (80%), validación (10%) y prueba (10%). Para realizar esta partición, se empleó la función `train_test_split` de la biblioteca `sklearn`, aplicando estratificación en función de las etiquetas. La estratificación garantiza que la distribución de clases en cada subconjunto sea representativa del conjunto de datos completo, lo cual es fundamental para evitar sesgos en modelos entrenados sobre datos desbalanceados. La división se realiza en dos pasos: primero, los datos se separan en un conjunto de entrenamiento (80%) y un subconjunto temporal (20%). Posteriormente, el subconjunto temporal se divide equitativamente en los conjuntos de validación y prueba, asignando el 50% a cada uno. De este modo, se asegura que la proporción de clases se mantenga constante en todas las divisiones (Sadaiyandi et al., 2023).

El conjunto de entrenamiento se utiliza para ajustar los parámetros del modelo, mientras que el conjunto de validación permite optimizar hiperparámetros y monitorear la posible aparición de sobreajuste durante el proceso de aprendizaje. Finalmente, el conjunto de prueba proporciona una evaluación objetiva del rendimiento del modelo sobre datos completamente nuevos, que no fueron utilizados en etapas previas del entrenamiento.

2.2.2.3 Preprocesamiento y Generación de Datos

Los datos de entrenamiento, validación y prueba fueron preparados mediante la función `create_gens`, diseñada para generar tensores a partir de imágenes utilizando la clase `ImageDataGenerator` de la biblioteca Keras. Este enfoque permitió convertir las imágenes en un formato compatible con el modelo de aprendizaje profundo, además permite aplicar técnicas de aumento de datos y gestionar los lotes de imágenes de manera eficiente durante las etapas de entrenamiento, validación y prueba.

Las dimensiones de las imágenes fueron configuradas en 224x224 píxeles, mientras que el número de canales de color se estableció en formato RGB para las imágenes en color. Estos parámetros garantizaron que todas las imágenes fueran ajustadas al mismo tamaño y formato antes de ser utilizadas como entrada del modelo.

El tamaño del lote para el conjunto de datos de prueba fue calculado dinámicamente para optimizar el uso de la memoria y los recursos computacionales durante la evaluación. Este cálculo se realizó dividiendo el tamaño total del conjunto de prueba en grupos de imágenes, de tal manera que el número de lotes fuera un valor exacto, asegurando que cada lote tuviera un tamaño adecuado para ser procesado de manera eficiente.

Para el preprocesamiento de las imágenes, se implementó una función llamada `scalar`, el cual devolvía las imágenes en su formato original. Aunque esta función no realizó modificaciones adicionales, fue agregada con el propósito de ser modular y sustituible por funciones más complejas, como normalizaciones o transformaciones específicas.

Los datos de entrenamiento, validación y prueba fueron procesados mediante generadores de imágenes configurados con `ImageDataGenerator`. El generador de entrenamiento incluyó técnicas de aumento de datos, como la inversión horizontal de imágenes, lo que incrementó la diversidad del conjunto de datos y redujo el riesgo de sobreajuste del modelo. Por otro lado, los generadores de validación y prueba utilizaron únicamente la función de preprocesamiento básica, sin aplicar aumentos de datos, para asegurar una evaluación consistente del modelo.

Por otro lado, los generadores fueron configurados para cargar las imágenes y sus etiquetas directamente desde los `DataFrame` que contenían las rutas de los archivos y las categorías correspondientes. El generador de prueba, a diferencia de los demás, fue configurado sin barajar las imágenes para preservar el orden original de los datos.

2.2.2.4 Configuración de la visualización de datos

Se utilizó la función `show_images`, se tomó como entrada un generador de datos, desde este se extrajeron tanto las imágenes como las etiquetas correspondientes usando el método `next(gen)`, que devuelve un lote de datos (batch). Para identificar las clases presentes en el generador, se utilizó el atributo `class_indices`, que devuelve un diccionario en el que las claves representan los nombres de las clases y los valores corresponden a índices numéricos. Las claves de este diccionario se almacenaron en una lista para facilitar la asociación posterior de cada imagen con su clase.

Se determinó el número de imágenes a visualizar en cada lote, limitando la muestra a un máximo de 25 imágenes. Si el tamaño del lote era inferior a 25, se utilizó el total de imágenes disponibles en el lote actual. Este parámetro permitió mantener una visualización clara y ordenada dentro de un espacio definido.

Para representar las imágenes, se utilizó la biblioteca `matplotlib`, en esta se creó una figura de tamaño 20x20 píxeles para garantizar una visualización adecuada de todas las imágenes seleccionadas. A continuación, se iteró sobre el número de imágenes de la muestra, y en cada iteración se realizaron las siguientes acciones:

- Se normalizó cada imagen dividiendo los valores entre 255, escalándolos así al rango $[0, 1]$, lo cual es un paso estándar en el procesamiento de imágenes.
- Se determinó la clase correspondiente a la imagen mediante el uso de `np.argmax(labels[i])`, que identificó la posición del valor máximo en el vector de etiquetas, ya que este formato corresponde a una codificación *one-hot*. A partir de la posición obtenida, se recuperó el nombre de la clase correspondiente desde la lista de clases.
- Se añadió un título a cada imagen que mostraba el nombre de la clase en color azul y con un tamaño de fuente de 12. Además, se desactivaron los ejes para mejorar la claridad de la visualización.

Esta función ayudó en la inspección visual de los datos de entrada y facilitó la verificación de que el generador estuviera funcionando correctamente. Además, permitió comprobar la calidad de las imágenes preprocesadas y la distribución adecuada de las clases en cada lote.

2.2.2.5 Configuración de Callbacks para el Control del Entrenamiento

La clase MyCallback fue configurada con parámetros específicos que controlan su comportamiento. En primer lugar, el parámetro `patience` definió el número de épocas sin mejora en las que la tasa de aprendizaje se ajustaría antes de realizar una intervención. Por otro lado, `stop_patience` estableció el número máximo de veces que se permitiría ajustar la tasa de aprendizaje sin observar mejoras significativas, momento en el cual se detendría el entrenamiento automáticamente. Esto fue útil para evitar que el modelo siguiera entrenando innecesariamente en ausencia de progreso.

El parámetro `threshold` se utilizó para fijar un nivel de precisión en el entrenamiento. Una vez que la precisión alcanzaba ese nivel, la tasa de aprendizaje se ajustó según la pérdida en el conjunto de validación, lo que ayudó a mejorar el modelo en las últimas etapas del entrenamiento. Por otro lado, el parámetro `factor` controló cuánto se reducía la tasa de aprendizaje cada vez que no había mejoras, permitiendo que el modelo se acercara de manera más eficiente a valores mínimos de pérdida.

Adicionalmente, se incluyeron los `batches` y `epochs`, los cuales definieron la estructura del proceso de entrenamiento, es decir, la cantidad de lotes por época y el número total de épocas a ejecutar. El parámetro `ask_epoch` permitió introducir un control adicional al entrenamiento, solicitando una confirmación manual para continuar después de un número específico de épocas, lo cual resultó útil en experimentos iterativos y en situaciones donde era necesario evaluar resultados intermedios.

La clase también configuró variables de control internas que facilitaron el monitoreo del proceso. La variable `count` registró el número de veces que se redujo la tasa de aprendizaje sin mejora, mientras que `stop_count` monitoreó las intervenciones realizadas para detener el entrenamiento. Las variables `highest_tracc` y `lowest_vloss` almacenaron, respectivamente, la precisión más alta alcanzada en el conjunto de entrenamiento y la pérdida más baja observada en el conjunto de validación. Además, se definieron `initial_lr` e `initial_weights` para guardar la tasa de aprendizaje y los pesos iniciales del modelo, respectivamente, garantizando que el entrenamiento pudiera ser reiniciado o restaurado si era necesario. De igual forma, la variable `best_weights` fue incluida para almacenar los mejores pesos del modelo durante el entrenamiento, específicamente en la época en la que se alcanzó la pérdida mínima. Esto restauró el estado óptimo del modelo al final del proceso.

2.2.2.6 Creación de imágenes del historial y entrenamiento del modelo

Se implementó la función `plot_training` con el objetivo de visualizar el desempeño del modelo durante el proceso de entrenamiento mediante gráficos que mostraron la evolución de la precisión y la pérdida tanto en el conjunto de entrenamiento como en el conjunto de validación. Esta visualización ayudó a identificar las mejores épocas de entrenamiento y evaluar la estabilidad y generalización del modelo.

La función tomó como entrada el objeto de historial generado durante el entrenamiento, el cual contenía registros de las métricas de precisión y pérdida para entrenamiento (`accuracy` y `loss`) y validación (`val_accuracy` y `val_loss`). A partir de este historial, se extrajeron los valores requeridos para representar gráficamente cómo evolucionaron las métricas a lo largo de todas las épocas.

Se identificaron dos parámetros clave dentro del historial: la época con la menor pérdida en validación y la época con la mayor precisión en validación. La primera se obtuvo utilizando la función `np.argmin(val_loss)`, que permitió localizar el índice donde la pérdida alcanzó su valor mínimo. La segunda se calculó mediante `np.argmax(val_accuracy)`, que identificó el índice en el que se observó la mayor precisión en el conjunto de validación. Estos valores fueron resaltados posteriormente en los gráficos con marcadores especiales y etiquetas descriptivas.

La función organizó los resultados en una figura compuesta. En el primer gráfico, se representaron las curvas de pérdida para los datos de entrenamiento y validación a lo largo de las épocas. Se destacó la época con la menor pérdida utilizando un marcador en color azul y una etiqueta que indicó la mejor época encontrada. En el segundo gráfico, se trazaron las curvas de precisión para entrenamiento y validación, señalando la época con la mayor precisión en validación de manera similar. Estos gráficos facilitaron la interpretación del progreso del modelo y su capacidad de generalización a datos no vistos.

Para mejorar la visualización, se configuró el estilo `fivethirtyeight` de la biblioteca `matplotlib`, proporcionando un diseño más limpio y estético. Además, se definió un tamaño de figura de `20x8` para asegurar una representación clara de ambos gráficos. Los puntos más relevantes, correspondientes a la mejor pérdida y la mayor precisión, se resaltaron con un marcador de gran tamaño (`s=150`) y en color azul. Se incluyeron etiquetas descriptivas en los ejes, títulos apropiados y leyendas para facilitar la interpretación de las curvas y los puntos destacados.

Adicionalmente, se aplicó `plt.show()` para mostrar la figura generada, con lo cual se analizó de manera visual el progreso del modelo durante su entrenamiento. Esta representación gráfica fue fundamental para evaluar el comportamiento del modelo, identificar problemas como sobreajuste o subajuste, y determinar en qué época se alcanzó el mejor desempeño para la validación.

2.2.2.7 Creación de la matriz de confusión

La función denominada `plot_confusion_matrix`, tomó como entradas principales la matriz de confusión (`cm`), los nombres de las clases (`classes`), un indicador de normalización (`normalize`), el título del gráfico (`title`) y el mapa de colores a utilizar (`cmap`). Estas configuraciones permitieron ajustar tanto la apariencia como el contenido del gráfico según los requerimientos del análisis.

Inicialmente se generó una figura de 10 x 10 pulgadas para la representación gráfica de la matriz. Se utilizó la función `plt.imshow` para mostrar la matriz como una imagen en la que los valores se codificaron por colores, definidos por el mapa `plt.cm.Blues`. A continuación, se añadió un título y barra de colores para proporcionar contexto visual y facilitar la interpretación de los valores. Los nombres de las clases se mostraron como etiquetas en los ejes X e Y, utilizando la función `plt.xticks` y `plt.yticks`. Para mejorar la legibilidad, las etiquetas del eje X se rotaron 45 grados.

Para resaltar los valores de la matriz en el gráfico, se utilizó un bucle que iteró sobre todas las celdas de la matriz. Con la función `plt.text`, se añadió el valor de cada celda directamente en su posición correspondiente. El color del texto cambió dinámicamente según el valor de la celda: blanco para valores mayores a la mitad del valor máximo de la matriz (`thresh`) y negro para los menores. Este contraste mejoró la legibilidad del gráfico.

Se ajustaron los márgenes con `plt.tight_layout()` y se añadieron etiquetas a los ejes, indicando claramente que el eje Y correspondía a las etiquetas verdaderas y el eje X a las etiquetas predichas. La función permitió analizar fácilmente el desempeño del modelo, destacando las clases correctamente clasificadas (diagonal principal) y los errores de clasificación (valores fuera de la diagonal).

2.2.3 Estructura del modelo CNN

2.2.3.1 Preparación de los datos para el modelo de aprendizaje profundo

Se definió la ruta de acceso al directorio que contenía las imágenes preprocesadas, las cuales estuvieron organizadas en un formato de 224x224 píxeles.

Se implementó un bloque try-except para manejar posibles errores durante la ejecución. En el bloque try, los datos se dividieron en conjuntos de entrenamiento, validación y prueba mediante la función `split_data`, garantizando que cada conjunto estuviera correctamente organizado para las etapas posteriores del modelo. A continuación, se utilizaron estas divisiones para generar tensores a partir de las imágenes mediante la función `create_gens`, que permitió configurar generadores de datos con un tamaño de lote definido en 40. Estos generadores fueron esenciales para alimentar las imágenes al modelo en lotes, optimizando así el uso de memoria durante el entrenamiento.

2.2.3.2 Definición de la arquitectura del modelo de red neuronal convucional (CNN)

El modelo inició con una serie de capas convolucionales diseñadas para extraer características relevantes de las imágenes de entrada, las cuales tienen un tamaño de 224x224 píxeles y tres canales de color (RGB). En la primera capa convolucional, se aplicaron 32 filtros de tamaño 3x3, con activación ReLU (Rectified Linear Unity) un esquema de relleno (padding) establecido en "same" para mantener las dimensiones espaciales. Posteriormente, se incorporó una capa de agrupamiento máximo (MaxPooling) con un tamaño de 2x2 para reducir la dimensionalidad de las características extraídas y optimizar el procesamiento.

El proceso se repitió en las siguientes capas convolucionales, aumentando el número de filtros en cada etapa (64, 128 y 256 filtros, respectivamente). Cada capa permitió que la red aprendiera características más complejas, desde patrones básicos en las primeras capas hasta patrones más abstractos en las capas profundas. Todas las capas convolucionales emplearon la función de activación ReLU, la cual introduce no linealidad en el modelo al permitir que solo los valores positivos se mantengan, mientras que los negativos se establecen en cero. Esto facilitó que la red neuronal capturó patrones complejos y no lineales en los datos. Además, cada capa convolucional fue seguida de una capa de agrupamiento máximo (MaxPooling), diseñada para reducir las dimensiones espaciales de las características extraídas. Este proceso seleccionó el

valor máximo en regiones específicas de las características generadas, lo que ayudó a disminuir la carga computacional y a resaltar los patrones más relevantes aprendidos en cada etapa del modelo.

Una vez utilizadas las capas convolucionales, se utilizó una capa de aplanamiento (Flatten) para transformar los datos multidimensionales en un vector unidimensional. Esto permitió conectar las características extraídas a una serie de capas densas completamente conectadas, conocidas como capas fully connected. La primera de estas capas incluyó 512 neuronas y una activación ReLU, seguida de una capa Dropout con una tasa del 50 %, lo que ayudó a prevenir el sobreajuste al eliminar aleatoriamente conexiones durante el entrenamiento. Este patrón se repitió con capas densas de 256 y 128 neuronas, asegurando una progresiva reducción de dimensiones hacia la capa de salida.

En la capa final, se definieron 5 neuronas con una función de activación softmax, adecuada para problemas de clasificación multiclase, ya que produce probabilidades normalizadas para cada categoría. Adicionalmente, se incluyó una capa Dense con 4 salidas y activación softmax, adaptada a un caso específico del modelo.

