



UNIVERSIDAD DE MÁLAGA



Grado en Ingeniería de la Salud

Clasificación de Imágenes de Resonancia Magnética de Alzheimer mediante Aprendizaje Profundo

Classification of Alzheimer's Magnetic Resonance Imaging using Deep Learning

Realizado por
Marta Cuevas Rodríguez

Tutorizado por
Miguel Ángel Molina Cabello
Rafael Marcos Luque Baena

Departamento
Lenguajes y Ciencias de la Computación
UNIVERSIDAD DE MÁLAGA

MÁLAGA, JUNIO 2025



UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA DE LA SALUD

Clasificación de Imágenes de Resonancia Magnética de Alzheimer mediante Aprendizaje Profundo

Classification of Alzheimer's Magnetic Resonance Imaging using Deep Learning

Realizado por
Marta Cuevas Rodríguez

Tutorizado por
Miguel Ángel Molina Cabello
Rafael Marcos Luque Baena

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, JUNIO 2025

Fecha defensa: julio de 2025

Abstract

This work focuses on the development of an automatic classification system for structural magnetic resonance imaging (MRI) aimed at detecting Alzheimer's disease across different clinical stages (CN, MCI, and VMCI), using deep learning models. A subset of data derived from the public ADNI study, obtained via the Kaggle platform, was used. Several convolutional neural network architectures were evaluated, including well-known models such as ResNet18, AlexNet, and MobileNet, as well as a CNN proposed in the literature. Throughout the project, various preprocessing strategies were explored, including normalization techniques, slice selection per patient, data augmentation, and different optimizers. The results indicate that the combination of carefully selected axial slices, standard normalization, and the use of the Adam optimizer provides the best performance in terms of accuracy and F1-score. Moreover, the study highlights the importance of proper patient-wise data separation and a critical evaluation of publicly available models to prevent data leakage. Although the achieved performance is not optimal, this work lays the groundwork for future improvements and more in-depth analyses in this field.

Keywords: Alzheimer's disease, medical imaging, MRI, deep learning, classification, convolutional neural networks, ADNI.

Resumen

Este trabajo se centra en el desarrollo de un sistema de clasificación automática de imágenes de resonancia magnética (MRI) estructural para la detección de la enfermedad de Alzheimer en diferentes estadios clínicos (CN, MCI y VMCI), utilizando modelos de aprendizaje profundo. Se ha empleado un subconjunto de datos derivado del estudio público ADNI, procesado desde la plataforma Kaggle, y se han evaluado diferentes arquitecturas de redes convolucionales, incluyendo modelos conocidos como ResNet18, AlexNet y MobileNet, así como una red CNN propuesta en la literatura. A lo largo del proyecto se han explorado distintas configuraciones de preprocesado, normalización, selección de cortes cerebrales por paciente, aumento de datos y uso de optimizadores. Los resultados muestran que la combinación de una selección cuidadosa de cortes axiales, junto con normalización estándar y el uso del optimizador Adam, proporciona el mejor rendimiento en términos de precisión y F1-score. Además, se ha evidenciado la importancia de una correcta separación de datos por paciente y una evaluación crítica de modelos disponibles en repositorios públicos, evitando fugas de información. Aunque los rendimientos alcanzados no son óptimos, el trabajo sienta las bases para futuras mejoras y análisis más profundos en este ámbito.

Palabras clave: Alzheimer, imágenes médicas, MRI, aprendizaje profundo, clasificación, redes neuronales convolucionales, ADNI.

Índice

1. Introducción	7
1.1. Motivación	7
1.2. Objetivos	8
1.3. Tecnologías usadas	9
1.3.1. Lenguaje de programación y entorno de desarrollo	9
1.3.2. Procesamiento de datos y visualización	9
1.3.3. Tratamiento de imágenes	9
1.3.4. Modelado y entrenamiento	10
1.3.5. Modelos preentrenados de ImageNet	10
1.3.6. Plataforma de datos	10
1.4. Metodología	10
1.4.1. Marco ágil SCRUM	10
1.4.2. Aplicación del método científico	11
2. Estado del arte	13
2.1. Enfermedad de Alzheimer y neuroimagen	13
2.1.1. Biomarcadores en la enfermedad de Alzheimer	13
2.1.2. Técnicas de neuroimagen	14
2.2. Aplicación de Machine Learning en el diagnóstico de Alzheimer	15
2.2.1. Primeros enfoques clásicos de Machine Learning	15
2.2.2. Deep Learning en imágenes médicas	16
2.2.3. Modelos y arquitecturas utilizadas en Alzheimer	16
2.2.4. Limitaciones de los enfoques actuales	17
2.3. Generación de datos sintéticos en imágenes médicas	17
2.3.1. Data augmentation tradicional	17
2.3.2. Uso de GANs y técnicas avanzadas	17
2.4. Resumen del estado del arte	18
3. Implementación	19
3.1. Diseño	19
3.2. Descripción del conjunto de datos	19
3.3. Métricas usadas	21

4. Resultados Obtenidos	23
4.1. Versión 0 – Exploración inicial	24
4.2. Versión 1 – Clasificación por paciente	32
4.3. Versión 2 – Reducción del dataset y modelos base	41
4.4. Versión 3 – Imagen central por paciente	49
4.5. Versión 4 – Múltiples <i>slices</i> por paciente	57
4.6. Versión 5 – Ajustes de normalización y optimización	64
5. Discusión	71
5.1. Discusión	71
6. Aplicación Web	75
6.1. Funcionamiento de la aplicación demo	75
7. Conclusiones y Líneas Futuras	79
7.1. Conclusiones	79
7.2. Líneas Futuras	80
Apéndice A. Manual de	
Instalación	87
A.1. Requisitos previos	87
A.2. Descarga del repositorio	87
A.3. Instalación de dependencias	87
A.4. Descarga y preparación de los datos	87
A.5. Notas adicionales	88
A.6. Interfaz de Usuario	88
Apéndice B. Manual de	
Usuario	89

Introducción

1.1. Motivación

La demencia es un síndrome caracterizado por el deterioro progresivo de las funciones cognitivas y de la capacidad para realizar actividades cotidianas de forma independiente [1]. Entre las distintas formas de demencia, la enfermedad de Alzheimer (AD) es la más común, afectando principalmente a personas de edad avanzada. En sus fases iniciales, el Alzheimer suele manifestarse como un deterioro cognitivo leve (MCI, por sus siglas en inglés o VMCI para deterioro cognitivo muy leve), una condición que implica déficits de memoria sin que exista una alteración significativa en la vida diaria. No obstante, cuando las alteraciones cognitivas afectan funciones como el razonamiento, el lenguaje y la percepción espacial, interfiriendo con la autonomía del individuo, se diagnostica demencia.

A medida que la población envejece, la prevalencia del Alzheimer ha ido en aumento, lo que ha generado una creciente preocupación tanto en el ámbito sanitario como en el social. Según el *World Alzheimer Report* [2], en 2010 más de 35 millones de personas vivían con demencia, y se estima que esta cifra superará los 115 millones para el año 2050. Además, el coste anual del cuidado de estos pacientes representa aproximadamente el 1 % del PIB global, lo que refleja la magnitud del desafío que supone esta patología para los sistemas de salud y las economías de los países.

En este contexto, la detección precoz de la enfermedad se convierte en un objetivo prioritario, ya que, si bien no existe una cura, una identificación temprana permite aplicar estrategias que pueden ralentizar la progresión de los síntomas y mejorar la calidad de vida del paciente. Sin embargo, este diagnóstico temprano continúa siendo complejo debido a la superposición de los síntomas iniciales del Alzheimer con los procesos normales del envejecimiento y con otras enfermedades neurodegenerativas.

Las técnicas de neuroimagen han emergido como una herramienta clave para la detección de la enfermedad, siendo la resonancia magnética estructural (MRI) una de las más utilizadas. Esta modalidad permite observar cambios anatómicos en el cerebro, como la atrofia del hipocampo, que están estrechamente relacionados con el Alzheimer [3]. Comparada con otras técnicas como la tomografía por emisión de positrones (PET), la MRI es menos costosa, más accesible y ha demostrado ofrecer resultados diagnósticos de calidad comparable [3].

El avance de la inteligencia artificial ha permitido explorar nuevas vías para mejorar el análisis de estas imágenes médicas. En particular, las redes neuronales convolucionales (CNNs) han demostrado una gran eficacia en la extracción de características complejas y patrones ocultos en las imágenes [4]. Estas redes permiten clasificar de manera automática a los pacientes según su estado clínico —control sano, MCI o Alzheimer—, ofreciendo así una herramienta poderosa para el diagnóstico asistido por ordenador [5].

No obstante, la aplicación de estas técnicas aún enfrenta ciertos desafíos. Entre ellos, destacan la calidad variable de las imágenes médicas debido a artefactos o errores de segmentación, la ambigüedad en los criterios diagnósticos entre MCI y AD, y la escasez de conjuntos de datos amplios, equilibrados y bien etiquetados para entrenar modelos robustos [5].

Este Trabajo de Fin de Grado aborda estos retos mediante el uso del conjunto de datos ADNI (Alzheimer’s Disease Neuroimaging Initiative) [6], uno de los más completos y estandarizados en la investigación de la enfermedad. ADNI proporciona imágenes MRI estructurales junto con otros biomarcadores, permitiendo un análisis profundo y multivariable del progreso del Alzheimer. El objetivo de este proyecto es desarrollar un modelo de aprendizaje profundo que, a partir de imágenes MRI, permita clasificar de forma precisa y automática las distintas etapas de la enfermedad, contribuyendo así al avance en su detección precoz.

1.2. Objetivos

El objetivo principal del proyecto es desarrollar un sistema que permita la utilización de uno o varios modelos de aprendizaje computacional para la clasificación automática de imágenes de resonancia magnética (MRI) de pacientes con Alzheimer en diferentes estadios clínicos.

Para lograr este objetivo general, se plantean los siguientes objetivos específicos:

- Realizar una recopilación y preprocesamiento de imágenes del conjunto de datos **ADNI** (*Alzheimer’s Disease Neuroimaging Initiative*) [6], con el fin de estructurar un subconjunto adecuado para el entrenamiento y evaluación de modelos, centrando el análisis en los cortes axiales centrales más representativos de cada paciente.
- Aplicar técnicas de *deep learning* utilizando distintos modelos de varios artículos, así como modelos arquitectónicos propuestos en la literatura, como *ResNet18*, *AlexNet*, *MobileNet*, con el objetivo de evaluar su rendimiento en la clasificación del estado clínico del Alzheimer a partir de imágenes.
- Analizar el comportamiento de los modelos en escenarios con disponibilidad limitada de datos, simulando condiciones comunes en entornos clínicos reales donde solo se dispone de una cantidad reducida de imágenes por clase.

- Implementar técnicas de **aumento de datos (data augmentation)** mediante transformaciones geométricas, para mejorar la diversidad del conjunto de entrenamiento y potenciar la capacidad de generalización de los modelos.

Respecto al conjunto de datos, se prevé utilizar la base pública **ADNI**, que proporciona imágenes MRI estructurales, junto con biomarcadores, datos clínicos y genéticos de pacientes en distintas fases de la enfermedad.

1.3. Tecnologías usadas

El desarrollo de este trabajo ha requerido una combinación de herramientas de programación, bibliotecas para el manejo de datos e imágenes, entornos de experimentación y modelos preentrenados. A continuación, se detallan las principales tecnologías empleadas, organizadas según su funcionalidad.

1.3.1. Lenguaje de programación y entorno de desarrollo

- **Python** [7]: Lenguaje principal empleado en el desarrollo. Su sintaxis clara y la disponibilidad de librerías específicas para inteligencia artificial lo convierten en una elección idónea para proyectos de aprendizaje automático.
- **Jupyter Notebook** [8]: Entorno interactivo utilizado para la implementación, visualización y documentación del flujo de trabajo, facilitando la depuración y presentación de resultados.
- **Google Colab** [9]: Plataforma en la nube que facilita el entrenamiento de modelos con acceso gratuito a GPUs, ideal para proyectos de deep learning sin recursos locales potentes.

1.3.2. Procesamiento de datos y visualización

- **NumPy** [10] y **Pandas** [11]: Utilizadas para la manipulación y análisis de datos estructurados. Pandas ha sido especialmente útil para el manejo de metadatos del conjunto de imágenes.
- **Matplotlib** [12] y **Seaborn** [13]: Herramientas fundamentales para la visualización de estadísticas, gráficos de distribución de clases y métricas de evaluación.

1.3.3. Tratamiento de imágenes

- **Pillow (PIL)** [14]: Librería empleada para operaciones de preprocesamiento sobre las imágenes, como redimensionamiento, normalización y conversión de formatos.

1.3.4. Modelado y entrenamiento

- **PyTorch** [15]: Framework de aprendizaje profundo utilizado para definir y entrenar redes neuronales convolucionales (CNNs), así como para aplicar técnicas de *data augmentation* y transfer learning.
- **scikit-learn** [16]: Empleado para modelos clásicos de machine learning y para el cálculo de métricas como precisión, recall y matriz de confusión.

1.3.5. Modelos preentrenados de ImageNet

Se han empleado modelos CNN preentrenados en ImageNet para beneficiarse del *transfer learning*, técnica útil cuando se dispone de un conjunto de datos limitado, como es el caso. Los modelos utilizados han sido:

- **ResNet-18** [17]: Red neuronal profunda con conexiones residuales que permiten entrenar redes más profundas sin que se degrade el rendimiento.
- **MobileNetV2** [18]: Modelo eficiente y ligero, especialmente diseñado para dispositivos con recursos limitados. Su arquitectura es óptima para tareas en las que la eficiencia computacional es crítica.
- **AlexNet** [19]: Uno de los primeros modelos de CNN en lograr resultados destacados en clasificación de imágenes, cuya arquitectura relativamente simple permite un entrenamiento rápido.

1.3.6. Plataforma de datos

- **Kaggle** [20]: Plataforma utilizada para la descarga del conjunto de datos, así como para la consulta de notebooks y referencias similares como punto de partida.

1.4. Metodología

La metodología seguida en el desarrollo del Trabajo de Fin de Grado se ha basado en una combinación del marco ágil **SCRUM** y del enfoque sistemático del **método científico**, permitiendo así una gestión iterativa del trabajo junto con una aproximación experimental rigurosa.

1.4.1. Marco ágil SCRUM

El proyecto se ha estructurado en *sprints* de entre 2 y 3 semanas, cada uno con objetivos concretos y alcanzables. Entre dichos objetivos se incluyeron tareas como la selección y preparación de datos, el preprocesamiento de imágenes MRI, la implementación de modelos de deep

learning, el análisis comparativo entre arquitecturas, la aplicación de técnicas de aumento de datos (*data augmentation*) y la evaluación con distintos optimizadores.

Al final de cada sprint se han llevado a cabo reuniones con los tutores, en las que se ha revisado el progreso, discutido los resultados obtenidos y planificado los siguientes pasos. Este enfoque ha permitido mantener una comunicación fluida y continua con los tutores, garantizando retroalimentación constante y una adaptación flexible a las necesidades que fueron surgiendo durante el desarrollo del trabajo.

1.4.2. Aplicación del método científico

Paralelamente, el desarrollo técnico del proyecto se ha guiado por el método científico, siguiendo una estructura lógica y empírica que ha permitido extraer conclusiones válidas a partir de los resultados. Las etapas aplicadas han sido:

1. **Observación:** Análisis inicial del problema clínico del diagnóstico del Alzheimer mediante imágenes médicas y revisión de trabajos previos en la literatura.
2. **Formulación de hipótesis:** Por ejemplo, evaluar si el uso de modelos preentrenados como ResNet o AlexNet ofrece un mejor rendimiento que modelos personalizados; o si la normalización de datos y el aumento de datos afectan significativamente al desempeño.
3. **Experimentación:** Diseño de experimentos en los que se varían las condiciones (número de imágenes, técnicas de preprocesamiento, modelos utilizados, optimizadores, etc.) para validar o refutar las hipótesis planteadas.
4. **Análisis de resultados:** Evaluación cuantitativa mediante métricas como *accuracy*, *F1-score*, *recall* y *loss*, así como visualización mediante gráficas y matrices de confusión.
5. **Conclusión:** Interpretación de los resultados experimentales para extraer aprendizajes y reflexiones, y establecer recomendaciones de mejora.