Finalmente, el modelo se compiló utilizando el optimizador Adam con una tasa de aprendizaje de 0.0001, lo que permitió ajustes eficientes de los pesos durante el entrenamiento. La función de pérdida categorical_crossentropy se seleccionó por su capacidad en problemas de clasificación multiclase, y la métrica de evaluación definida fue accuracy, para monitorear la proporción de predicciones correctas durante el entrenamiento y validación.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 224, 224, 32)	896
max_pooling2d_4 (MaxPooling2D)	(None, 112, 112, 32)	0
conv2d_5 (Conv2D)	(None, 112, 112, 64)	18,496
max_pooling2d_5 (MaxPooling2D)	(None, 56, 56, 64)	0
conv2d_6 (Conv2D)	(None, 56, 56, 128)	73,856
max_pooling2d_6 (MaxPooling2D)	(None, 28, 28, 128)	0
conv2d_7 (Conv2D)	(None, 28, 28, 256)	295,168
max_pooling2d_7 (MaxPooling2D)	(None, 14, 14, 256)	0
flatten_1 (Flatten)	(None, 50176)	0
dense_5 (Dense)	(None, 512)	25,690,624
dropout_2 (Dropout)	(None, 512)	0
dense_6 (Dense)	(None, 256)	131,328
dropout_3 (Dropout)	(None, 256)	0
dense_7 (Dense)	(None, 128)	32,896
dense_8 (Dense)	(None, 5)	645
dense_9 (Dense)	(None, 4)	24

Total params: 26,243,933 (100.11 MB)

Trainable params: 26,243,933 (100.11 MB)

Non-trainable params: 0 (0.00 B)

Figura 4. Resumen de la arquitectura del modelo

Nota. El modelo está compuesto por capas convolucionales (Conv2D) que incrementan progresivamente el número de filtros (32, 64, 128 y 256), permitiendo extraer características cada vez más complejas de las imágenes, seguidas de capas de agrupamiento máximo (MaxPooling2D), las cuales reducen la dimensionalidad espacial sin afectar los canales. Estas capas convolucionales culminan en una capa de aplanamiento (Flatten), que convierte las características tridimensionales en un vector unidimensional con 50,176 valores. Posteriormente, las capas densas (Dense) procesan este vector en unidades completamente conectadas, comenzando con 512 neuronas y reduciéndose gradualmente hasta 5, para realizar la clasificación final, con activación softmax. En total, el modelo tiene 26,243,933 parámetros, todos entrenables, lo que refleja su alta capacidad de aprendizaje y su capacidad para manejar datos complejos.

2.2.3.3 Entrenamiento del modelo CNN

En este paso, se definieron los valores clave para los parámetros de los *callbacks*. El tamaño de lote (*batch_size*) y el número de épocas (*epochs*) se fijaron en 40, mientras que la paciencia inicial (*patience*) y la paciencia máxima antes de detener el entrenamiento (*stop_patience*) se configuraron en 1 y 3 respectivamente. El umbral de precisión del entrenamiento (*threshold*) se estableció en 0.9, y el factor para

reducir la tasa de aprendizaje dinámicamente (factor) se fijó en 0.5. Además, el parámetro `ask_epoch` indicó que se solicitaría confirmación manual para continuar después de 5 épocas, y el número de lotes por época se calculó dividiendo el tamaño del conjunto de entrenamiento entre el tamaño del lote.

Se implementaron tres *callbacks* principales:

1. **MyCallback:** Este supervisó el progreso del entrenamiento y ajustó dinámicamente la tasa de aprendizaje cuando las métricas no mostraban mejoras, además de gestionar interrupciones basadas en los parámetros definidos.
2. **EarlyStopping:** El cual interrumpió el entrenamiento cuando no se observaban mejoras en la pérdida de validación durante 17 épocas consecutivas, restaurando automáticamente los pesos del modelo correspondientes a la mejor época.
3. **ModelCheckpoint:** Se configuró para guardar automáticamente el modelo completo en puntos de control, utilizando un formato basado en las épocas y la precisión de validación (`val_accuracy`). Esto permitió recuperar el modelo en su estado óptimo, incluso después de interrupciones.

Se proporcionó el conjunto de datos de entrenamiento al modelo mediante el generador `train_gen`, mientras que el conjunto de validación, suministrado por `valid_gen`, se utilizó para evaluar el rendimiento del modelo en cada época y evitar el sobreajuste. El objeto `history` recopiló los resultados de cada época, incluyendo métricas como la pérdida y la precisión para los conjuntos de entrenamiento y validación. Estos resultados fueron esenciales para evaluar el comportamiento del modelo, identificar posibles problemas como el sobreajuste o subajuste, y guiar el análisis posterior del rendimiento.

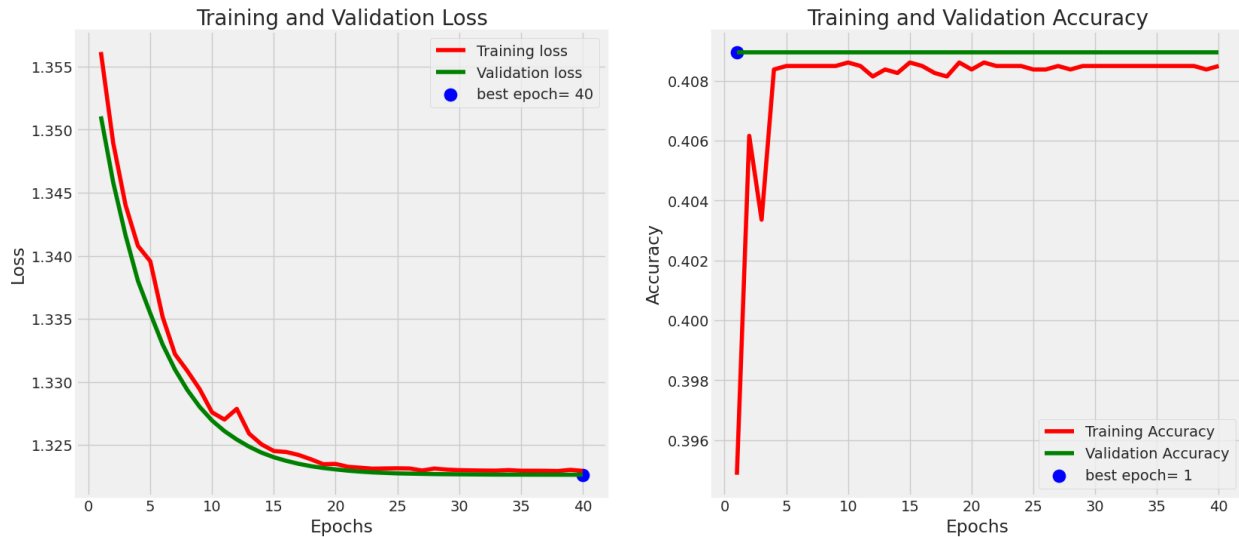


Figura 5. Curvas de Pérdida y Precisión del Modelo Durante el Entrenamiento

Nota: En el gráfico de la izquierda se representa la evolución de la pérdida en los conjuntos de entrenamiento (línea roja) y validación (línea verde). Ambas curvas muestran una tendencia decreciente, indicando que el modelo mejoró su capacidad para minimizar el error en ambos conjuntos. La mejor época en términos de pérdida de validación fue en la última, donde se resalta con un punto azul, dando en conclusión que el modelo continuó aprendiendo de manera efectiva hasta el final del entrenamiento.

Por otro lado, en el gráfico de la derecha se muestra la evolución de la precisión en los conjuntos de entrenamiento (línea roja) y validación (línea verde). En el cual, el modelo alcanzó su mejor precisión de validación en la primera época, que está marcado con azul. Sin embargo, las curvas son relativamente estables después de las primeras épocas, lo que se comprende que el modelo logró una generalización consistente, aunque con una mejora limitada en la precisión con respecto al conjunto de validación.

2.2.4 Construcción del modelo EfficientNetB7 y ResNet50

2.2.4.1 Carga del modelo base

Por un lado, se utilizó la arquitectura EfficientNetB7 y de igual forma ResNet50 como modelos base. Se aplicó los mismos pasos para las dos arquitecturas. En primera instancia se excluyó su capa superior de clasificación (`include_top=False`), lo cual permitió reutilizar las capacidades preentrenadas de extracción de características, optimizadas en conjuntos de datos como ImageNet. Los pesos del modelo preentrenado

se cargaron desde un archivo externo, ubicado en una ruta específica, asegurando la configuración inicial para el problema de clasificación.

2.2.4.2 Congelación de capas

Inicialmente, todas las capas del modelo base se congelaron (`layer.trainable = False`) para evitar que los pesos preentrenados fueran modificados durante las primeras etapas del entrenamiento. Esta decisión permitió preservar las características generales previamente aprendidas y enfocarse en personalizar el modelo con las capas superiores.

2.2.4.3 Ajuste fino

Para adaptar el modelo a las necesidades del problema, se descongelaron las últimas 10 capas base. Estas capas más profundas, responsables de aprender patrones específicos, ajustaron sus pesos en función de las particularidades del nuevo conjunto de datos.

2.2.4.4 Adición de capas personalizadas

Para EfficientNetB7 y ResNet50, se utilizó una capa llamada `GlobalAveragePooling2D`, la cual resume la información tridimensional generada por el modelo base en un vector unidimensional. Reduciendo la complejidad de los datos y condensando las características más importantes en un formato manejable para las capas posteriores.

En el caso de EfficientNetB7, se agregó una capa totalmente conectada con 512 neuronas, a diferencia de ResNet50 donde la capa densa agregada fue de 1024 neuronas y ambas con activación ReLU. Esta capa tomó las características resumidas por la capa anterior y aprendió relaciones más complejas entre ellas. Para evitar que el modelo se sobreajuste, se incluyó una capa de Dropout con una tasa del 50 %. Esta capa elimina aleatoriamente conexiones entre las neuronas durante el entrenamiento, lo que mejora la capacidad del modelo para generalizar en datos nuevos.

A continuación, se añadió una capa de salida con 4 neuronas, correspondiente al número de categorías del problema de clasificación. Esta capa utilizó la activación softmax, que transforma las salidas en

probabilidades normalizadas, es decir, cada neurona indica la probabilidad de que una muestra pertenezca a una categoría específica.

2.2.4.5 Compilación y entrenamiento

Los modelos fueron compilados utilizando el optimizador Adam, configurado con una tasa de aprendizaje baja 0.0001 para garantizar ajustes precisos y estables durante el entrenamiento. La función de pérdida seleccionada fue `categorical_crossentropy`, adecuada para problemas de clasificación multiclase, y se monitorizó la métrica de precisión (`accuracy`) para evaluar el rendimiento del modelo.

El entrenamiento se llevó a cabo utilizando los generadores definidos previamente para los conjuntos de entrenamiento y validación, los cuales fueron: monitoreo del progreso, puntos de control para guardar el modelo, y parada temprana para finalizar el entrenamiento si no se observaban mejoras significativas.

2.2.4.6 Descongelar algunas capas del modelo base

Se descongelaron las últimas 10 capas de la arquitectura base de EfficientNetB7, para que dichas capas congeladas entrenaran con los datos específicos del problema

2.2.4.7 Recopilación del modelo

Tras descongelar las capas seleccionadas, se recompiló el modelo utilizando una tasa de aprendizaje más baja 0.00001 para evitar alteraciones bruscas en los pesos preentrenados. Ya que permite que el modelo realice ajustes graduales y estables en las capas descongeladas, maximizando la efectividad de la transferencia de aprendizaje. La función de pérdida `categorical_crossentropy` y la métrica `accuracy` se mantuvieron para evaluar el desempeño.

2.2.4.8 Entrenamiento con capas descongeladas

El entrenamiento continuó utilizando los generadores de entrenamiento y validación previamente definidos en los callbacks, para lo cual, el modelo utilizó las capacidades preentrenadas en las capas congeladas y

adaptó las capas descongeladas a las características específicas del conjunto de datos, mejorando su capacidad en datos no vistos.

2.2.5 Evaluación de la precisión del modelo de aprendizaje automático sobre datos de pacientes con la enfermedad de Alzheimer.

2.2.5.1 Realizar las predicciones

Las predicciones se generaron por medio de `model.predict`, que devolvió una matriz con probabilidades asociadas a cada clase para cada imagen. Posteriormente, se determinó la clase predicha para cada imagen seleccionando el índice con la probabilidad más alta mediante la función `np.argmax`. Estas predicciones fueron comparadas con las etiquetas verdaderas, que se obtuvieron directamente del generador de prueba (`test_gen.classes`).

Para facilitar la interpretación de los resultados, las etiquetas predichas, representadas inicialmente como índices numéricos, se mapearon a sus nombres correspondientes. Esto se logró utilizando la lista de nombres de clases obtenida desde `test_gen.class_indices.keys()`. Este paso permitió que las etiquetas se mostraran en texto, lo que facilitó el análisis visual de las predicciones. Adicionalmente, el generador de prueba fue reiniciado (`test_gen.reset()`) para garantizar que las imágenes pudieran ser leídas nuevamente para la visualización.

Se implementó una función personalizada, denominada `plot_images`, para visualizar un subconjunto de imágenes del generador de prueba junto con sus etiquetas verdaderas y las predichas. La función tomó como entrada las imágenes, las etiquetas reales, las predicciones realizadas por el modelo, los nombres de las clases y el número de imágenes a mostrar. Dentro de esta función, se configuró una figura adecuada para mostrar las imágenes seleccionadas, y cada imagen se presentó en un subgráfico con un título que incluía la etiqueta verdadera y la predicha. Además, se desactivaron los ejes para centrar la atención en las imágenes y la información de las etiquetas.

Luego, se utilizaron cinco imágenes del conjunto de prueba como ejemplo para visualizar las predicciones del modelo. Esta visualización sirvió para inspeccionar cualitativamente el rendimiento del modelo, permitiendo identificar casos donde las predicciones coincidieron con las etiquetas reales y casos donde ocurrieron errores.

2.2.5.2 Generación de curva ROC-AUC

Se obtuvieron las etiquetas verdaderas de las clases y se binarizaron utilizando la función `label_binarize`. Este paso convirtió las etiquetas originales en una matriz binaria, donde cada clase se representó como una columna. Esto permitió calcular las métricas ROC y AUC para cada clase individualmente. El número de clases (`n_classes`) se obtuvo a partir del número de columnas en esta matriz binarizada.

Para cada clase, se calcularon las tasas TPR y FPR utilizando la función `roc_curve`, que toma como entrada las etiquetas binarizadas y las probabilidades predichas por el modelo. A partir de estas tasas, se calculó el área bajo la curva (AUC) utilizando la función `auc`, que cuantifica el rendimiento de clasificación para cada clase.

Además de las curvas ROC individuales, se calculó una curva macro-promediada, que combina la información de todas las clases en una sola curva. Para esto, se interpolaron las TPRs de cada clase a través de un conjunto único de FPRs (`all_fpr`), y se calculó el promedio de las TPRs interpoladas. El área bajo esta curva macro-promediada se computó nuevamente utilizando `auc`.

Se generó un gráfico que muestra las curvas ROC individuales para cada clase y la curva macro-promediada. Cada curva ROC individual se representó con un color distinto, acompañado de su valor AUC en la leyenda. La curva macro-promediada se trazó con una línea punteada de color azul marino, proporcionando una visión general del desempeño del modelo en todas las clases. También se incluyó la diagonal de referencia ($y=x$), que representa un rendimiento aleatorio.