Esta combinación metodológica ha permitido abordar el proyecto de forma estructurada, flexible y rigurosa, adaptándose a las particularidades del problema y del flujo de trabajo necesario para el desarrollo y validación de modelos de aprendizaje profundo aplicados a imágenes médicas.

Estado del arte

El diagnóstico precoz de la enfermedad de Alzheimer (AD) es un desafío fundamental en el ámbito de la neurociencia y la medicina, debido a la complejidad de los síntomas iniciales y a la superposición con los procesos normales del envejecimiento. En este contexto, las técnicas de neuroimagen, en particular la resonancia magnética estructural (MRI), han demostrado ser herramientas clave para la identificación de biomarcadores relacionados con la progresión de la enfermedad. Sin embargo, la interpretación manual de estas imágenes resulta compleja y requiere de personal especializado, lo que ha motivado la exploración de métodos automatizados basados en inteligencia artificial.

Durante la última década, el aprendizaje automático (*machine learning*) y, más recientemente, el aprendizaje profundo (*deep learning*) han mostrado un gran potencial para el análisis de imágenes médicas. Las redes neuronales convolucionales (CNNs) han sido especialmente efectivas en la clasificación automática de imágenes MRI en diferentes etapas de la enfermedad de Alzheimer. No obstante, la aplicación de estos métodos enfrenta diversos retos, como la limitada disponibilidad de datos etiquetados, el desequilibrio entre clases y la variabilidad en la calidad de las imágenes.

2.1. Enfermedad de Alzheimer y neuroimagen

2.1.1. Biomarcadores en la enfermedad de Alzheimer

El diagnóstico y seguimiento de la enfermedad de Alzheimer se basa en el uso de diversos biomarcadores que reflejan las alteraciones estructurales y funcionales asociadas a la progresión de la enfermedad. Estos biomarcadores se agrupan habitualmente en el marco AT(N): depósitos de amiloide (A), alteraciones en tau (T), y neurodegeneración o daño neuronal (N) [21].

En este trabajo nos centramos en el biomarcador N, evaluable mediante imágenes de resonancia magnética estructural (MRI). Este biomarcador refleja la presencia de atrofia cerebral, que es uno de los cambios más característicos en la progresión de la enfermedad. Entre los biomarcadores estructurales más relevantes destacan:

- **Atrofia del hipocampo**, correlacionada con el deterioro de la memoria episódica y con-

siderada uno de los principales predictores estructurales de la conversión de deterioro cognitivo leve (MCI) a Alzheimer [22].

- **Reducción del volumen cortical**, en particular en la corteza temporal medial, la corteza parietal y regiones del precúneo y del cíngulo posterior, asociada con el avance de los síntomas clínicos [23].
- **Pérdida de materia gris global y regional**, que se intensifica a medida que progresa la enfermedad [24].

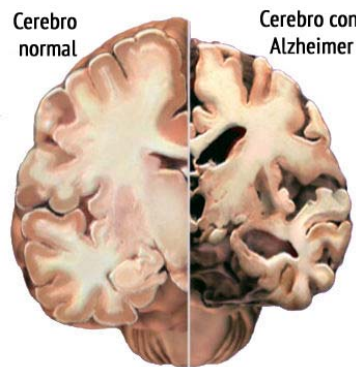


Figura 1: Comparación visual de un cerebro sano (izquierda) y un cerebro con Alzheimer (derecha). Se observa una notable reducción de masa cerebral y agrandamiento de los ventrículos. Imagen tomada de MD.Saúde (2020) [25].

Los patrones de neurodegeneración en MRI siguen una progresión característica: la atrofia comienza en el hipocampo y las regiones temporales mediales durante las fases preclínicas y de deterioro cognitivo leve, y se extiende posteriormente a la corteza parietal, la corteza frontal y otras áreas corticales en fases avanzadas [21, 26].

El uso de modelos de aprendizaje profundo, en particular redes neuronales convolucionales (CNNs), ha mostrado una gran eficacia en la identificación automática de estos patrones de atrofia a partir de imágenes MRI [26, 24], lo que constituye la base metodológica del enfoque adoptado en este trabajo.

2.1.2. Técnicas de neuroimagen

La neuroimagen desempeña un papel esencial en el diagnóstico y seguimiento de la enfermedad de Alzheimer, al permitir la visualización in vivo de los cambios estructurales y funcionales del cerebro. Entre las diferentes modalidades de neuroimagen disponibles, en este trabajo se ha empleado exclusivamente la **resonancia magnética estructural** (MRI), que constituye una herramienta ampliamente utilizada en la práctica clínica.

La MRI estructural permite obtener imágenes de alta resolución de la anatomía cerebral, facilitando la cuantificación de parámetros relevantes como el volumen del hipocampo, el volumen cortical, o la integridad de la materia gris, todos ellos biomarcadores asociados a la progresión de la enfermedad [22, 21].

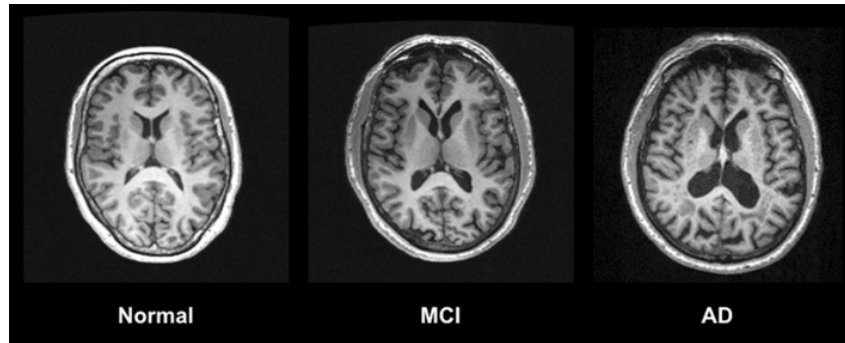


Figura 2: Imagen de resonancia magnética (MRI) mostrando un cerebro normal, con deterioro cognitivo leve (MCI) y con enfermedad de Alzheimer (AD). Se puede observar una progresiva pérdida de volumen cerebral y aumento de los ventrículos. Imagen captada de Barkhof et al. (2018) [27].

Otras técnicas de neuroimagen utilizadas en la investigación del Alzheimer incluyen la tomografía por emisión de positrones (PET), que permite visualizar directamente los depósitos de amiloide y tau, y la resonancia magnética funcional (fMRI), que evalúa la conectividad funcional del cerebro [24]. Sin embargo, estas técnicas presentan ciertas limitaciones, como un mayor coste, menor disponibilidad, y la necesidad de agentes de contraste en el caso de la PET.

En comparación, la MRI estructural ofrece varias ventajas: es una técnica no invasiva, más accesible y menos costosa, y permite detectar los cambios neurodegenerativos que subyacen a la progresión del Alzheimer [23]. Por estos motivos, ha sido la modalidad elegida en este proyecto para el entrenamiento y validación de modelos de aprendizaje profundo.

2.2. Aplicación de Machine Learning en el diagnóstico de Alzheimer

2.2.1. Primeros enfoques clásicos de Machine Learning

En los primeros estudios sobre automatización del diagnóstico de Alzheimer se utilizaron técnicas clásicas como Support Vector Machine (SVM), Random Forest (RF) y k-Nearest Neighbors (k-NN). Estos métodos requerían una extracción manual de características, como volúmenes de regiones específicas o textura de la materia gris, limitando su capacidad para capturar representaciones complejas [28, 29]. Por ejemplo, se han aplicado Random Forest

con éxito para predecir la conversión de MCI a AD, alcanzando alrededor de un 94 % de exactitud [28]. Aun así, su desempeño era sensible a la calidad del preprocesado y a la selección de características.