2.2.5.3 Matriz de confusión y reporte de clasificación

La matriz de confusión fue calculada utilizando la función `confusion_matrix`, comparando las etiquetas verdaderas (`true_classes`) con las predichas (`predicted_classes`). Esta matriz identificó los aciertos y errores de clasificación para cada categoría, mostrando en filas las clases reales y en columnas las clases predichas. La visualización de la matriz de confusión se realizó mediante un gráfico de calor generado con Seaborn, donde los valores en cada celda fueron anotados y las etiquetas de las clases se incluyeron en los ejes para una interpretación clara.

Además, se calcularon las especificidades para cada clase, utilizando como base la matriz de confusión (cm) y las etiquetas de las clases (class_labels). El cálculo de la especificidad se realiza sumando los verdaderos negativos y los falsos positivos para cada clase, y luego dividiendo los verdaderos negativos entre el total.

2.2.5.4 Gráfico Grad-CAM

Para preprocesar la imagen se la cargó desde una ruta específica, se redimensionó al tamaño esperado por el modelo y se transformó en un arreglo numérico. Luego, se expandieron sus dimensiones para simular un lote y se aplicó el preprocesamiento específico de EfficientNet para normalizar los valores de los píxeles según el rango requerido.

Posteriormente, se definió una función para generar el heatmap Grad-CAM asociado a una capa específica del modelo. Este proceso implicó crear un modelo secundario que conectara las entradas originales con la salida de una capa seleccionada y las predicciones finales. Utilizando GradientTape, se calcularon los gradientes de las predicciones con respecto a la salida de la capa seleccionada, permitiendo identificar qué regiones de la imagen contribuyeron más a la decisión del modelo. Estos gradientes se combinaron con las activaciones de la capa para generar un heatmap. Finalmente, el heatmap se normalizó aplicando ReLU para eliminar valores negativos y escalándolo entre 0 y 1.

Para facilitar la interpretación visual, se creó otra función que superpuso el heatmap generado sobre la imagen original. Esta función redimensionó el heatmap para que coincidiera con el tamaño de la imagen, se aplicó un mapa de colores (COLORMAP_JET) para destacar las intensidades y combinó el heatmap con la imagen mediante un coeficiente de transparencia ($\alpha=0.4$).

De igual forma, se seleccionaron varias capas internas del modelo para analizar diferentes niveles de abstracción, incluidas block6d_project_conv, block7d_expand_conv, block7d_project_conv y top_conv. Para cada capa, se generó un heatmap y se superpuso a la imagen original. Los resultados fueron visualizados en un gráfico con múltiples subtramas, donde cada subgráfico mostró el heatmap asociado a una capa diferente, resaltando las regiones clave que influyeron en las decisiones del modelo.

2.2.5.5 Prueba con datos nuevos de OASIS y ADNI

El flujo de trabajo se realizó tanto para los datos ADNI como para OASIS, con esto se pudo obtener una base de datos de prueba grande para poder comparar entre ellas la eficiencia del algoritmo.

Para adaptar las imágenes al formato requerido por el modelo preentrenado, se definió una función personalizada de preprocesamiento. Esta función utilizó las herramientas de Keras, como `load_img` para cargar las imágenes desde sus rutas y redimensionarlas a un tamaño estándar de 224x224 píxeles. Posteriormente, se convirtió cada imagen en un arreglo numérico mediante `img_to_array`, y se normalizaron los valores de los píxeles utilizando `preprocess_input`, que es específico para la arquitectura EfficientNet y ResNet, se expandieron las dimensiones del arreglo con `np.expand_dims` para simular un lote de entrada.

La clase predicha fue determinada como la categoría con la mayor probabilidad usando `np.argmax`, tanto la etiqueta verdadera como la predicha fueron almacenadas para el análisis posterior. Además, se calcularon estadísticas globales y por clase, incluyendo el conteo de clasificaciones correctas e incorrectas. Estas métricas se mantuvieron en diccionarios específicos para cada clase, permitiendo un análisis desglosado por categoría.

Una vez procesadas todas las imágenes, se generaron métricas de clasificación utilizando las herramientas de Scikit-learn. Se empleó la función `classification_report` para calcular precisión, recuperación y F1-score por clase, mientras que `confusion_matrix` se utilizó para crear una matriz de confusión que identificó los aciertos y errores entre las categorías.

Además del análisis cuantitativo, se visualizó un conjunto de imágenes con sus etiquetas verdaderas y las predicciones realizadas por el modelo. Esto permitió una inspección cualitativa del comportamiento del modelo. Utilizando Matplotlib, se mostraron hasta 100 imágenes con títulos que indicaban la etiqueta verdadera y la predicha.

CAPÍTULO III.- RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de los resultados

3.1.1 Entrenamiento de las arquitecturas EfficientNetB7 y ResNet50

La implementación de arquitecturas modernas como EfficientNetB7 y ResNet50 ha llevado este enfoque un paso más allá. Estas redes neuronales han sido diseñadas para optimizar la extracción de características relevantes, adaptándose a la complejidad y variabilidad de los datos de imágenes médicas (**Tan & Le, 2020**). Técnicas como el aumento de datos y la transferencia de aprendizaje permiten entrenar modelos robustos capaces de generalizar diferentes conjuntos de datos, como OASIS y ADNI. Esto no solo mejora la precisión diagnóstica, sino que también abre posibilidades para la aplicación clínica en contextos diversos.

El impacto de estas tecnologías no se limita al diagnóstico. Al proporcionar una herramienta rápida y precisa, los modelos de IA permiten a los profesionales de la salud identificar a los pacientes en riesgo de manera oportuna, facilitando la implementación de estrategias de tratamiento temprano e inclusión en ensayos clínicos (**Lu et al., 2018**). Además, estos avances contribuyen al estudio de la progresión del Alzheimer, ofreciendo una perspectiva más profunda sobre los mecanismos de la enfermedad y ayudan a personalizar las intervenciones terapéuticas. Aunque todavía existen desafíos, como la necesidad de datos estandarizados y la validación clínica de los modelos, los avances recientes posicionan a la IA como un aliado fundamental en la lucha contra el Alzheimer, con el potencial de transformar significativamente la forma en que se diagnostica y gestiona esta devastadora enfermedad.

El vínculo entre los avances en IA y la mejora del diagnóstico temprano del Alzheimer radica en la capacidad de estas tecnologías para abordar las complejidades inherentes a la enfermedad. Al combinar modelos avanzados como EfficientNetB7 y ResNet50 con bases de datos clínicas robustas, es posible identificar patrones sutiles que no son fácilmente detectables mediante métodos convencionales.

Tras importar los datos y entrenar las arquitecturas EfficientNetB7 y ResNet50, se obtuvieron resultados significativos que permiten evaluar el desempeño y comportamiento de ambos modelos a lo largo de 40 épocas. En esta sección, se presentan las gráficas de evolución de pérdida y precisión tanto en los datos de

entrenamiento como en los de validación, lo cual es esencial para identificar problemas como sobreajuste, subajuste y estabilidad del entrenamiento.

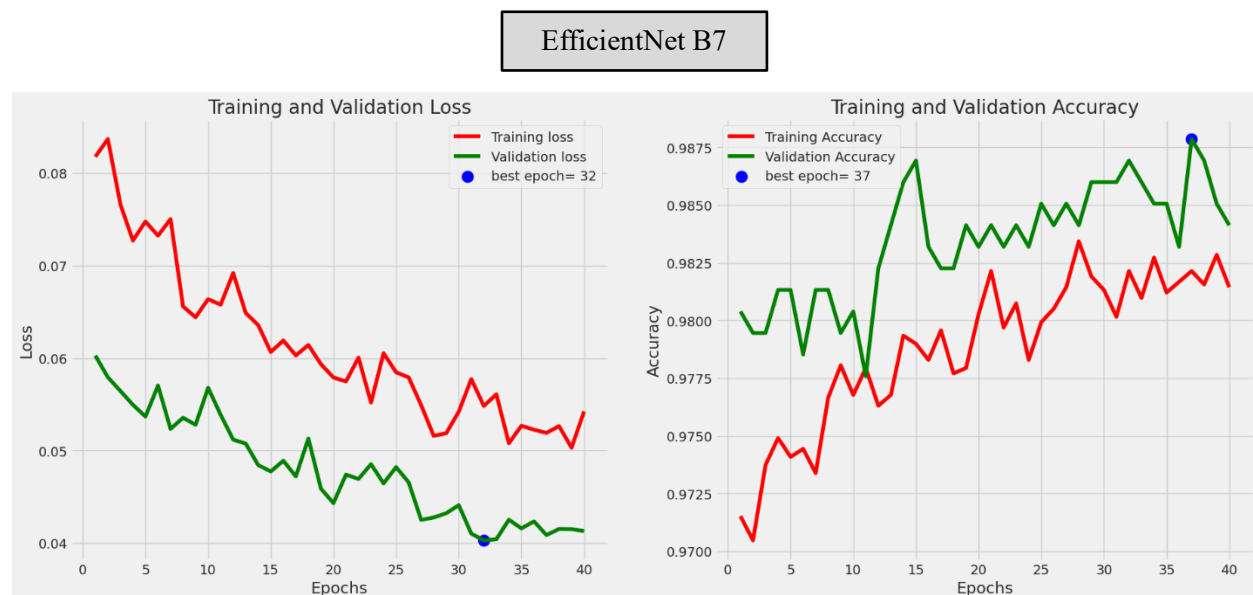


Figura 6. Gráfica de pérdida y precisión de la arquitectura EfficientNetB7

Nota: Se presentan los resultados del proceso de entrenamiento el modelo durante las 40 épocas.

En la Figura 6, se observa que la pérdida de entrenamiento (curva roja) disminuye de forma constante desde aproximadamente 0.08 hasta cerca de 0.05 al término de 40 épocas, lo que sugiere un aprendizaje efectivo del modelo. La pérdida de validación (curva verde) sigue una tendencia similar, estabilizándose alrededor de 0.04 hacia el final del entrenamiento. La mejor época para la pérdida de validación ocurre en la época 32, después de la cual el modelo muestra indicios de sobreajuste, ya que la pérdida de validación no mejora significativamente mientras que la pérdida de entrenamiento sigue disminuyendo. Este comportamiento refleja que el modelo está aprendiendo características específicas de los datos de entrenamiento, lo cual podría limitar su capacidad de generalización. Este patrón es consistente con lo observado por **Kermany et al., (2018)**, quienes destacaron que arquitecturas complejas como EfficientNet tienden a sobreajustar cuando se entrenan con conjuntos de datos limitados o bien estructurados.

En cuanto a la precisión, la gráfica muestra un aumento progresivo tanto en los datos de entrenamiento como en los de validación. El modelo alcanza una precisión máxima de validación del 98.75% en la época 37, lo que indica que EfficientNetB7 es capaz de aprender características discriminativas de las imágenes MRI del conjunto de datos OASIS. Sin embargo, es evidente que la precisión de validación experimenta

algunas oscilaciones, especialmente en las primeras épocas, lo que podría deberse a variaciones inherentes en el proceso de entrenamiento o al tamaño del conjunto de validación. Dicho comportamiento se alinea con las conclusiones de **Jack et al., (2008)**, quienes señalaron que el uso de datos preprocesados y homogéneos, como los de OASIS, puede limitar la capacidad de los modelos para generalizar.

El procedimiento observado en ResNet50 se alinea con las características de su arquitectura, que utiliza bloques residuales para mitigar problemas como la degradación del gradiente y facilita el aprendizaje en redes profundas. Como lo destacaron **He et al., (2015)**, los bloques residuales permiten a ResNet50 extraer características jerárquicas con mayor precisión, lo que es particularmente útil en tareas de clasificación multiclase. Sin embargo, esta capacidad también puede hacer que el modelo sea más sensible a conjuntos de datos preprocesados y homogéneos, como OASIS, lo que explica las oscilaciones en las métricas de validación observadas en este estudio.

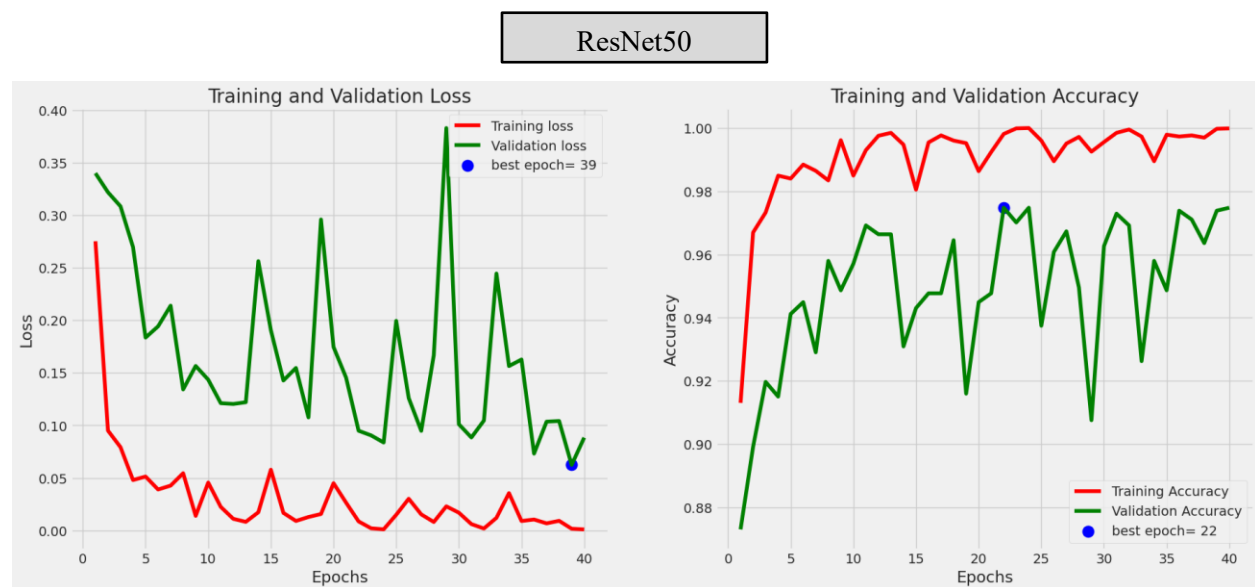


Figura 7. Gráfica de pérdida y precisión de la arquitectura ResNet50

Nota: Resultados del proceso de entrenamiento del modelo durante las 40 épocas.

La figura 7 muestra que la pérdida de entrenamiento en ResNet50 disminuye consistentemente desde 0.27 hasta aproximadamente 0.01, lo que representa un aprendizaje efectivo. Sin embargo, la pérdida de validación presenta fluctuaciones notables entre las épocas 10 y 30, lo que podría estar relacionado con la sensibilidad del modelo hacia características específicas del conjunto de validación. La pérdida de validación se estabiliza alrededor de 0.07 hacia el final del entrenamiento, con la mejor época registrada en

la 39. Estas variaciones podrían estar asociadas con características específicas del conjunto de datos de validación y una posible sensibilidad del modelo hacia los datos de entrenamiento.

La precisión del entrenamiento en ResNet50 aumenta rápidamente, alcanzando casi el 100% al final del proceso, mientras que la precisión de validación logra un máximo del 98.5% en la época 22. Sin embargo, las fluctuaciones en la validación subrayan la necesidad de estrategias como la implementación de técnicas avanzadas de aumento de datos o ajustes más refinados de los hiperparámetros para reducir el impacto del sobreajuste.

Este comportamiento es coherente con las características de la arquitectura ResNet50, que utiliza bloques residuales para facilitar el aprendizaje en redes profundas y permitir la extracción de características jerárquicas. Sin embargo, este diseño también puede ser más sensible a datos preprocesados y homogéneos, como los de OASIS. Estudios previos han señalado que ResNet50 puede ser propenso a inestabilidad en los datos de validación, especialmente cuando los conjuntos de datos son pequeños o presentan una representación limitada de ciertas clases (**He et al., 2015**).

ResNet50 y EfficientNetB7 muestran resultados competitivos en comparación con estudios similares. **Liu et al., (2018)**, reportaron precisiones entre el 92% y el 95% al utilizar redes convolucionales profundas para el diagnóstico del Alzheimer, lo que coloca a ResNet50 en una posición destacada en términos de capacidad discriminativa. De igual forma, **Islam & Zhang, (2018)** reportaron precisiones de validación de hasta el 95% utilizando ResNet en tareas similares de clasificación del Alzheimer, mientras que este modelo alcanza una precisión máxima del 98.5%, lo que demuestra su efectividad. Sin embargo, estudios como el de **Alsubaie et al., (2024)** sugieren que técnicas de regularización más agresivas y un mayor enfoque en el aumento de datos podrían reducir las oscilaciones observadas y mejorar la estabilidad del modelo.

3.1.2 Evaluación de rendimiento e interpretabilidad de las arquitecturas EfficientNet B7 y ResNet50

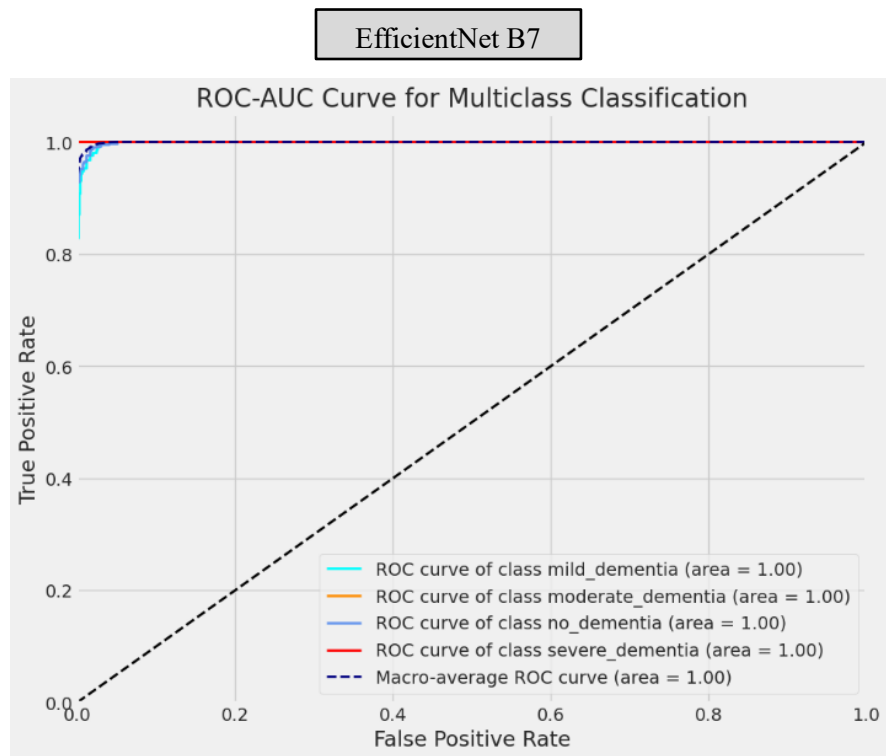
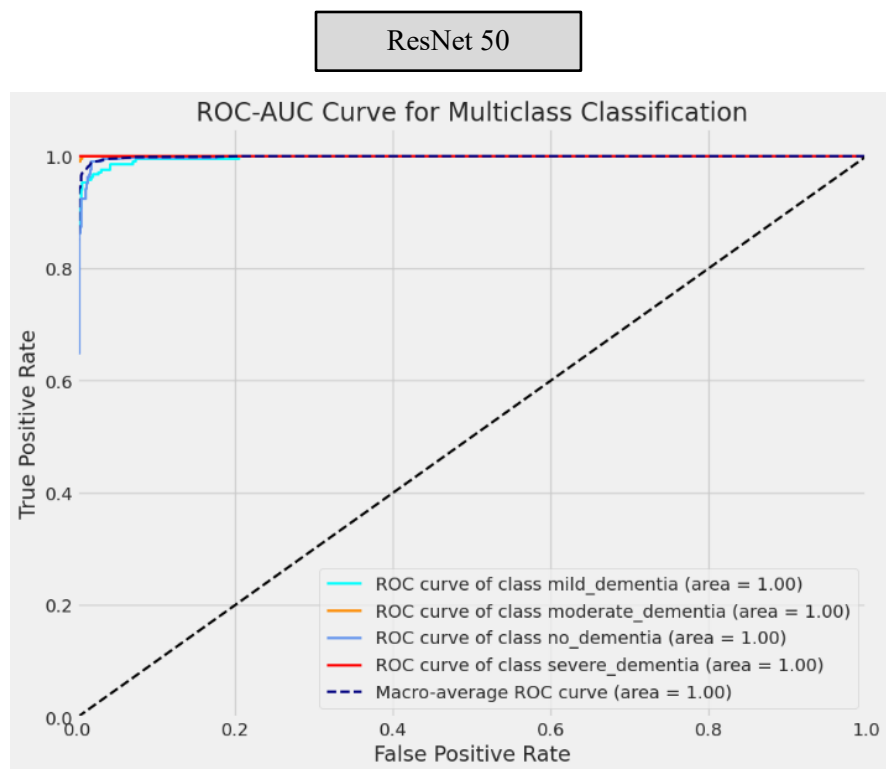


Figura 8. Curvas ROC-AUC comparativas de las arquitecturas EfficientNetB7 y ResNet50 para clasificación multiclase.

Nota: En la figura se presenta las curvas ROC-AUC para las arquitecturas EfficientNetB7 y ResNet50 en la clasificación multiclase de la enfermedad de Alzheimer. Cada clase incluye "sin demencia," "demencia leve," "demencia moderada" y "demencia severa." Ambas arquitecturas muestran un rendimiento óptimo, alcanzando un área bajo la curva (AUC) de 1.0 en todas las categorías, lo que indica una capacidad de clasificación altamente precisa.

Las dos arquitecturas alcanzan un AUC de 1.0 en todas las categorías, lo que indica una capacidad de clasificación altamente precisa. Este rendimiento óptimo es consistente con estudios como los de **Ntampakis et al., 2024**, quienes destacaron la efectividad de estas arquitecturas en tareas de clasificación basadas en datos de OASIS con una precisión total de las dos arquitecturas de 94.12%.

EfficientNetB7, con su diseño optimizado basado en escalabilidad compuesta, equilibra eficientemente el rendimiento y los requisitos computacionales. Esto lo hace especialmente adecuado para aplicaciones donde la eficiencia computacional es una prioridad (**Tan & Le, 2020**). ResNet50, por otro lado, se beneficia de sus bloques residuales, que mitigan problemas de degradación en redes profundas, permitiendo un aprendizaje más estable y eficiente incluso en conjuntos de datos más complejos (**He et al., 2015**).

Si bien los resultados de las curvas ROC-AUC son alentadores, es necesario reconocer que las métricas perfectas pueden ser indicativas de características inherentes al conjunto de datos y no necesariamente de una capacidad de generalización adecuada a datos clínicos no vistos. Las bases de datos OASIS y ADNI que fueron utilizadas, son reconocidas por su calidad, pero no representan completamente la diversidad global de pacientes con Alzheimer (**Mahmud Joy et al., 2025; Ntampakis et al., 2024**). Esta limitación puede afectar la aplicabilidad de los modelos en poblaciones heterogéneas. Además, aunque EfficientNetB7 y ResNet50 demostraron un rendimiento ideal en validación, su implementación práctica en entornos clínicos requiere más evaluaciones en datos reales, que a menudo incluyen ruido y artefactos no presentes en conjuntos de datos controlados (**B. & Kalirajan, 2024**).

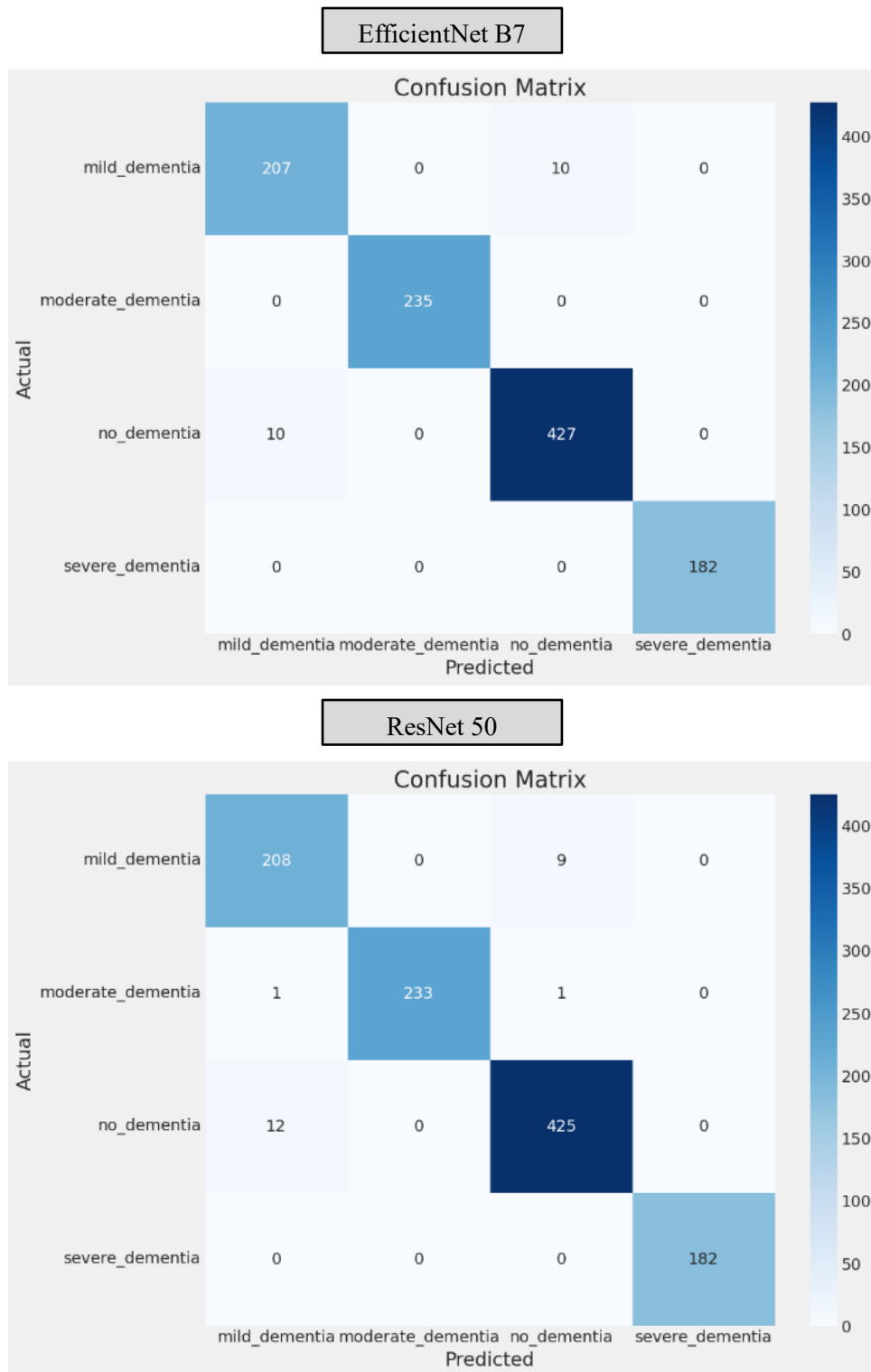


Figura 9. Matrices de confusión en la clasificación multiclase del alzheimer

Nota: Se muestran las matrices de confusión obtenidas para las arquitecturas EfficientNetB7 y ResNet50 en la clasificación multiclase de la enfermedad de Alzheimer. Las arquitecturas demuestran una alta

capacidad de clasificación correcta, con errores mínimos concentrados principalmente en casos límite entre clases adyacentes. Estas matrices ilustran las fortalezas y debilidades específicas de cada modelo en términos de precisión y manejo de transiciones entre categorías.

Se obtuvieron también los resultados de la matriz de confusión de cada una de las arquitecturas, las cuales se pueden observar en la Figura 9. En el caso de EfficientNet B7, la matriz de confusión revela que el modelo identificó correctamente 207 casos de demencia leve, 235 de demencia moderada, 427 de pacientes sin demencia y 182 de demencia severa. Sin embargo, se observan 10 casos de demencia leve clasificados erróneamente como "sin demencia." Esta confusión indica que EfficientNet B7, aunque eficaz en general, tiende a ser más conservador en la clasificación de casos límite, particularmente en la transición de demencia leve a otras categorías. Esto puede deberse a su diseño optimizado para captar características específicas, pero que podría presentar limitaciones al enfrentar variaciones sutiles en los datos **(Tan & Le, 2020)**.

Por otro lado, la matriz de confusión de ResNet50 muestra una ligera diferencia en el comportamiento del modelo. ResNet50 clasificó correctamente 208 casos de demencia leve, 233 de demencia moderada, 425 de pacientes sin demencia y 182 de demencia severa. Aquí, el modelo presentó tres errores distribuidos de manera más equitativa: un caso de demencia leve clasificado como moderada, uno de demencia moderada clasificado como leve y 12 de pacientes sin demencia clasificados erróneamente. Esto sugiere que ResNet50 maneja mejor la variación dentro de las clases, distribuyendo los errores de manera más uniforme, pero con una ligera penalización en la clase "sin demencia" **(He et al., 2015)**.

En términos de desempeño general, EfficientNet B7 se muestra más efectivo al manejar las clases "moderada" y "severa," mientras que ResNet50 destaca por su capacidad para manejar transiciones más sutiles entre categorías cercanas, como "leve" y "moderada." Estas diferencias reflejan las características intrínsecas de las arquitecturas: EfficientNet B7 se beneficia de su diseño eficiente para extraer patrones claros y bien definidos, mientras que ResNet50, con sus bloques residuales, es más robusto frente a variaciones en los datos.

Tabla 4. Desempeño de las arquitecturas basada en parámetros precisión, recall, f1-score y especificidad.

EfficientNet B7					
	Precisión	Recall	F1-score	Support	Especificidad
Mild_dementia	0,95	0,95	0,95	217	0,99
Moderate_dementia	1,00	1,00	1,00	235	1,00
No_dementia	0,98	0,98	0,98	437	0,98
Severe_dementia	1,00	1,00	1,00	182	1,00
Accuracy			0,98	1071	
Macro avg	0,98	0,98	0,98	1071	
Weighted avg	0,98	0,98	0,98	1071	
ResNet50					
	Precisión	Recall	F1-score	Support	Especificidad
Mild_dementia	0,94	0,96	0,95	217	0,98
Moderate_dementia	1,00	0,99	1,00	235	1,00
No_dementia	0,98	0,97	0,97	437	0,98
Severe_dementia	1,00	1,00	1,00	182	1,00
Accuracy			0,98	1071	
Macro avg	0,98	0,98	0,98	1071	
Weighted avg	0,98	0,98	0,98	1071	

Nota: Se presenta las métricas de evaluación que incluyen precisión, recall, f1-score y especificidad para cada clase, además de promedios macro y ponderados. Estos resultados reflejan el desempeño final de los modelos, permiten un análisis comparativo entre ellos y correlación con estudios previos.

En la Tabla 4 se presentan los resultados de la predicción de ambos modelos, los cuales alcanzaron una precisión promedio del 98% y métricas de f1-score igualmente altas, lo que indica su capacidad para identificar correctamente las cuatro categorías clínicas.