2.2.2. Deep Learning en imágenes médicas

Las redes neuronales convolucionales (CNNs) han revolucionado el análisis de imágenes médicas gracias a su capacidad para aprender características directamente de los píxeles [24]. En neuroimagen, Payan & Montana (2015) demostraron que una CNN 3D entrenada con MRI de ADNI mejora significativamente el diagnóstico automático, superando a clasificadores tradicionales [26]. Estudios posteriores han corroborado estos hallazgos utilizando CNNs 2D y modelos híbridos, evidenciando su eficacia en la clasificación de Alzheimer [30].

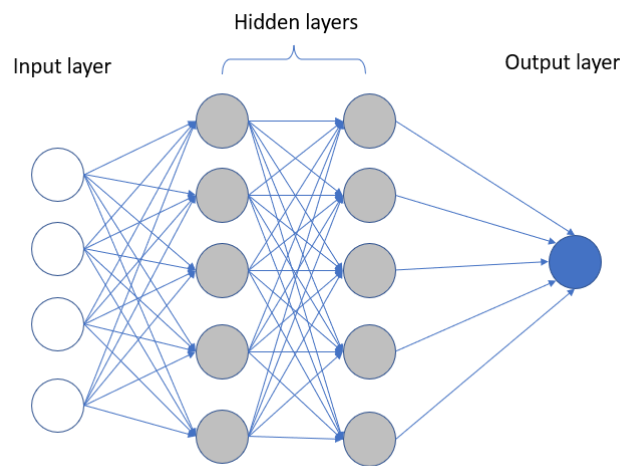


Figura 3: Ejemplo esquemático de una red neuronal artificial. Se muestran las capas de entrada, ocultas y de salida, con sus respectivas conexiones. Imagen tomada de Hrushikesh, “How to visualize the layers of a deep learning model”[31].

2.2.3. Modelos y arquitecturas utilizadas en Alzheimer

- **CNNs 3D:** capturan patrones volumétricos de atrofia y han logrado más del 90 % de precisión en MRI estructural [26, 32].
- **Transfer Learning:** adaptar modelos pre-entrenados como VGG o ResNet ha permitido buenos resultados incluso con datos limitados [24].
- **Data Augmentation:** se utilizan transformaciones geométricas y generación de imágenes sintéticas para equilibrar clases y robustecer el entrenamiento.

2.2.4. Limitaciones de los enfoques actuales

- **Escasez de datos:** incluso conjuntos grandes como ADNI son limitados para entrenar redes profundas.
- **Desequilibrio de clases:** etapas como MCI y Alzheimer moderado son poco representadas.
- **Variabilidad entre escáneres:** protocolos e intensidades variables dificultan la generalización [29].
- **Interpretabilidad:** las CNNs funcionan como “cajas negras”, lo que es una barrera para su adopción clínica.

2.3. Generación de datos sintéticos en imágenes médicas

2.3.1. Data augmentation tradicional

El *data augmentation* clásico consiste en aplicar transformaciones geométricas y de intensidad a las imágenes originales para aumentar la diversidad del conjunto de entrenamiento. Entre las técnicas más comunes se encuentran la rotación, traslación, escalado y la inyección de ruido gaussiano o sal y pimienta. Estas transformaciones ayudan a mejorar la generalización de los modelos al simular variaciones realistas en los datos [33, 34]. En el contexto de la clasificación de Alzheimer mediante MRI, su aplicación ha demostrado incrementar la precisión en torno a un 2–10 % [35].

2.3.2. Uso de GANs y técnicas avanzadas

Las redes generativas adversariales (GANs) son una familia de modelos que aprenden a generar imágenes sintéticas realistas mediante la confrontación entre un generador y un discriminador. Su uso en MRI cerebral permite tanto aumentar el tamaño del dataset como mejorar la calidad de las imágenes (p.ej. resolución o reducción de ruido) [36]. Aplicadas al diagnóstico de Alzheimer, estudios han mostrado que las imágenes sintéticas generadas con GANs mejoran la exactitud del clasificador y aumentan la calidad objetiva de la imagen (mejora del SNR) [37].

Además, investigaciones recientes utilizando GANs específicos (como Super Resolution GANs) han logrado clasificar múltiples etapas de la enfermedad con una precisión del 99 % [38]. No obstante, el uso de datos sintéticos también plantea cuestiones éticas y clínicas, tales como la fidelidad anatómica, posibles sesgos introducidos por el generador, y la necesidad de validación cruzada en distintos centros para asegurar su aplicabilidad clínica.

2.4. Resumen del estado del arte

En los últimos años, el uso de técnicas de aprendizaje automático y aprendizaje profundo ha permitido avances notables en el diagnóstico asistido de la enfermedad de Alzheimer, especialmente a través del análisis de neuroimágenes como la MRI estructural. Modelos clásicos como SVM o Random Forest han demostrado su utilidad inicial, aunque su rendimiento suele estar limitado por la necesidad de una extracción manual de características. Por otro lado, las redes neuronales convolucionales (CNNs) han permitido automatizar este proceso, mostrando buenos resultados en la clasificación de distintos estados clínicos del Alzheimer [4, 29].

Pese a estos avances, persisten retos relevantes: la escasez de datos bien etiquetados, el desequilibrio entre clases, y la variabilidad en los escáneres o protocolos de adquisición. En este contexto, el uso de técnicas de *data augmentation*, tanto tradicionales como basadas en métodos generativos, ha emergido como una estrategia clave para mejorar la generalización de los modelos [35].

3

Implementación

3.1. Diseño

El diseño general del flujo de trabajo seguido durante el desarrollo del proyecto se resume en la Figura 4. El proceso completo puede dividirse en cuatro grandes etapas: selección y preparación de datos, preprocesamiento, entrenamiento de modelos y clasificación/predicción final.

- **Imagen MRI de entrada:** se seleccionaron las clases de interés del conjunto de datos, y se realizó una primera exploración del número y distribución de imágenes por clase.
- **Preprocesamiento:** las imágenes se redimensionaron y se normalizaron y, en algunas versiones experimentales, se aplicaron técnicas de aumento de datos. También se aplicó una división por paciente para evitar el solapamiento entre los conjuntos de entrenamiento y validación.
- **Entrenamiento:** se evaluaron distintos optimizadores (*Adam*, *AdamW*, *SGD*) y tamaños de lote. Además, se consideraron modelos convolucionales preentrenados y arquitecturas desde cero, incluyendo *Sivaranjani-CNN*, *ResNet18*, *AlexNet* y *MobileNet*.
- **Clasificación y predicción:** las salidas del modelo se evaluaron utilizando métricas de *accuracy* y *F1-score*, prestando especial atención al rendimiento en clases desbalanceadas.

3.2. Descripción del conjunto de datos

Para el desarrollo de este trabajo se ha utilizado un conjunto de datos que constituye un subconjunto del reconocido dataset ADNI (*Alzheimer's Disease Neuroimaging Initiative*) [6]. Este subconjunto ha sido recopilado y puesto a disposición en la plataforma Kaggle (una plataforma para compartir datasets y competiciones de ciencia de datos) [20], y está basado en imágenes provenientes del proyecto OASIS (*Open Access Series of Imaging Studies*) [39]. Se ha optado por el uso de este subconjunto debido a su tamaño reducido y accesibilidad, lo que

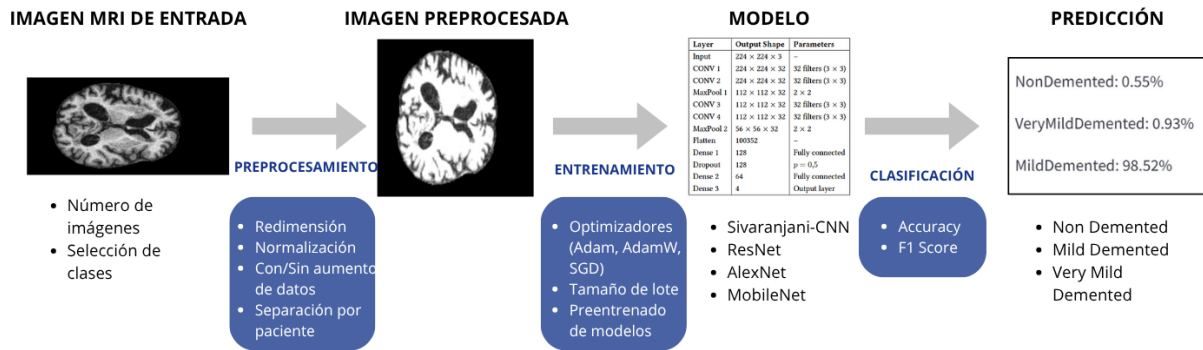


Figura 4: Resumen gráfico del flujo de trabajo seguido en el proyecto, desde la imagen MRI original hasta la predicción final de clase.

permite una gestión más eficiente de los recursos computacionales durante el proceso de entrenamiento y evaluación de modelos.