EfficientNet B7, diseñado para optimizar la eficiencia computacional mediante una escalabilidad compuesta, demostró una precisión sobresaliente en las categorías más críticas, como "moderada" y "severa," con métricas perfectas de precisión, recall, y f1-score en ambas clases. Sin embargo, en la clase "leve," EfficientNet mostró un f1-score ligeramente inferior (95%), lo que podría deberse a la dificultad del modelo para captar las características más sutiles asociadas con etapas iniciales de la enfermedad.

Por otro lado, ResNet50, presentó un desempeño igualitario en la clase "leve," con un f1-score de 95%, sin embargo, mostró una ligera penalización en la clase "sin demencia," con un f1-score de 97% frente al 98% de EfficientNet B7. Esto podría deberse a una mayor sensibilidad hacia características de las otras clases, lo que lleva a confusiones ocasionales en esta categoría.

Farooq et al., (2017), utilizó arquitecturas CNN tradicionales para clasificar etapas del Alzheimer en MRI, se reportaron precisiones promedio cercanas al 90%. Esto se debe, en parte, al avance en las arquitecturas de redes neuronales utilizadas, ya que EfficientNet y ResNet integran diseños optimizados que maximizan la extracción de características discriminativas. Por ejemplo, EfficientNet B7 aprovecha su escalabilidad compuesta para equilibrar precisión y eficiencia computacional (**Tan & Le, 2020**), mientras que ResNet50 utiliza bloques residuales para mejorar el aprendizaje jerárquico y reducir errores en las transiciones entre clases (**He et al., 2015**).

En comparación con el trabajo de **Islam & Zhang, (2018)**, donde se aplicaron ResNets preentrenados a datos MRI con un rendimiento que oscilaba entre el 92% y el 95% en términos de f1-score, los resultados de este estudio también se posicionan como superiores. Esto puede atribuirse a que los datos de OASIS están altamente preprocesados y bien estructurados, lo que facilita el aprendizaje de patrones específicos en comparación con bases de datos más heterogéneas. Sin embargo, esta ventaja podría limitar la capacidad de generalización de los modelos a datos clínicos reales, un desafío identificado en múltiples investigaciones (**Jack et al., 2008; Ntampakis et al., 2024**).

Otro punto de comparación relevante es el estudio de **B. & Kalirajan, (2024)**, quien utilizó arquitecturas Swish-CNN para clasificar datos de OASIS y reportó precisiones cercanas al 96%. Aunque este desempeño es comparable, el uso de Swish-CNN implicó mayores requerimientos computacionales, lo que posiciona a EfficientNet y ResNet como alternativas más equilibradas en términos de precisión y eficiencia.

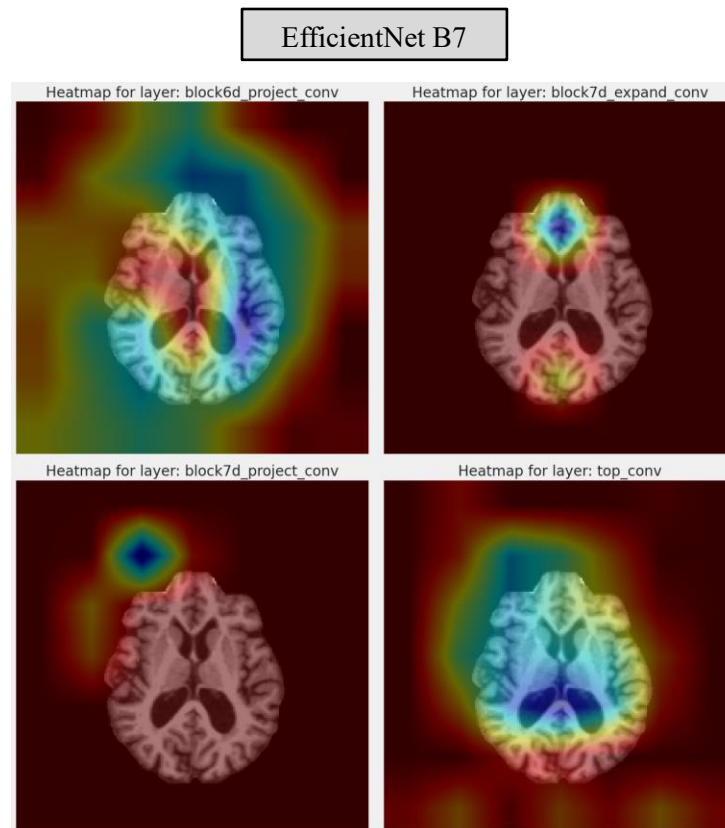
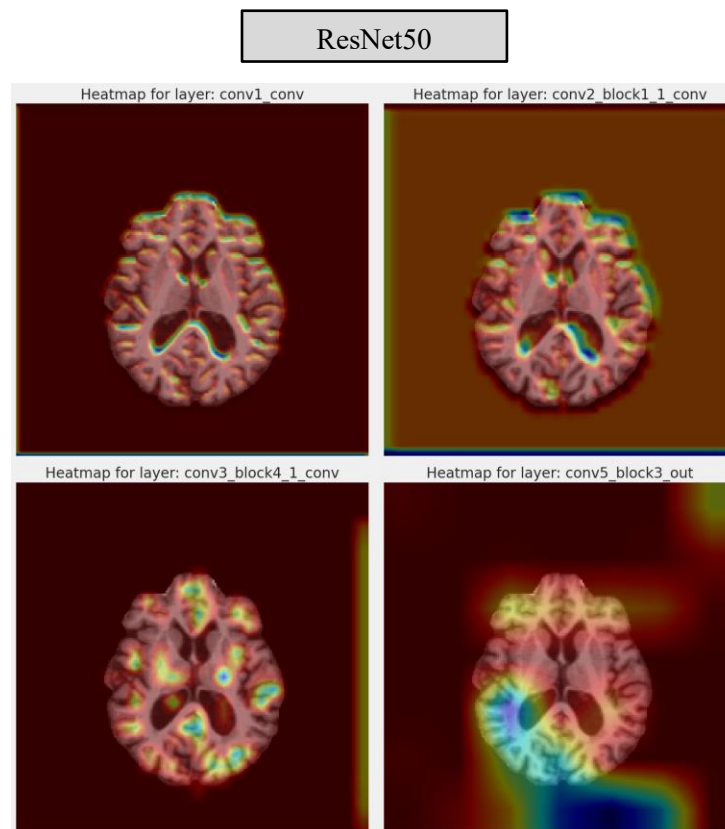


Figura 10. Comparación de los mapas generados por las dos arquitecturas EfficientNet y ResNet50

Nota: Estas imágenes representan los mapas de calor generados por las capas internas de las arquitecturas EfficientNet y ResNet50 al procesar las imágenes de MRI para la clasificación de la EA. Los colores en los mapas de calor reflejan la intensidad de las activaciones neuronales, donde los tonos cálidos (rojo y amarillo) indican las regiones del cerebro que tienen mayor influencia en las decisiones de los modelos, mientras que los tonos fríos (azul y verde) señalan áreas con menor relevancia. Las áreas oscuras o negras representan regiones con activaciones insignificantes o nulas.

Los mapas de calor generados por las arquitecturas EfficientNet y ResNet50 presentes en la Figura 10, muestran cómo estas redes neuronales profundas procesan y priorizan diferentes regiones de las imágenes de MRI para clasificar las etapas de la EA. En la primera imagen, correspondiente a EfficientNet, se observa cómo las activaciones están distribuidas en áreas relevantes del cerebro, particularmente en los lóbulos temporales y frontales, regiones comúnmente asociadas con los primeros signos de atrofia en el Alzheimer. Las capas internas, como *block6d_project_conv* y *block7d_expand_conv*, se centran en características específicas de estas áreas, mientras que las capas superiores, como *top_conv*, combinan estas características en un análisis más global. Este enfoque es característico de EfficientNet, que utiliza un diseño escalable para equilibrar profundidad, resolución y ancho, logrando una integración eficiente de patrones locales y globales (Tan & Le, 2020). Sin embargo, algunas regiones muestran activaciones más difusas, lo que sugiere que el modelo prioriza la cobertura amplia de patrones generales, algo que podría ser particularmente útil para identificar características tempranas de la enfermedad.

Por otro lado, los mapas de calor de ResNet50 muestran un enfoque más localizado en áreas específicas, como el hipocampo y otras regiones temporales, que son críticas en el diagnóstico del Alzheimer. Las primeras capas, como *conv1_conv*, detectan características básicas, como bordes y texturas, mientras que las capas más profundas, como *conv5_block3_out*, concentran sus activaciones en regiones estrechamente relacionadas con el deterioro cognitivo. Este enfoque localizado es una consecuencia directa del diseño de bloques residuales en ResNet50, que preservan la información importante de las capas iniciales y permiten un refinamiento progresivo de características complejas (He et al., 2015). Sin embargo, esta especificidad puede limitar la capacidad del modelo para capturar patrones globales en las imágenes, lo que lo diferencia de EfficientNet.

Al comparar ambas arquitecturas, se observa que EfficientNet tiende a integrar patrones globales con mayor amplitud, lo que podría ser ventajoso para identificar cambios sutiles y tempranos asociados con la EA. ResNet50, en cambio, muestra un enfoque más detallado y localizado, lo que podría hacerlo más adecuado para detectar características específicas de etapas intermedias o tardías de la enfermedad. Esta diferencia

en los enfoques también se refleja en otros trabajos. Por ejemplo, **Liu et al., (2018)** destacaron que las arquitecturas basadas en ResNet y suelen sobresalir en la extracción de características locales, mientras que arquitecturas optimizadas como EfficientNet pueden capturar mejor relaciones globales en imágenes médicas. Además, el trabajo de **Islam & Zhang, (2018)** reportó resultados similares al aplicar ResNet en la clasificación del Alzheimer, señalando que esta arquitectura es especialmente útil en tareas donde los patrones locales son predominantes.

Desde una perspectiva clínica, ambos modelos tienen fortalezas complementarias. EfficientNet, con su capacidad para analizar características globales, podría ser más efectivo en la detección temprana de la enfermedad, donde los signos son sutiles y distribuidos a lo largo del cerebro. Por otro lado, ResNet50, con su enfoque localizado, puede ser ideal para detectar características más avanzadas y específicas que aparecen en etapas intermedias o tardías del Alzheimer. Los mapas de calor generados refuerzan esta idea, ya que cada modelo prioriza diferentes regiones y patrones en las imágenes MRI, lo que podría ser aprovechado en sistemas de diagnóstico combinados que integren las fortalezas de ambas arquitecturas.

3.1.3 Evaluación de rendimiento de las arquitecturas EfficientNet B7 y ResNet50 frente a nuevos datos generados de ADNI y OASIS

Tabla 5. Aplicación de los modelos EfficientNet B7 y ResNet50 en datos nuevos ADNI

EfficientNet B7						
Total de imágenes				46		
Correctamente clasificados				21 (45,65%)		
Incorrectamente clasificados				25 (54,35%)		
	Precisión	Recall	F1-score	Support	Correctamente clasificados	Incorrectamente clasificados
MCI	0,42	0,81	0,55	16	13 (81,25%)	3 (18,75%)
CN	0,55	0,40	0,46	15	6 (40,00%)	9 (60,00%)
AD	0,50	0,13	0,21	15	2 (13,33%)	13 (86,67%)
Accuracy			0,46	46		
Macro avg	0,49	0,45	0,41	46		
Weighted avg	0,49	0,46	0,41	46		

ResNet50						
Total de imágenes					46	
Correctamente clasificados					18 (39,13%)	
Incorrectamente clasificados					28 (60,87%)	
	Precisión	Recall	F1-score	Support	Correctamente clasificados	Incorrectamente clasificados
MCI	0,38	0,94	0,54	16	15 (93,75%)	1 (6,25%)
CN	0,00	0,00	0,00	15	0 (0,00%)	15 (100,00%)
AD	0,50	0,20	0,29	15	3 (20,00%)	12 (80,00%)
Accuracy			0,39	46		
Macro avg	0,29	0,38	0,27	46		
Weighted avg	0,29	0,39	0,28	46		

Nota: En esta tabla se presentan los resultados obtenidos al evaluar las arquitecturas EfficientNet B7 y ResNet50 con 46 imágenes de MRI provenientes de la base de datos ADNI, un conjunto completamente nuevo y diferente al utilizado durante el entrenamiento. Los resultados reflejan que las imágenes de OASIS, al ser altamente preprocesadas, limitan la capacidad de los modelos para generalizar y adaptarse a las características propias de datos distintos como ADNI. Sin embargo, a pesar de estas diferencias y del desafío que representan las imágenes procesadas de manera diferente, los modelos lograron clasificar correctamente casi la mitad de las imágenes en el caso de EfficientNet B7, destacando su capacidad para identificar ciertos patrones comunes entre ambas bases.

Para complementar y verificar la estabilidad del modelo frente a nuevos datos, se realizó una prueba utilizando imágenes de la base de datos ADNI, dado que el modelo fue entrenado exclusivamente con imágenes MRI provenientes de OASIS. Esta prueba evidenció las limitaciones de generalización que enfrentan las arquitecturas EfficientNet B7 y ResNet50 en contextos clínicos más diversos. En la Tabla 5 se observa que EfficientNet B7 obtuvo una precisión del 45.65%, superando el 39.13% de ResNet50. Aunque este rendimiento sugiere una ligera ventaja de EfficientNet B7, ambos modelos tuvieron dificultades para adaptarse a las variaciones inherentes a las imágenes de ADNI, lo que subraya la necesidad de entrenar con datos más representativos y heterogéneos.

En la categoría de demencia leve (MCI), EfficientNet B7 logró un recall del 81%, clasificando correctamente 13 de las 16 imágenes, pero con una precisión del 42%, lo que refleja un número considerable de falsos positivos. ResNet50, por su parte, alcanzó un recall del 94%, con 15 imágenes correctamente clasificadas, aunque su precisión fue aún menor, de 38%. Si bien estos resultados indican un potencial para

identificar casos de MCI, el alto nivel de falsos positivos limita su aplicabilidad en entornos clínicos. Esto sugiere que, aunque estas arquitecturas pueden contribuir a la detección temprana del Alzheimer, necesitan mejoras significativas para reducir errores de clasificación y evitar diagnósticos equivocados.

En la categoría sin demencia (CN), EfficientNet B7 mostró un recall del 40%, clasificando correctamente 6 de las 15 imágenes. En contraste, ResNet50 no logró clasificar correctamente ninguna imagen de esta categoría, con un recall y una precisión del 0%. Este bajo desempeño en la identificación de pacientes cognitivamente normales destaca las limitaciones de ambos modelos para generalizar de manera adecuada frente a datos más diversos, como los presentes en ADNI. Estas dificultades podrían explicarse por diferencias en las características intrínsecas de las imágenes CN entre OASIS y ADNI, siendo OASIS una base de datos más homogénea y menos representativa de la diversidad clínica.

En la categoría de Alzheimer avanzado (AD), ambos modelos presentaron un rendimiento limitado. EfficientNet B7 alcanzó un recall de 13%, clasificando correctamente solo 2 de las 15 imágenes, mientras que ResNet50 logró un recall del 20%, identificando correctamente 3 imágenes. Ninguno de los modelos pudo capturar patrones consistentes en esta categoría, evidenciando desafíos significativos para generalizar frente a un conjunto de datos clínicamente más diverso. Las diferencias en el preprocesamiento, la calidad de las imágenes y las características clínicas de los participantes entre OASIS y ADNI podrían ser factores determinantes en este desempeño limitado.