El conjunto contiene imágenes de resonancia magnética estructural cerebral (MRI) ya clasificadas en función del estado de deterioro cognitivo del paciente. Estas imágenes corresponden a cortes bidimensionales (2D slices) de volúmenes tridimensionales (3D).

Las clases presentes en el conjunto de datos son: *NonDemented* (personas sin signos de demencia), *VeryMildDemented* (personas con signos muy leves de deterioro cognitivo), *MildDemented* (personas con deterioro cognitivo leve), y *ModerateDemented* (personas con deterioro cognitivo moderado).

El total de imágenes en el dataset asciende a 82 916, distribuidas entre las distintas clases de la siguiente manera:

- **NonDemented:** 63 560 imágenes
- **VeryMildDemented:** 13 796 imágenes
- **MildDemented:** 5 184 imágenes
- **ModerateDemented:** 376 imágenes

En la Figura 5 se presentan ejemplos visuales representativos de cada una de las cuatro clases del conjunto de datos. Cada columna corresponde a una clase distinta, y se muestran dos imágenes por clase para ilustrar la variabilidad interna de las muestras.

Además de las imágenes, el dataset incluye un archivo CSV asociado que contiene metadatos relevantes para cada muestra, tales como el identificador del paciente, sexo, edad, nivel educativo, puntuaciones clínicas (*MMSE*, *CDR*), y medidas neuroanatómicas derivadas como

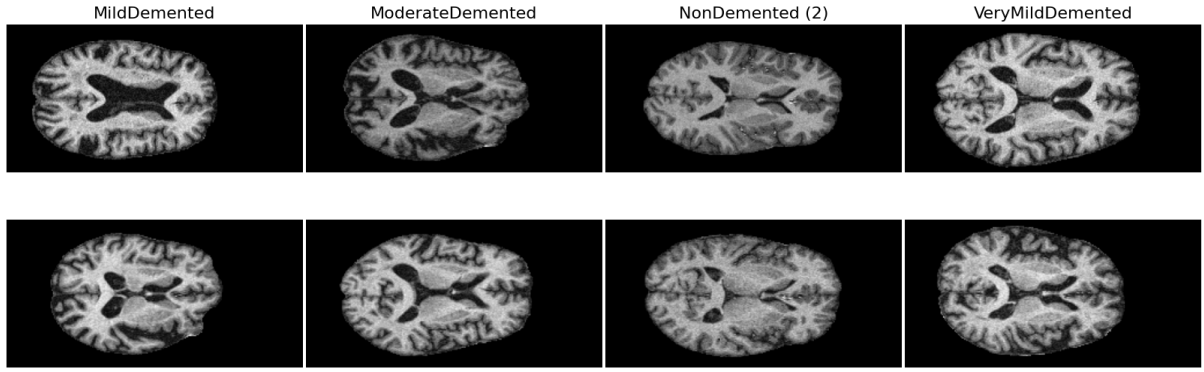


Figura 5: Ejemplos representativos de imágenes por clase del conjunto de datos utilizado.

SubjectID	M/F	Hand	Age	Educ	SES	MMSE	CDR	eTIV	nWBV	ASF	Delay
OAS1_0001_MR1	F	R	74	2.0	3.0	29.0	0.0	1344	0.743	1.306	NaN
OAS1_0002_MR1	F	R	55	4.0	1.0	29.0	0.0	1147	0.810	1.531	NaN
OAS1_0003_MR1	F	R	73	4.0	3.0	27.0	0.5	1454	0.708	1.207	NaN

Cuadro 1: Ejemplo de filas del conjunto de datos OASIS.

eTIV (volumen intracraneal estimado), *nWBV* (volumen cerebral normalizado), y *ASF* (factor de escala anatómico). Un extracto representativo del CSV es el siguiente:

Este subconjunto concreto ha sido extraído directamente del repositorio publicado por Yiwei Lu en Kaggle [40], lo que permite una implementación directa de modelos supervisados.

3.3. Métricas usadas

Durante el entrenamiento y la evaluación de los modelos, se han empleado diversas métricas para medir su rendimiento. A continuación, se describen las principales métricas utilizadas,:

- **Precisión global (Accuracy):** mide el porcentaje de ejemplos correctamente clasificados sobre el total.

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

- **F1-Score:** métrica que combina la precisión (Precision) y la sensibilidad (Recall), resultando especialmente útil en contextos con clases desbalanceadas. Es la media armónica entre ambas:

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

- **Pérdida (Loss):** función que mide la discrepancia entre las predicciones del modelo y las etiquetas reales. En este trabajo se ha utilizado la *Cross Entropy Loss* (entropía cruzada), que para clasificación multiclase se define como:

$$Loss = - \sum_i y_i \log(p_i)$$

donde y_i es la etiqueta real (one-hot) y p_i es la probabilidad predicha para la clase i .

- **Matriz de confusión (Confusion Matrix):** proporciona un resumen visual del desempeño del modelo mostrando los conteos de:
 - **TP (True Positives):** verdaderos positivos.
 - **TN (True Negatives):** verdaderos negativos.
 - **FP (False Positives):** falsos positivos.
 - **FN (False Negatives):** falsos negativos.

A partir de la matriz de confusión, se derivan otras métricas útiles:

- **Sensibilidad o Recall (TPR, True Positive Rate):**

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

- **Especificidad (TNR, True Negative Rate):**

$$Specificity = \frac{TN}{TN + FP}$$

- **Precisión (Precision):**

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

- **Tasa de Falsos Positivos (FPR, False Positive Rate):**

$$FPR = \frac{FP}{FP + TN}$$

- **Curva ROC (Receiver Operating Characteristic):** curva que representa la relación entre la TPR y la FPR para distintos umbrales de decisión. Cuanto más cercana esté la curva al vértice superior izquierdo, mejor es el modelo.
- **Área bajo la curva ROC (AUC, Area Under the Curve):** mide el área total bajo la curva ROC. Un AUC de 1 indica un modelo perfecto, mientras que un AUC de 0.5 indica un modelo aleatorio.

4

Resultados Obtenidos

A lo largo del desarrollo del trabajo, se realizaron múltiples experimentos que fueron evolucionando en respuesta a las limitaciones observadas y a la búsqueda de un enfoque más robusto y realista. Inicialmente, los modelos se entrenaban sobre el conjunto completo de imágenes sin separar por paciente, lo cual generaba métricas artificialmente elevadas. Posteriormente, se ajustó la organización de los datos, se exploraron diferentes configuraciones de red neuronal, técnicas de aumento de datos y estrategias de preprocesamiento.

Para estructurar adecuadamente este proceso evolutivo, se han definido varias **versiones experimentales**. Estas versiones permiten comprender la progresión seguida durante el proyecto y facilitan la comparación entre enfoques. La Tabla 2 resume estas versiones y los aspectos clave de cada una.

Cuadro 2: Resumen de las versiones experimentales desarrolladas

Versión	Nombre	Cambios aplicados	Modelos probados
V0	Exploración inicial	■ Uso de todas las imágenes sin separar por paciente.	4
V1	Clasificación por paciente	■ Reorganización por paciente. ■ Eliminación de clase minoritaria. ■ Primeros experimentos con aumento de datos y binarización.	7

Versión	Nombre	Cambios aplicados	Modelos probados
V2	Reducción del dataset y modelos base	- Reducción del conjunto de datos a 1000 y 5000 imágenes por clase. - Evaluación con modelos pre-entrenados (ResNet, MobileNet, AlexNet).	8
V3	Imagen central por paciente	Selección de una única imagen (slice central) por paciente como entrada.	7
V4	Múltiples slices por paciente	Entrada compuesta por tres imágenes (slices) por paciente.	5
V5	Ajustes de normalización y optimización	- Normalización al rango [0,1]. - Comparación de distintos optimizadores (Adam, AdamW, SBG).	6

4.1. Versión 0 – Exploración inicial

En esta primera versión, se probaron varios modelos que ya habían sido publicados, tanto en artículos científicos como en plataformas públicas, para analizar cómo se comportan al ser entrenados directamente con el conjunto de imágenes utilizado en este trabajo. En concreto, se eligieron **tres modelos sobre la clasificación de resonancias magnéticas para detectar el Alzheimer**, además de un modelo que se encuentra disponible en Kaggle. Todos estos modelos se reentrenaron desde cero utilizando los datos específicos del estudio, sin aplicar ajustes o técnicas de mejora adicionales por el momento.

Modelo de Kaggle

Se empleó un modelo publicado por Naeem Nagori en la plataforma Kaggle [41], desarrollado específicamente sobre el mismo conjunto de datos empleado en este trabajo. El modelo se basa en la arquitectura **EfficientNet-B0**, preentrenada en ImageNet y adaptada a la clasificación multiclase con cuatro categorías mediante la modificación de su capa final. Este enfoque proporciona un punto de partida relevante y permite establecer una comparación inicial con otros modelos más complejos o personalizados desarrollados posteriormente.

El entrenamiento empleó la función de pérdida **CrossEntropyLoss**, el optimizador **AdamW** (con una tasa de aprendizaje de 0.0001 y un *weight decay* de 10^{-4}), y un programador de tasa de aprendizaje **ReduceLROnPlateau**. A lo largo de 20 épocas, el modelo alcanzó una **precisión final en test del 99.95 %**.

Los resultados obtenidos fueron notablemente altos, con métricas de validación que se mantuvieron cercanas al 100 % a lo largo del entrenamiento. Como muestra representativa del rendimiento del modelo, en la Figura 6 se presenta la curva ROC multiclase, en la que se observa un comportamiento sobresaliente en las cuatro clases evaluadas. No obstante, unos valores tan elevados resultan inusuales en tareas reales de clasificación médica.

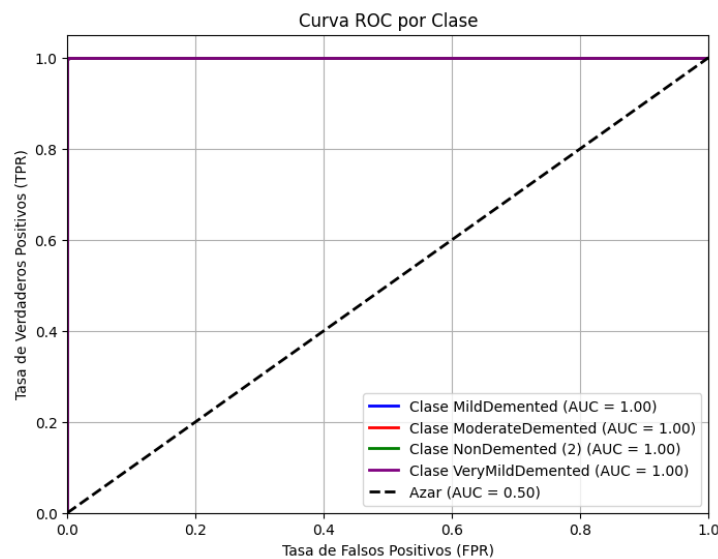


Figura 6: Curva ROC multiclase del modelo de Kaggle [41].

Además, en la Figura 7 se incluye la matriz de confusión correspondiente, donde se evidencia una clasificación prácticamente perfecta en todas las categorías, lo cual refuerza la necesidad de revisar la configuración experimental en versiones posteriores.

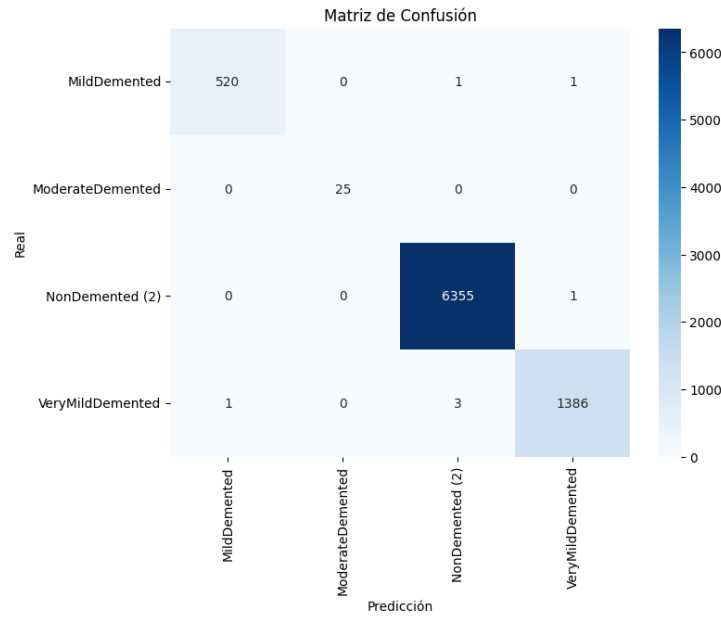


Figura 7: Matriz de confusión del modelo de Kaggle [41].

Modelo basado en artículo: *A deep learning based CNN approach on MRI for Alzheimer's disease detection*

El primer modelo reproducido fue propuesto en el artículo *A deep learning based CNN approach on MRI for Alzheimer's disease detection* [42]. En dicho trabajo, los autores emplean una red neuronal convolucional (CNN) de arquitectura sencilla, entrenada sobre un conjunto de imágenes de resonancia magnética (MRI) procedentes de la base de datos pública OASIS. En concreto, utilizan cortes 2D seleccionados manualmente para clasificar las imágenes en dos categorías: pacientes con Alzheimer y sujetos sanos.

La arquitectura del modelo original está compuesta por tres capas convolucionales, una capa de *max pooling*, dos bloques de *dropout* y dos capas completamente conectadas (*fully connected*). La red utiliza funciones de activación ReLU y fue diseñada para una tarea de clasificación binaria.

En este trabajo, se ha tomado como inspiración dicha arquitectura, adaptándola a la tarea de clasificación multiclase con cuatro categorías diagnósticas. La adaptación incluyó una modificación en la última capa densa para producir cuatro salidas, en lugar de dos. Además, se ha empleado el conjunto de imágenes proporcionado por el proyecto ADNI, procesado según los criterios descritos previamente. A modo ilustrativo, la arquitectura utilizada en el presente trabajo es equivalente a la representada en artículo original.

El modelo adaptado mostró un rendimiento desigual entre clases. Tal como se observa en el reporte de clasificación 4, la clase *NonDemented* fue clasificada correctamente con una pre-

Cuadro 3: Estructura del modelo basado en el trabajo de Roy et al.[42], adaptado a cuatro clases.