Estos resultados son consistentes con trabajos anteriores. Por ejemplo, **Jack et al., (2008)** destacaron que las diferencias entre bases de datos, como ADNI y OASIS, impactan significativamente en el desempeño de los modelos debido a las variaciones en las características demográficas y clínicas de las imágenes. Por otro lado, **Islam & Zhang, (2018)** señalaron que las arquitecturas como ResNet suelen mostrar un desempeño limitado cuando se entrenan en bases de datos homogéneas y se prueban con datos más diversos. Asimismo, **Lu et al., (2018)** y **Liu et al., (2018)** sugirieron que el uso de estrategias como la transferencia de aprendizaje y el aumento de datos puede mejorar significativamente la capacidad de generalización, especialmente cuando se integran datos de múltiples bases durante el entrenamiento.

Tabla 6. Aplicación de los modelos EfficientNet B7 y ResNet50 en datos nuevos OASIS

EfficientNet B7						
Total de imágenes					300	
Correctamente clasificados					286 (95,33%)	
Incorrectamente clasificados					14 (4,67%)	
	Precisión	Recall	F1-score	Support	Correctamente clasificados	Incorrectamente clasificados
Mild_dementia	1,00	0,84	0,91	75	63 (84,00%)	12 (16,00%)
Moderate_dementia	0,91	1,00	0,96	75	75 (100,00%)	0 (0,00%)
No_dementia	0,91	0,97	0,94	75	73 (97,33%)	2 (2,67%)
Severe_dementia	1,00	1,00	1,00	75	75 (100,00%)	0 (0,00%)
Accuracy			0,95	300		
Macro avg	0,96	0,95	0,95	300		
Weighted avg	0,96	0,95	0,95	300		
ResNet50						
Total de imágenes					300	
Correctamente clasificados					225 (75,00%)	
Incorrectamente clasificados					75 (25,00%)	
	Precisión	Recall	F1-score	Support	Correctamente clasificados	Incorrectamente clasificados
Mild_dementia	0,82	0,48	0,61	75	36 (48,00%)	39(52,00%)
Moderate_dementia	0,56	1,00	0,72	75	75 (100,00%)	0 (0,00%)
No_dementia	0,85	0,55	0,67	75	41 (54,67%)	34 (45,33%)
Severe_dementia	0,97	0,97	0,97	75	73 (97,33%)	2 (2,67%)
Accuracy			0,75	300		
Macro avg	0,80	0,75	0,74	300		
Weighted avg	0,80	0,75	0,74	300		

Nota: En las tablas se presentan los resultados obtenidos al evaluarlos con un conjunto de datos generado a partir de imágenes modificadas de OASIS utilizando data augmentation con Keras. EfficientNet B7 destaca con un 95.33% de imágenes correctamente clasificadas, mientras que ResNet50 logró un 75%. Esto resalta la capacidad superior de EfficientNet para generalizar y adaptarse a las modificaciones en los datos, especialmente en categorías como demencia leve y sin demencia, donde ResNet50 presentó mayores limitaciones.

En esta evaluación, las imágenes a probar fueron la de OASIS, pero generadas artificialmente por medio de Keras, aquí se las modificó levemente para que el conjunto de datos sea completamente nuevo. Los resultados obtenidos visualizados en la Tabla 6, destacan el contraste entre ambas arquitecturas en cuanto a su capacidad para manejar datos modificados, lo que ofrece una perspectiva importante sobre su robustez frente a variaciones no vistas previamente en el entrenamiento. EfficientNet B7 obtuvo un desempeño significativamente superior, clasificando correctamente el 95.33% de las imágenes, mientras que ResNet50 logró un 75% de precisión general. Estos resultados reflejan la capacidad de EfficientNet para manejar mejor las variaciones introducidas en las imágenes modificadas, lo que puede atribuirse a su diseño escalable que optimiza la relación entre profundidad, ancho y resolución. Por el contrario, ResNet50 mostró mayores limitaciones, con un 25% de las imágenes mal clasificadas, lo que podría ser indicativo de una menor capacidad para adaptarse a cambios en las características visuales de los datos.

EfficientNet B7 se destacó especialmente en la clasificación de la categoría demencia leve, logrando un recall del 84%, con 63 de las 75 imágenes correctamente clasificadas y una precisión perfecta del 100%. Este rendimiento demuestra que el modelo es capaz de identificar con alta precisión casos de demencia leve, incluso en un conjunto de datos alterado, lo cual es particularmente relevante dado que la detección temprana de la enfermedad de Alzheimer puede ser crucial para su manejo clínico. En cambio, ResNet50 mostró un recall limitado de solo 48%, clasificando correctamente menos de la mitad de las imágenes y generando un número significativo de falsos negativos (52%). Lo cual resaltan la superioridad de EfficientNet en la detección de características sutiles asociadas con etapas tempranas de la enfermedad.

En la categoría Moderate Dementia, ambos modelos lograron identificar correctamente todas las imágenes, alcanzando un recall del 100%. Sin embargo, la precisión de EfficientNet B7 fue del 91%, lo que indica un menor número de falsos positivos en comparación con ResNet50, cuya precisión fue de solo 56%.

En la categoría No Dementia, EfficientNet B7 obtuvo un recall de 97.33%, clasificando correctamente 73 de las 75 imágenes, mientras que ResNet50 logró un 55%. Esto indica que ResNet50 tuvo dificultades significativas para identificar correctamente patrones normales en las imágenes modificadas, lo que resultó en una alta tasa de falsos negativos (45.33%). Por el contrario, el desempeño de EfficientNet en esta categoría refuerza su capacidad para adaptarse a las variaciones introducidas en los datos, proporcionando un rendimiento más consistente y confiable.

En la categoría Severe Dementia, ambos modelos mostraron resultados sobresalientes, con un recall del 100%. EfficientNet B7 y ResNet50 clasificaron correctamente todas las imágenes de esta clase, lo que

sugiere que las características asociadas con casos avanzados de Alzheimer son lo suficientemente distintivas como para ser detectadas incluso en un conjunto de datos modificado. Sin embargo, la precisión de EfficientNet fue ligeramente superior al alcanzar el 100%, mientras que ResNet50 quedó en un 97%, lo que refuerza la consistencia de EfficientNet en términos de clasificación general.

Los resultados obtenidos con las arquitecturas EfficientNet B7 y ResNet50 son consistentes con investigaciones recientes que exploran el uso de redes neuronales en la clasificación y detección de Alzheimer. Por ejemplo, el trabajo que destaca la efectividad de las redes neuronales convolucionales optimizadas, como las Swish-CNN, para clasificar múltiples etapas del Alzheimer mediante imágenes de resonancia magnética. Este estudio subraya que las configuraciones arquitectónicas modernas no solo mejoran la precisión, sino que también ofrecen una mayor flexibilidad para manejar datos clínicos heterogéneos y preprocesados. Asimismo, EfficientNet B7, con su diseño escalable, presenta resultados prometedores en escenarios controlados y modificados, lo que refuerza la idea de que los modelos avanzados tienen el potencial de superar limitaciones de generalización observadas en arquitecturas más tradicionales como ResNet50.

Además, otros trabajos como el de **Islam & Zhang, (2018)**, que destacan cómo los enfoques basados en CNN pueden ser limitados cuando se enfrentan a conjuntos de datos con mayor variabilidad, un hallazgo que también se refleja en el desempeño más bajo de ResNet50, particularmente en categorías como sin demencia. En contraste, los modelos más recientes, como EfficientNet, tienden a manejar mejor las diferencias entre conjuntos de datos, algo que también fue observado en los experimentos con imágenes modificadas de OASIS. Asimismo, la investigación de **Lu et al., (2018)** subraya la importancia del aumento de datos para mejorar la robustez de los modelos, algo que podría haber contribuido al rendimiento superior de EfficientNet B7 en escenarios de datos más desafiantes.

El trabajo de **B. & Kalirajan, (2024)** también destaca la importancia de la arquitectura de este modelo en la clasificación de etapas tempranas, como el deterioro cognitivo leve, un área en la que EfficientNet B7 demostró un rendimiento notorio con datos modificados. Sin embargo, el rendimiento limitado de ResNet50 en estas categorías sugiere que su capacidad para extraer características relevantes en etapas tempranas podría beneficiarse de configuraciones arquitectónicas más avanzadas o estrategias de preentrenamiento más especializadas. Este contraste enfatiza la importancia de adoptar modelos modernos para aplicaciones clínicas donde la precisión en la detección temprana es fundamental.

La aplicabilidad de los modelos basados en aprendizaje profundo para la detección temprana del Alzheimer, como EfficientNet B7 y ResNet50, representa un avance significativo en el campo del diagnóstico médico asistido por IA. Estos modelos han demostrado su capacidad para identificar patrones sutiles en imágenes de MRI, especialmente en etapas tempranas como el deterioro cognitivo leve, una fase crítica para la intervención temprana. No obstante, el desempeño de estos modelos depende en gran medida de la calidad y diversidad de los datos con los que se entrenan, lo que plantea desafíos para su implementación en escenarios clínicos reales **(Liu et al., 2018; Lu et al., 2018)**.

En este estudio, EfficientNet B7 destacó por su robustez al clasificar correctamente el 95.33% de las imágenes modificadas de OASIS y el 45.65% de las imágenes de ADNI, mientras que ResNet50 mostró un rendimiento más limitado. Estas diferencias resaltan la importancia de las arquitecturas modernas en la extracción de características relevantes, pero también evidencian la dependencia de los modelos en datos preprocesados y controlados, como los de OASIS y ADNI, que no siempre reflejan las condiciones heterogéneas y desafiantes de las imágenes obtenidas en los hospitales. En entornos clínicos reales, las imágenes pueden presentar variabilidad en calidad, resolución y técnicas de adquisición, lo que podría afectar el rendimiento de los modelos entrenados exclusivamente con datos homogéneos **(Jack et al., 2018; Moradi et al., 2015)**.

El uso de bases de datos controladas como OASIS y ADNI ha sido fundamental para el desarrollo y validación de modelos de aprendizaje profundo. Estas bases proporcionan datos uniformes y de alta calidad, ideales para entrenar modelos y evaluar su rendimiento en condiciones óptimas. Sin embargo, su capacidad para generalizar conjuntos de datos más diversos, como los obtenidos en hospitales, sigue siendo un desafío. Por ejemplo, las imágenes en ADNI suelen ser preprocesadas para garantizar consistencia, mientras que las imágenes hospitalarias pueden incluir ruido, artefactos y variabilidad en los protocolos de adquisición. Esto subraya la necesidad de integrar datos más diversos durante el entrenamiento para garantizar que los modelos sean aplicables en contextos clínicos reales **(Jo et al., 2019)**.

La incorporación de técnicas avanzadas de aprendizaje profundo, como el aumento de datos, la transferencia de aprendizaje y el entrenamiento multitarea, puede ayudar a superar las limitaciones relacionadas con la generalización de los modelos a datos más diversos. Estas estrategias permiten a los modelos aprender patrones más generales y robustos, mejorando su capacidad para manejar datos heterogéneos. Por ejemplo, el uso de enfoques multimodales que combinen imágenes MRI con otros biomarcadores, como análisis de líquido cefalorraquídeo o datos genéticos, puede proporcionar una visión

más completa de la progresión de la enfermedad, aumentando así la precisión del diagnóstico temprano **(Jack et al., 2018; Tan & Le, 2020)**.

Para finalizar, los resultados obtenidos indican que EfficientNet B7 es una herramienta prometedora para la detección temprana del Alzheimer, especialmente en etapas iniciales como el deterioro cognitivo leve. Esto es crucial, ya que la identificación temprana permite la intervención en una etapa donde los tratamientos pueden ser más efectivos, retrasando la progresión de la enfermedad **(Scheltens et al., 2021)**. Sin embargo, la aplicación clínica de estos modelos requiere una validación exhaustiva con datos más representativos obtenidos en escenarios reales, imágenes obtenidas directamente en hospitales, que suelen tener mayor variabilidad en términos de calidad y adquisición.

CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- La creación del conjunto de datos fue un paso fundamental, ya que permitió integrar y preparar imágenes de MRI de las bases de datos OASIS y ADNI. Estas bases son ampliamente reconocidas en la investigación neurológica por su calidad y diversidad, proporcionando una plataforma sólida para entrenar y evaluar modelos de aprendizaje profundo. Durante este proceso, se enfrentaron desafíos relacionados con la estandarización de los formatos de imagen, la variabilidad en las técnicas de adquisición y las diferencias en el procesamiento previo de los datos entre ambas. A pesar de ello, el resultado fue un conjunto de datos consistente y representativo que incluyó pacientes con diferentes etapas de Alzheimer, los cuales son: sin demencia, demencia leve, moderada y severa.
- El desarrollo de los modelos EfficientNet B7 y ResNet50 marcó un avance significativo en la implementación de técnicas de inteligencia artificial para la detección del Alzheimer. Estas arquitecturas modernas, optimizadas para la extracción de características relevantes en imágenes complejas como las MRI, permitieron explorar cómo las redes neuronales convolucionales pueden identificar patrones tempranos asociados con la enfermedad. EfficientNet B7, con su diseño escalable, mostró un desempeño superior en términos de precisión y generalización, lo que refleja su capacidad para capturar características distintivas en diferentes etapas de la enfermedad. Por otro lado, ResNet50, aunque eficaz en escenarios específicos, demostró limitaciones en su capacidad de generalización, especialmente con datos modificados o no vistos previamente. Este desarrollo no solo valida el uso de modelos avanzados en el ámbito médico, sino que también resalta la necesidad de seguir optimizando las arquitecturas y los procesos de entrenamiento para abordar de manera efectiva las particularidades de los datos clínicos.
- La evaluación del rendimiento de los modelos entrenados arrojó resultados que destacan tanto los avances como las áreas de oportunidad en la aplicación de estas tecnologías. EfficientNet B7 logró una precisión notable al clasificar imágenes provenientes de OASIS, con tasas de hasta el 95.33% en datos modificados y un rendimiento prometedor al enfrentarse a imágenes de ADNI, aunque con una menor precisión debido a la variabilidad de estas últimas. En contraste, ResNet50 presentó un

desempeño más limitado, particularmente en la identificación de etapas tempranas como el deterioro cognitivo leve. Estas diferencias refuerzan la importancia de elegir arquitecturas avanzadas y robustas para maximizar el potencial de los modelos en escenarios clínicos. Además, las pruebas realizadas con datos no vistos previamente subrayan la necesidad de utilizar conjuntos de datos más diversos y realistas durante el entrenamiento, asegurando así que los modelos puedan generalizar de manera efectiva en entornos clínicos reales.

4.2 Recomendaciones

- Una de las principales limitaciones observadas en este estudio fue la dependencia de datos controlados y preprocesados, como los provenientes de las bases de datos OASIS y ADNI. Para futuros trabajos, se recomienda integrar imágenes obtenidas directamente de entornos clínicos reales, que incluyan variabilidad en calidad, resolución y técnicas de adquisición. Además, sería valioso considerar datos multimodales, como biomarcadores genéticos, análisis de líquido cefalorraquídeo o imágenes funcionales, para mejorar la capacidad de los modelos de capturar características complejas asociadas con las diferentes etapas del Alzheimer. Esto no solo aumentará la generalización de los modelos, sino que también los hará más robustos frente a datos heterogéneos.
- Si bien las arquitecturas utilizadas, EfficientNet B7 y ResNet50, demostraron ser efectivas, su desempeño podría mejorarse utilizando modelos preentrenados en conjuntos de datos médicos más específicos o en grandes bases de datos de imágenes genéricas. La transferencia de aprendizaje permite que los modelos aprovechen representaciones previamente aprendidas, lo que puede acelerar el entrenamiento y mejorar la precisión en tareas específicas como la detección temprana del Alzheimer. Además, la inclusión de varias técnicas de ajuste fino en capas específicas del modelo podría optimizar aún más su capacidad de identificar patrones relacionados con la enfermedad en diferentes tipos de imágenes.
- Para garantizar la aplicabilidad de estos modelos en la práctica médica, futuros trabajos deben centrarse en diseñar e implementar sistemas que integren estas arquitecturas en los flujos de trabajo clínicos. Esto incluiría la validación de los modelos en tiempo real con imágenes provenientes de hospitales y clínicas, así como la incorporación de herramientas explicativas para que los profesionales médicos puedan interpretar los resultados generados por la IA. Además, sería útil realizar estudios de impacto que evalúen cómo el uso de estos modelos puede influir en la toma de decisiones clínicas, la detección temprana y el tratamiento del Alzheimer en escenarios del mundo real.