Layer	Input shape	Output shape
Input	$(176 \times 176 \times 1)$	$(176 \times 176 \times 1)$
Conv2D	$(176 \times 176 \times 1)$	$(174 \times 174 \times 12)$
Conv2D	$(174 \times 174 \times 12)$	$(172 \times 172 \times 4)$
Conv2D	$(172 \times 172 \times 4)$	$(170 \times 170 \times 2)$
MaxPooling2D	$(170 \times 170 \times 2)$	$(85 \times 85 \times 2)$
Dropout	$(85 \times 85 \times 2)$	$(85 \times 85 \times 2)$
Flatten	$(85 \times 85 \times 2)$	14,450
Dense	14,450	16
Dropout	16	16
Dense (Output)	16	4

cisión del 82 % y una *recall* perfecta del 100 %, lo que sugiere que el modelo tiende a clasificar la mayoría de las muestras dentro de esta categoría mayoritaria. Sin embargo, el rendimiento sobre las demás clases fue significativamente inferior: tanto *MildDemented* como *ModerateDemented* obtuvieron métricas nulas (precisión y *recall* de 0.00), mientras que *VeryMildDemented* alcanzó una precisión del 63 % y un *recall* del 27 %.

Cuadro 4: Reporte de clasificación del modelo basado en el artículo Roy et al. [42].

Clase	Precisión	Recall	F1-score	Support
MildDemented	0.00	0.00	0.00	522
ModerateDemented	0.00	0.00	0.00	25
NonDemented	0.82	1.00	0.90	6356
VeryMildDemented	0.63	0.27	0.38	1390
Accuracy	0.81 (sobre 8293 muestras)			
Macro avg	0.36	0.32	0.32	8293
Weighted avg	0.74	0.81	0.75	8293

El valor de **accuracy global fue del 81 %**, pero esta métrica resulta engañosa debido al fuerte desbalance de clases. En efecto, el modelo mostró una tendencia clara a favorecer la clase más representada, lo cual se refleja en el *f1-score* ponderado de 0.75 frente a un *f1-score* macro de solo 0.32. La Figura 9 presenta la matriz de confusión correspondiente, mientras que en la Figura 8 se incluye la curva ROC multiclase, que ilustra las diferencias de comportamiento entre clases.

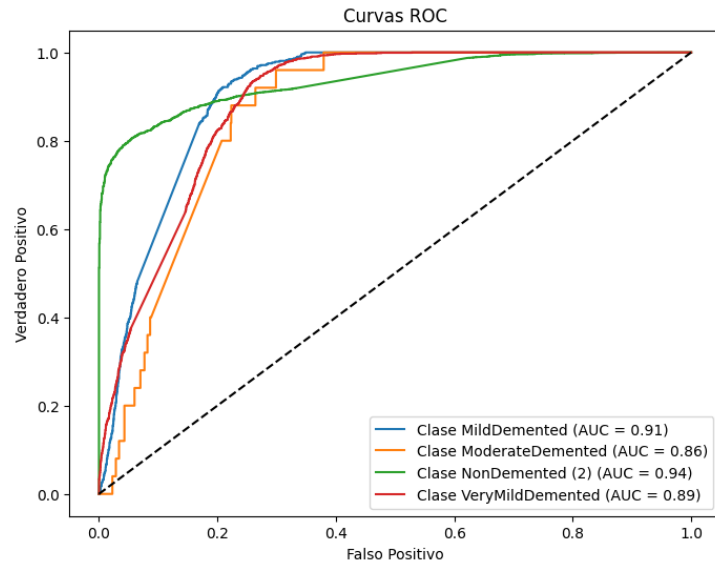


Figura 8: Curva ROC multiclase del modelo basado en la CNN del artículo de Roy et al. [42].

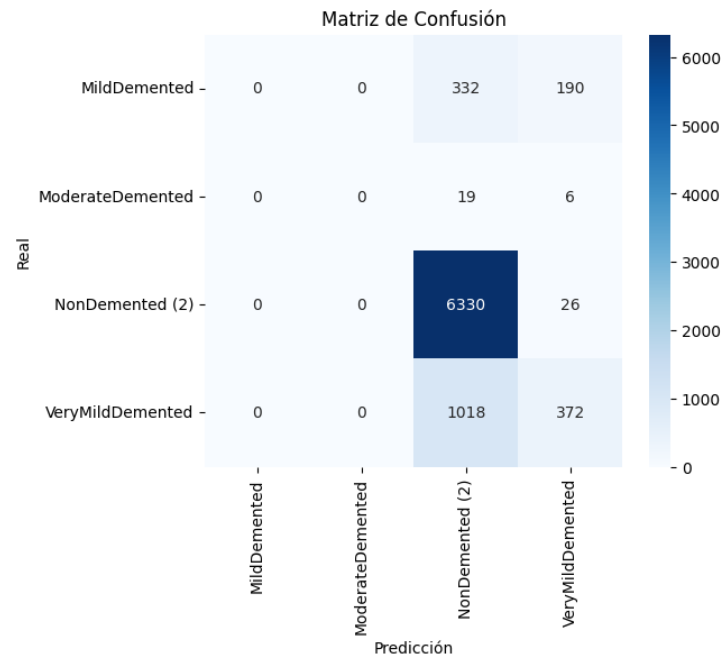


Figura 9: Matriz de confusión del modelo basado en la CNN del artículo de Roy et al. [42].

Modelo basado en el artículo: *A CNN based framework for classification of Alzheimer's disease*

El segundo modelo considerado está inspirado en el trabajo de Sivaranjani et al., *A CNN based framework for classification of Alzheimer's disease* [43], en el que se propone un enfoque basado en redes convolucionales para la clasificación de la enfermedad de Alzheimer a partir de imágenes de resonancia magnética. En el estudio original, se emplean imágenes de tamaño reducido (64×64) y se evalúan distintas configuraciones arquitectónicas, variando tanto la profundidad como el número de filtros de las capas convolucionales. La arquitectura propuesta combina bloques de convolución y agrupamiento, seguidos por capas densas, y se aplica tanto a tareas de clasificación binaria (presencia o ausencia de la enfermedad) como a clasificación multiclase, considerando diferentes etapas de progresión del Alzheimer. El enfoque propuesto logró resultados prometedores, sirviendo como base para el diseño del modelo implementado en este trabajo.

Para este trabajo, se adaptó la arquitectura presentada a un problema de clasificación multiclase de cuatro categorías diagnósticas. Además, se aumentó la resolución de entrada a 224×224 píxeles, ajustando las dimensiones intermedias de las capas. El modelo implementado consta de dos bloques de convoluciones dobles con funciones de activación ReLU, seguidos de capas de agrupamiento MaxPool2d, y culmina en un conjunto de capas densas totalmente conectadas con regularización mediante Dropout. Esta estructura se puede observar en la tabla 5.

Cuadro 5: Estructura del modelo basado en Sivaranjani et al. [43].

Layer	Output Shape	Parameters
Input	$224 \times 224 \times 3$	–
CONV 1	$224 \times 224 \times 32$	32 filters (3×3)
CONV 2	$224 \times 224 \times 32$	32 filters (3×3)
MaxPool 1	$112 \times 112 \times 32$	2×2
CONV 3	$112 \times 112 \times 32$	32 filters (3×3)
CONV 4	$112 \times 112 \times 32$	32 filters (3×3)
MaxPool 2	$56 \times 56 \times 32$	2×2
Flatten	100352	–
Dense 1	128	Fully connected
Dropout	128	$p = 0,5$
Dense 2	64	Fully connected
Dense 3	4	Output layer

Tal y como se recoge en el reporte de clasificación (Tabla 6), el modelo presentó un compor-

tamiento excelente para la clase mayoritaria (NonDemented), con valores de precisión y recall cercanos a 1. También mostró un buen rendimiento en las clases MildDemented y VeryMildDemented, aunque se observó una caída notable en el recall para la clase ModerateDemented, probablemente debido al reducido número de muestras disponibles para esta categoría.

Cuadro 6: Reporte de clasificación del modelo basado en el segundo artículo de Sivaranjani et al. [43].

Clase	Precisión	Recall	F1-score	Support
MildDemented	0.95	0.90	0.92	519
ModerateDemented	1.00	0.47	0.64	38
NonDemented (2)	0.98	1.00	0.99	6356
VeryMildDemented	0.95	0.90	0.93	1380
Accuracy	0.97 (sobre 8293 muestras)			
Macro avg	0.97	0.82	0.87	8293
Weighted avg	0.97	0.97	0.97	8293

Este modelo demostró un rendimiento significativamente alto en la clasificación de las distintas etapas del Alzheimer, alcanzando una precisión global del 97 % sobre el conjunto de validación.