MATERIALES DE REFERENCIA

Referencias Bibliográficas

Alsubaie, M. G., Luo, S., & Shaukat, K. (2024). Alzheimer's Disease Detection Using Deep Learning on Neuroimaging: A Systematic Review. *Machine Learning and Knowledge Extraction*, 6(1), 464-505. <https://doi.org/10.3390/make6010024>

Alzheimer's Association. (2022). *¿Qué es el Alzheimer?* [Medica]. Alzheimer's Disease and Dementia. <https://alz.org/alzheimer-demencia/que-es-la-enfermedad-de-alzheimer>

Atri, A. (2019). Current and Future Treatments in Alzheimer's Disease. *Seminars in Neurology*, 39(02), 227-240. <https://doi.org/10.1055/s-0039-1678581>

B., A., & Kalirajan, K. (2024). An intelligent magnetic resonance imagining-based multistage Alzheimer's disease classification using swish-convolutional neural networks. *Medical & Biological Engineering & Computing*. <https://doi.org/10.1007/s11517-024-03237-2>

Basheera, S., & Ram, M. S. S. (2021). Deep learning based Alzheimer's disease early diagnosis using T2w segmented gray matter MRI. *International Journal of Imaging Systems and Technology*, 31(3), 1692-1710. <https://doi.org/10.1002/ima.22553>

Bateman, R. J., Xiong, C., Benzinger, T. L. S., Fagan, A. M., Goate, A., Fox, N. C., Marcus, D. S., Cairns, N. J., Xie, X., Blazey, T. M., Holtzman, D. M., Santacruz, A., Buckles, V., Oliver, A., Moulder, K., Aisen, P. S., Ghetti, B., Klunk, W. E., McDade, E., ... Morris, J. C. (2012). Clinical and Biomarker Changes in Dominantly Inherited Alzheimer's Disease. *New England Journal of Medicine*, 367(9), 795-804. <https://doi.org/10.1056/NEJMoa1202753>

Breijyeh, Z., & Karaman, R. (2020). Comprehensive Review on Alzheimer's Disease: Causes and Treatment. *Molecules*, 25(24), 5789. <https://doi.org/10.3390/molecules25245789>

Chang, C.-H., Lin, C.-H., & Lane, H.-Y. (2021). Machine Learning and Novel Biomarkers for the Diagnosis of Alzheimer's Disease. *International Journal of Molecular Sciences*, 22(5), 2761. <https://doi.org/10.3390/ijms22052761>

- Chen, C., Ou, X., Wang, J., Guo, W., & Ma, X. (2019). Radiomics-Based Machine Learning in Differentiation Between Glioblastoma and Metastatic Brain Tumors. *Frontiers in Oncology*, 9, 806. <https://doi.org/10.3389/fonc.2019.00806>
- Cummings, J. L., Tong, G., & Ballard, C. (2019). Treatment Combinations for Alzheimer's Disease: Current and Future Pharmacotherapy Options. *Journal of Alzheimer's Disease*, 67(3), 779-794. <https://doi.org/10.3233/JAD-180766>
- Fan, Z., Li, J., Zhang, L., Zhu, G., Li, P., Lu, X., Shen, P., Shah, S. A. A., Bennamoun, M., Hua, T., & Wei, W. (2021). U-net based analysis of MRI for Alzheimer's disease diagnosis. *Neural Computing and Applications*, 33(20), 13587-13599. <https://doi.org/10.1007/s00521-021-05983-y>
- Farooq, A., Anwar, S., Awais, M., & Rehman, S. (2017). A deep CNN based multi-class classification of Alzheimer's disease using MRI. *2017 IEEE International Conference on Imaging Systems and Techniques (IST)*, 1-6. <https://doi.org/10.1109/IST.2017.8261460>
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition* (No. arXiv:1512.03385). arXiv. <http://arxiv.org/abs/1512.03385>
- Herberg, E. (2023). *Lecture Notes: Neural Network Architectures* (No. arXiv:2304.05133). arXiv. <http://arxiv.org/abs/2304.05133>
- Islam, J., & Zhang, Y. (2018). Brain MRI analysis for Alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks. *Brain Informatics*, 5(2), 2. <https://doi.org/10.1186/s40708-018-0080-3>
- Jack, C. R., Bennett, D. A., Blennow, K., Carrillo, M. C., Dunn, B., Haeberlein, S. B., Holtzman, D. M., Jagust, W., Jessen, F., Karlawish, J., Liu, E., Molinuevo, J. L., Montine, T., Phelps, C., Rankin, K. P., Rowe, C. C., Scheltens, P., Siemers, E., Snyder, H. M., ... Silverberg, N. (2018). NIA-AA Research Framework: Toward a biological definition of Alzheimer's disease. *Alzheimer's & Dementia*, 14(4), 535-562. <https://doi.org/10.1016/j.jalz.2018.02.018>
- Jack, C. R., Bernstein, M. A., Fox, N. C., Thompson, P., Alexander, G., Harvey, D., Borowski, B., Britson, P. J., L. Whitwell, J., Ward, C., Dale, A. M., Felmlee, J. P., Gunter, J. L., Hill, D. L. G., Killiany, R., Schuff, N., Fox-Bosetti, S., Lin, C., Studholme, C., ... Weiner, M. W. (2008). The Alzheimer's disease neuroimaging initiative (ADNI): MRI methods. *Journal of Magnetic Resonance Imaging*, 27(4), 685-691. <https://doi.org/10.1002/jmri.21049>

- Jo, T., Nho, K., & Saykin, A. J. (2019). Deep Learning in Alzheimer's Disease: Diagnostic Classification and Prognostic Prediction Using Neuroimaging Data. *Frontiers in Aging Neuroscience*, 11, 220. <https://doi.org/10.3389/fnagi.2019.00220>
- Jude Chukwura Obi. (2023). A comparative study of several classification metrics and their performances on data. *World Journal of Advanced Engineering Technology and Sciences*, 8(1), 308-314. <https://doi.org/10.30574/wjaets.2023.8.1.0054>
- Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C. S., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., Dong, J., Prasadha, M. K., Pei, J., Ting, M. Y. L., Zhu, J., Li, C., Hewett, S., Dong, J., Ziyar, I., ... Zhang, K. (2018). Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell*, 172(5), 1122-1131.e9. <https://doi.org/10.1016/j.cell.2018.02.010>
- Khan, S., Barve, K. H., & Kumar, M. S. (2020). Recent Advancements in Pathogenesis, Diagnostics and Treatment of Alzheimer's Disease. *Current Neuropharmacology*, 18(11), 1106-1125. <https://doi.org/10.2174/1570159X18666200528142429>
- Lin, E., Lin, C.-H., & Lane, H.-Y. (2021). Deep Learning with Neuroimaging and Genomics in Alzheimer's Disease. *International Journal of Molecular Sciences*, 22(15), 7911. <https://doi.org/10.3390/ijms22157911>
- Liu, M., Cheng, D., Wang, K., & Wang, Y. (2018). Multi-Modality Cascaded Convolutional Neural Networks for Alzheimer's Disease Diagnosis. *Neuroinformatics*, 16(3-4), 295-308. <https://doi.org/10.1007/s12021-018-9370-4>
- Lu, D., Popuri, K., Ding, G. W., Balachandar, R., & Beg, M. F. (2018). Multiscale deep neural network based analysis of FDG-PET images for the early diagnosis of Alzheimer's disease. *Medical Image Analysis*, 46, 26-34. <https://doi.org/10.1016/j.media.2018.02.002>
- Mahmud Joy, Md. A., Nasrin, S., Siddiqua, A., & Farid, D. Md. (2025). ViTAD: Leveraging modified vision transformer for Alzheimer's disease multi-stage classification from brain MRI scans. *Brain Research*, 1847, 149302. <https://doi.org/10.1016/j.brainres.2024.149302>
- Marcus, D. S., Wang, T. H., Parker, J., Csernansky, J. G., Morris, J. C., & Buckner, R. L. (2007). Open Access Series of Imaging Studies (OASIS): Cross-sectional MRI Data in Young, Middle Aged, Nondemented, and Demented Older Adults. *Journal of Cognitive Neuroscience*, 19(9), 1498-1507. <https://doi.org/10.1162/jocn.2007.19.9.1498>

- Moradi, E., Pepe, A., Gaser, C., Huttunen, H., & Tohka, J. (2015). Machine learning framework for early MRI-based Alzheimer's conversion prediction in MCI subjects. *NeuroImage*, 104, 398-412. <https://doi.org/10.1016/j.neuroimage.2014.10.002>
- National Institute on Aging. (2023, abril 5). *La enfermedad de Alzheimer* [Médica]. National Institute on Aging. <https://www.nia.nih.gov/espanol/enfermedad-alzheimer/enfermedad-alzheimer>
- Ntampakis, N., Diamantaras, K., Chouvarda, I., Argyriou, V., & Sarigiannidis, P. (2024). *Enhanced Deep Learning Methodologies and MRI Selection Techniques for Dementia Diagnosis in the Elderly Population* (No. arXiv:2407.17324). arXiv. <https://doi.org/10.48550/arXiv.2407.17324>
- Passeri, E., Elkhoury, K., Morsink, M., Broersen, K., Linder, M., Tamayol, A., Malaplate, C., Yen, F. T., & Arab-Tehrany, E. (2022). Alzheimer's Disease: Treatment Strategies and Their Limitations. *International Journal of Molecular Sciences*, 23(22), 13954. <https://doi.org/10.3390/ijms232213954>
- Puente-Castro, A., Fernandez-Blanco, E., Pazos, A., & Munteanu, C. R. (2020). Automatic assessment of Alzheimer's disease diagnosis based on deep learning techniques. *Computers in Biology and Medicine*, 120, 103764. <https://doi.org/10.1016/j.compbiomed.2020.103764>
- Qiu, S., Miller, M. I., Joshi, P. S., Lee, J. C., Xue, C., Ni, Y., Wang, Y., De Anda-Duran, I., Hwang, P. H., Cramer, J. A., Dwyer, B. C., Hao, H., Kaku, M. C., Kedar, S., Lee, P. H., Mian, A. Z., Murman, D. L., O'Shea, S., Paul, A. B., ... Kolachalama, V. B. (2022). Multimodal deep learning for Alzheimer's disease dementia assessment. *Nature Communications*, 13(1), 3404. <https://doi.org/10.1038/s41467-022-31037-5>
- Sadaiyandi, J., Arumugam, P., Sangaiah, A. K., & Zhang, C. (2023). Stratified Sampling-Based Deep Learning Approach to Increase Prediction Accuracy of Unbalanced Dataset. *Electronics*, 12(21), 4423. <https://doi.org/10.3390/electronics12214423>
- Saporta, A., Gui, X., Agrawal, A., Pareek, A., Truong, S. Q. H., Nguyen, C. D. T., Ngo, V.-D., Seekins, J., Blankenberg, F. G., Ng, A. Y., Lungren, M. P., & Rajpurkar, P. (2022). Benchmarking saliency methods for chest X-ray interpretation. *Nature Machine Intelligence*, 4(10), 867-878. <https://doi.org/10.1038/s42256-022-00536-x>
- Scheltens, P., De Strooper, B., Kivipelto, M., Holstege, H., Chételat, G., Teunissen, C. E., Cummings, J., & Van Der Flier, W. M. (2021). Alzheimer's disease. *The Lancet*, 397(10284), 1577-1590. [https://doi.org/10.1016/S0140-6736\(20\)32205-4](https://doi.org/10.1016/S0140-6736(20)32205-4)

- Schutte, K., Moindrot, O., Hérent, P., Schiratti, J.-B., & Jégou, S. (2021). *Using StyleGAN for Visual Interpretability of Deep Learning Models on Medical Images* (No. arXiv:2101.07563). arXiv. <http://arxiv.org/abs/2101.07563>
- Sidey-Gibbons, J. A. M., & Sidey-Gibbons, C. J. (2019). Machine learning in medicine: A practical introduction. *BMC Medical Research Methodology*, 19(1), 64. <https://doi.org/10.1186/s12874-019-0681-4>
- Silva, G. F. S., Fagundes, T. P., Teixeira, B. C., & Chiavegatto Filho, A. D. P. (2022). Machine Learning for Hypertension Prediction: A Systematic Review. *Current Hypertension Reports*, 24(11), 523-533. <https://doi.org/10.1007/s11906-022-01212-6>
- Tan, M., & Le, Q. V. (2020). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks* (No. arXiv:1905.11946). arXiv. <http://arxiv.org/abs/1905.11946>
- Targ, S., Almeida, D., & Lyman, K. (2016). *Resnet in Resnet: Generalizing Residual Architectures* (No. arXiv:1603.08029). arXiv. <http://arxiv.org/abs/1603.08029>
- Tiwari, S., Atluri, V., Kaushik, A., Yndart, A., & Nair, M. (2019). Alzheimer's disease: Pathogenesis, diagnostics, and therapeutics. *International Journal of Nanomedicine, Volume 14*, 5541-5554. <https://doi.org/10.2147/IJN.S200490>
- Warren, S. L., & Moustafa, A. A. (2023). Functional magnetic resonance imaging, deep learning, and Alzheimer's disease: A systematic review. *Journal of Neuroimaging: Official Journal of the American Society of Neuroimaging*, 33(1), 5-18. <https://doi.org/10.1111/jon.13063>
- Woodman, R. J., & Mangoni, A. A. (2023). A comprehensive review of machine learning algorithms and their application in geriatric medicine: Present and future. *Aging Clinical and Experimental Research*, 35(11), 2363-2397. <https://doi.org/10.1007/s40520-023-02552-2>
- World Health Organization. (2019). *Risk reduction of cognitive decline and dementia: WHO guidelines*. World Health Organization. <https://iris.who.int/handle/10665/312180>
- Xiao, M., Zhang, L., Shi, W., Liu, J., He, W., & Jiang, Z. (2021). A visualization method based on the Grad-CAM for medical image segmentation model. *2021 International Conference on Electronic Information Engineering and Computer Science (EIECS)*, 242-247. <https://doi.org/10.1109/EIECS53707.2021.9587953>
- Yang, Q., Li, X., Ding, X., Xu, F., & Ling, Z. (2022). Deep learning-based speech analysis for Alzheimer's disease detection: A literature review. *Alzheimer's Research & Therapy*, 14(1), 186. <https://doi.org/10.1186/s13195-022-01131-3>

Yang, S., & Berdine, G. (2023). Interpretable artificial intelligence (AI) – saliency maps. *The Southwest Respiratory and Critical Care Chronicles*, 11(48), 31-37. <https://doi.org/10.12746/swrccc.v11i48.1209>

Yousefzadeh, N., Tran, C., Ramirez-Zamora, A., Chen, J., Fang, R., & Thai, M. T. (2023). *LAVA: Granular Neuron-Level Explainable AI for Alzheimer's Disease Assessment from Fundus Images* (Versión 2). arXiv. <https://doi.org/10.48550/ARXIV.2302.03008>

Zhang, Y., Hong, D., McClement, D., Oladosu, O., Pridham, G., & Slaney, G. (2021). Grad-CAM helps interpret the deep learning models trained to classify multiple sclerosis types using clinical brain magnetic resonance imaging. *Journal of Neuroscience Methods*, 353, 109098. <https://doi.org/10.1016/j.jneumeth.2021.109098>

ANEXOS

Anexo 1.

Script para preprocesar las imágenes de la base de datos OASIS

```
import nibabel as nib

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

import os


# Ruta al archivo Excel con la información del estado de demencia
excel_path = "E:\\Tesis kk\\oasis_cross-sectional-5708aa0a98d82080.xlsx"

# Directorio raíz donde están las carpetas de cada paciente
root_dir = "E:\\Tesis kk\\Todas las imagenes"

# Directorio de salida para guardar las imágenes procesadas
output_dir = "E:\\Tesis kk" # Ajusta la ruta de salida


# Crear carpeta de salida si no existe
os.makedirs(output_dir, exist_ok=True)


# Leer el archivo Excel y asignar etiquetas basadas en CDR
df = pd.read_excel(excel_path, sheet_name='oasis_cross-sectional')
```

```

# Crear un diccionario de mapeo de CDR a etiquetas

cdr_to_label = {

    0.0: 'no_dementia',

    0.5: 'mild_dementia',

    1.0: 'moderate_dementia',

    2.0: 'severe_dementia'

}


# Agregar una columna de etiquetas basada en el CDR

df['Label'] = df['CDR'].map(cdr_to_label)


# Filtrar solo las filas con etiquetas válidas para evitar pacientes sin datos de demencia

df_labeled = df.dropna(subset=['Label'])


# Crear carpetas de salida para cada etiqueta si no existen

for label in set(df_labeled['Label']):

    os.makedirs(os.path.join(output_dir, label), exist_ok=True)


# Parámetro para filtrar imágenes casi vacías

brightness_threshold = 980000

# Número exacto de imágenes requeridas por paciente

required_slices = 35


# Contador de pacientes eliminados

```

```
removed_patients_count = 0
```

```
# Función para buscar el archivo brain.mgz recursivamente en subcarpetas y mostrar todas las rutas visitadas
```

```
def find_brain_mgz(start_dir):
```

```
    for root, dirs, files in os.walk(start_dir):
```

```
        print(f'Revisando: {root}')
```

```
        if 'brain.mgz' in files:
```

```
            return os.path.join(root, 'brain.mgz')
```

```
    return None
```

```
# Recorrer las carpetas de los discos y analizar las imágenes
```

```
for _, row in df_labeled.iterrows():
```

```
    patient_id = row['ID']
```

```
    label = row['Label']
```

```
# Buscar en todos los discos
```

```
found = False
```

```
for disc_folder in os.listdir(root_dir):
```

```
    disc_path = os.path.join(root_dir, disc_folder)
```

```
    if os.path.isdir(disc_path):
```

```
        # Recorrer todas las subcarpetas dentro del disco para encontrar la carpeta del paciente
```

```
        for subfolder in os.listdir(disc_path):
```

```

subfolder_path = os.path.join(disc_path, subfolder)

# Buscar la carpeta del paciente dentro de la subcarpeta actual
patient_folder = os.path.join(subfolder_path, patient_id)

if os.path.isdir(patient_folder):

    # Buscar el archivo brain.mgz dentro de las subcarpetas del paciente
    brain_mgz_path = find_brain_mgz(patient_folder)

    # Verificar si el archivo brain.mgz fue encontrado
    if brain_mgz_path:
        found = True
        break

# Si se ha encontrado el archivo, no seguir buscando en otros discos
if found:
    break

if not found:
    print(f"Archivo brain.mgz no encontrado para el paciente {patient_id}")
    continue

# Cargar la imagen brain.mgz
img = nib.load(brain_mgz_path)

```

```

data = img.get_fdata()

# Mostrar las dimensiones de los datos

print(f"Procesando {brain_mgz_path}, Dimensiones: {data.shape}")

# Lista para almacenar las imágenes válidas

valid_slices = []

# Extraer los cortes axiales y verificar

for i in range(data.shape[1]): # Itera sobre los cortes axiales (eje 1)

    slice_data = data[:, i, :]

    # Verificar si el corte contiene suficiente información

    if np.sum(slice_data) > brightness_threshold: # Verificar si la imagen no está vacía

        valid_slices.append((slice_data, i))

# Verificar si el número de imágenes válidas es suficiente

if len(valid_slices) < required_slices:

    # Incrementar contador de pacientes eliminados si no cumple con las características mínimas

    removed_patients_count += 1

    print(f"Paciente {patient_id} eliminado: {len(valid_slices)} imágenes válidas encontradas, se
requieren al menos {required_slices}.")

else:

    # Seleccionar los primeros 35 cortes si hay más

```

```

valid_slices = valid_slices[:required_slices]

# Guardar las imágenes

for slice_data, index in valid_slices:

    # Normalizar la imagen

    slice_data = (slice_data - np.min(slice_data)) / (np.max(slice_data) - np.min(slice_data)) * 255

    slice_data = slice_data.astype(np.uint8)

    # Guardar el corte axial como imagen PNG

    output_path = os.path.join(output_dir, label, f'{patient_id}_axial_slice_{index}.png')

    plt.imsave(output_path, slice_data.T, cmap='gray', origin='lower')

    print(f'Guardado: {output_path}')

# Mostrar el total de pacientes eliminados

print(f'Total de pacientes eliminados por no cumplir con el número mínimo de imágenes requeridas:
{removed_patients_count}')

#Transformación de las imágenes de BGR a RGB

import cv2

import os

# Directorio de entrada y salida

input_dir = "E:\\Tesis kk\\Imágenes etiquetadas\\mild_dementia" # Cambiar la ruta de las imágenes

output_dir = "E:\\Tesis kk\\Imágenes RGB\\mild_dementia" # Cambia a la ruta para guardar

```

```

# Iterar sobre todas las imágenes en el directorio de entrada

for img_name in os.listdir(input_dir):

    img_path = os.path.join(input_dir, img_name)


# Leer la imagen usando OpenCV

image = cv2.imread(img_path)


# Verificar si la imagen se cargó correctamente

if image is not None:

    # Convertir la imagen de BGR a RGB

    image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)


# Guardar la imagen convertida a RGB

output_path = os.path.join(output_dir, img_name)

cv2.imwrite(output_path, image_rgb)

print(f"Imagen guardada en RGB: {output_path}")

else:

    print(f"No se pudo cargar la imagen: {img_path}")


#Modificar tamaño de las imágenes a 224x224

import cv2

```

```

import os

# Directorio de entrada y salida

input_dir = "E:\\Tesis kk\\Imágenes RGB"

output_dir = "E:\\Tesis kk\\224x224"


# Dimensiones deseadas para las imágenes

target_size = (224, 224)


# Recorrer todas las carpetas y subcarpetas dentro del directorio de entrada

for root, dirs, files in os.walk(input_dir):

    for file in files:

        # Ruta completa del archivo de entrada

        img_path = os.path.join(root, file)


        # Leer la imagen usando OpenCV

        image = cv2.imread(img_path)


        # Verificar si la imagen se cargó correctamente

        if image is not None:

            # Redimensionar la imagen a 224x224 píxeles

            resized_image = cv2.resize(image, target_size)


            # Crear la ruta correspondiente en el directorio de salida manteniendo la estructura

```

```

relative_path = os.path.relpath(root, input_dir)

output_folder = os.path.join(output_dir, relative_path)

os.makedirs(output_folder, exist_ok=True)

# Guardar la imagen redimensionada en la misma estructura

output_path = os.path.join(output_folder, file)

cv2.imwrite(output_path, resized_image)

print(f"Imagen redimensionada y guardada: {output_path}")

else:

    print(f"No se pudo cargar la imagen: {img_path}")


#Generador de imágenes con Keras para el data augmentation

from tensorflow.keras.preprocessing.image import ImageDataGenerator, img_to_array, load_img

import os

import numpy as np

import shutil


# Directorios de las carpetas de moderate dementia y severe dementia

input_dir_moderate = "E:\\Tesis kk\\224x224\\moderate_dementia"

input_dir_severe = "E:\\Tesis kk\\224x224\\severe_dementia"


# Directorio donde se guardarán las imágenes aumentadas por paciente

output_dir_moderate = "E:\\Tesis kk\\224x224\\moderado"

```

```
output_dir_severe = "E:\\Tesis kk\\224x224\\severo"
```

```
# Definir una semilla para controlar la aleatoriedad
```

```
SEED = 123
```

```
# Crear un generador de imágenes con transformaciones mínimas
```

```
datagen = ImageDataGenerator(
```

```
    rotation_range=5,
```

```
    width_shift_range=0.02,
```

```
    height_shift_range=0.02,
```

```
    zoom_range=0.02,
```

```
    fill_mode='nearest'
```

```
)
```

```
# Función para aumentar imágenes agrupadas en sets de 35 con ID únicos
```

```
def augment_images_grouped(input_dir, output_dir, num_new_patients, slices_per_patient=35):
```

```
    # Crear directorios si no existen
```

```
    if not os.path.exists(output_dir):
```

```
        os.makedirs(output_dir)
```

```
    # Contar cuántas imágenes originales hay
```

```
    image_filenames = sorted(os.listdir(input_dir))
```

```
    # Crear nuevas imágenes por grupo de slices_per_patient
```

```

for patient_num in range(num_new_patients):

    # Crear un nuevo ID para el paciente

    new_patient_id = f'NEW_PATIENT_{patient_num + 1:03d}'

    patient_dir = os.path.join(output_dir, new_patient_id)


    # Crear un directorio para este nuevo paciente

    if not os.path.exists(patient_dir):

        os.makedirs(patient_dir)


    print(f'Generating images for {new_patient_id}')


    for i in range(slices_per_patient):

        img_index = (patient_num * slices_per_patient + i) % len(image_filenames)

        img_path = os.path.join(input_dir, image_filenames[img_index])

        img = load_img(img_path)

        img_array = img_to_array(img)

        img_array = np.expand_dims(img_array, axis=0)


        # Generar una imagen nueva por slice y guardarla

        img_name_prefix = f'{new_patient_id}_coronal_slice_{i:02d}'

        for batch in datagen.flow(img_array, batch_size=1, save_to_dir=patient_dir,
save_prefix=img_name_prefix, save_format='jpg', seed=SEED):

            break

```

```
# Aumentar las imágenes de moderate dementia, creando 50 nuevos "pacientes"

augment_images_grouped(input_dir_moderate, output_dir_moderate, num_new_patients=50)


# Aumentar las imágenes de severe dementia, creando 20 nuevos "pacientes"

augment_images_grouped(input_dir_severe, output_dir_severe, num_new_patients=50)
```

Anexo 2.

Script para preprocesar las imágenes de la base de datos ADNI

```
import numpy as np

import nibabel as nib

import matplotlib.pyplot as plt

import cv2

import os

import pandas as pd


# Función para cortar la imagen en un corte específico

def cortar_imagen_rm(volume_3d, plano, corte):

    """

    Cortar la imagen 3D en un corte específico del plano ('sagital', 'axial' o 'coronal')

    """

    if plano == "sagital":

        # Corte sagital (eje X)

        return volume_3d[corte, :, :]
```

```

elif plano == "coronal":

    # Corte coronal (eje Y)

    return volume_3d[:, corte, :]


elif plano == "axial":

    # Corte axial (eje Z)

    return volume_3d[:, :, corte]


else:

    raise ValueError("Plano no válido. Usar 'sagital', 'coronal' o 'axial'.")


# Cargar el archivo NIfTI

def load_nifti(nifti_file):

    nifti_data = nib.load(nifti_file)

    volume = nifti_data.get_fdata()

    return volume


# Asegurarse de que los datos sean del tipo adecuado (uint8)

imagen = np.uint8(np.interp(imagen, (imagen.min(), imagen.max()), (0, 255))) # Normalización a 0-255


# Función para cargar el archivo Excel y encontrar el grupo de cada sujeto

def cargar_excel(excel_file):

```

```

df = pd.read_excel(excel_file)

return df

# Función para encontrar el grupo del paciente según el ID
def obtener_grupo(df, id_sujeto):

    grupo = df[df['Subject'] == id_sujeto]['Group'].values

    if len(grupo) > 0:

        return grupo[0]

    else:

        return None

# Función para mostrar todos los cortes del plano sagital y elegir uno
def seleccionar_corte_sagital(volume_3d):

    # Mostrar todos los cortes sagitales y pedir al usuario que seleccione uno

    num_slices = volume_3d.shape[0]

    print(f"Se tienen {num_slices} cortes sagitales disponibles.")

    # Mostrar todos los cortes sagitales

    for i in range(num_slices):

        plt.imshow(volume_3d[i, :, :], cmap='gray', origin="lower")

        plt.title(f'Corte Sagital {i}')

        plt.axis('off')

        plt.show()

```

```

# Pedir al usuario que ingrese el número de corte

corte_seleccionado = int(input(f"Ingresa el número del corte sagital que deseas guardar (0-
{num_slices-1}): "))

# Validar que el corte seleccionado esté dentro del rango

if corte_seleccionado < 0 or corte_seleccionado >= num_slices:

    print("Corte no válido. Usando el corte central por defecto.")

    corte_seleccionado = num_slices // 2

return corte_seleccionado

# Función para procesar y guardar la imagen

def procesar_guardar_imagen(nifti_file, excel_file, id_sujeto, output_base_dir):

    # Cargar la imagen NIfTI

    volume = load_nifti(nifti_file)

    # Mostrar y seleccionar el corte sagital

    corte_seleccionado = seleccionar_corte_sagital(volume)

    # Obtener el corte sagital seleccionado

    imagen_sagital = cortar_imagen_rm(volume, "sagital", corte_seleccionado)

    # Cargar el Excel con las etiquetas de los pacientes

    df = cargar_excel(excel_file)

```

```

# Obtener el grupo del paciente

grupo = obtener_grupo(df, id_sujeto)

if grupo is None:

    print(f"No se encontró el grupo para el paciente {id_sujeto}. No se guardará la imagen.")

    return

# Crear la carpeta de salida según el grupo

output_dir = os.path.join(output_base_dir, grupo)

os.makedirs(output_dir, exist_ok=True)

# Guardar la imagen en la carpeta correspondiente

output_path = os.path.join(output_dir, f"{id_sujeto}_sagital_{corte_seleccionado}.jpg")

cv2.imwrite(output_path, imagen_rgb)

print(f"Imagen guardada en: {output_path}")

# Directorio

nifti_file = r"E:\Tesis kk\Todas las imagenes
ADNI\031_S_1066\MPR__GradWarp__B1_Correction__N3__Scaled\2006-12-
04_12_03_05.0\I90916\ADNI_031_S_1066_MR_MPR__GradWarp__B1_Correction__N3__Scaled_Br_
20080220103435137_S23507_I90916.nii"

excel_file = r"E:\Tesis kk\ADNI1_Annual_2_Yr_3T_11_16_2024.xlsx" # Ruta del archivo Excel con las
etiquetas de los pacientes

id_sujeto = "031_S_1066" # ID del sujeto, asegúrate de que coincida con el Excel

output_base_dir = r"E:\Tesis kk\Imagenes_Procesadas"

```

Procesar y guardar la imagen

procesar_guardar_imagen(nifti_file, excel_file, id_sujeto, output_base_dir)

Anexo 3.

Enlace hacia el script con sus resultados de la arquitectura ResNet50

[resNet50Final.html](#)

Anexo 4.

Enlace hacia el script con sus resultados de la arquitectura EfficientNet B7

[EfficientnetB7Final.html](#)