En la Figura 10 y la Figura 11 se muestran respectivamente la curva ROC multiclase y la matriz de confusión correspondientes al modelo implementado a partir del segundo artículo [43].

La curva ROC muestra un área bajo la curva (AUC) perfecta para todas las clases, lo cual indica una capacidad de discriminación sobresaliente del modelo, aunque este resultado tan elevado podría indicar cierto sobreajuste. La matriz de confusión permite observar de forma más detallada los errores cometidos por el modelo, que en este caso fueron escasos y principalmente concentrados en las clases minoritarias.

Modelo basado en el artículo: *Diagnosis of Alzheimer Disease Using 2D MRI Slices by Convolutional Neural Network*

El tercer modelo analizado se basa en el estudio de Al-Khzaie et al. [44], donde se propone una red neuronal convolucional (CNN) para la detección de la enfermedad de Alzheimer utilizando imágenes de resonancia magnética en dos dimensiones. A diferencia de otros enfoques que trabajan con volúmenes 3D, los autores emplean cortes axiales individuales en formato 2D, lo que simplifica el procesamiento y reduce los requerimientos computacionales. La arquitectura propuesta combina cinco bloques de convolución y max pooling, seguidos por capas densas y una capa de salida para clasificación binaria. El modelo fue entrenado con imágenes

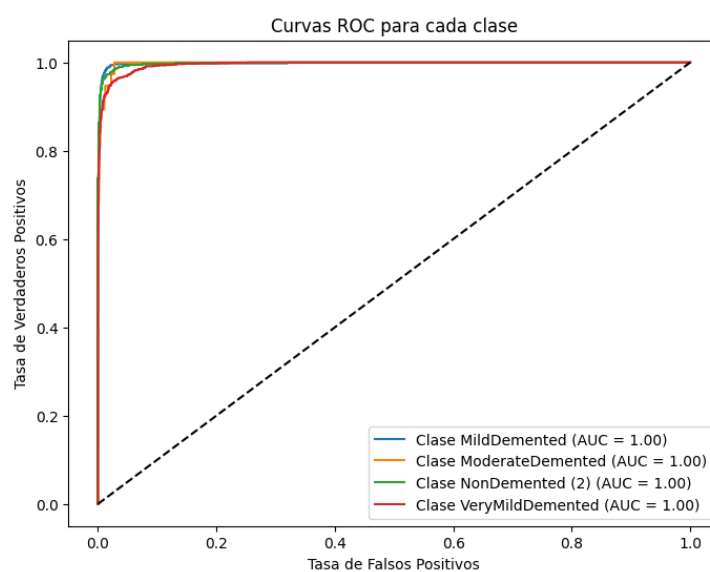


Figura 10: Curva ROC multiclase del modelo basado en el segundo artículo de Sivaranjani et al. [43].

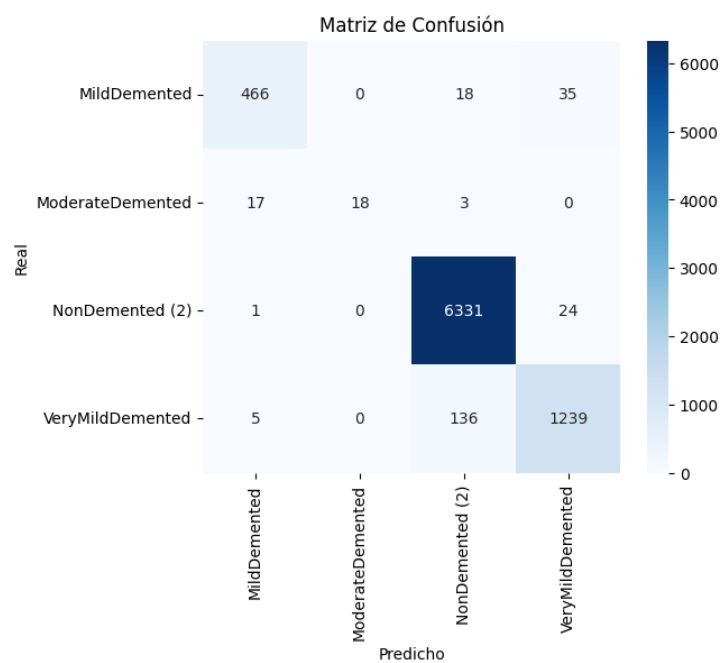


Figura 11: Matriz de confusión del modelo basado en el segundo artículo de Sivaranjani et al. [43].

del conjunto de datos OASIS y logró una precisión superior al 99 %. Dado que en este trabajo se aborda un problema multiclase, la arquitectura original fue adaptada para predecir cuatro clases distintas en lugar de solo dos.

Cuadro 7: Estructura del modelo propuesto por Al-Khuzai et al. [44].

Capa	Tamaño de salida	Parámetros
Conv2D (3×3 , 64)	$192 \times 192 \times 64$	15,616
MaxPooling2D	$96 \times 96 \times 64$	0
Conv2D (6×6 , 64)	$90 \times 90 \times 64$	200,768
MaxPooling2D	$45 \times 45 \times 64$	0
Conv2D (5×5 , 64)	$41 \times 41 \times 64$	102,464
MaxPooling2D	$20 \times 20 \times 64$	0
Conv2D (5×5 , 32)	$16 \times 16 \times 32$	51,232
MaxPooling2D	$8 \times 8 \times 32$	0
Conv2D (3×3 , 32)	$6 \times 6 \times 32$	9,248
MaxPooling2D	$3 \times 3 \times 32$	0
Flatten	288	0
Dense (121)	121	34,969
Dropout (0.5)	121	0
Dense (1)	1	122

El modelo se estructura en cinco bloques convolucionales seguidos de capas de max pooling, permitiendo una extracción jerárquica de características espaciales. Posteriormente, se incluye una capa de aplanamiento y dos capas densas separadas por una operación de *dropout* para mitigar el sobreajuste. Aunque el trabajo original se centraba en clasificación binaria, en esta implementación se ha adaptado la arquitectura al problema de clasificación multiclase que aborda este estudio.

4.2. Versión 1 – Clasificación por paciente

A pesar de que los modelos desarrollados en la Versión 0 arrojan resultados aparentemente excelentes —con precisiones cercanas al 97 % en algunos casos—, dichos valores resultan sospechosamente altos dadas las condiciones del entrenamiento. Al revisar en detalle el proceso de partición de los datos, se identifica una falla crítica en la metodología: **las imágenes (slices) no se organizan por paciente. Esto implica que cortes correspondientes a un mismo sujeto pueden aparecer tanto en el conjunto de entrenamiento como en los de validación y prueba.** Dado que cada paciente puede aportar alrededor de 200 imágenes, el modelo tiende a aprender características específicas del individuo en lugar de las asociadas a la

patología. Esta fuga de información compromete seriamente la validez de las métricas obtenidas, ya que el modelo no generaliza, sino que memoriza patrones particulares de los pacientes presentes en múltiples subconjuntos. Este hallazgo lleva a una reorganización completa del conjunto de datos y da paso a la **Versión 1**, en la que se garantiza una segmentación estricta por paciente en la partición de los datos.

1. Organización por paciente

Con el fin de evitar problemas de fuga de información entre los subconjuntos de entrenamiento, validación y prueba, se reorganiza el conjunto de datos original para que **cada paciente esté presente exclusivamente en uno de estos subconjuntos**. Esta medida busca prevenir que el modelo memorice características particulares de un mismo paciente distribuidas entre distintas fases del entrenamiento y evaluación.

El conjunto de datos empleado está compuesto por imágenes de resonancia magnética estructural (slices) distribuidas en carpetas según la clase diagnóstica: *MildDemented*, *ModerateDemented*, *VeryMildDemented* y *NonDemented*. Cada imagen presenta una nomenclatura que permite extraer el identificador del paciente, por ejemplo:

OAS1_0028_MR1_1.nii_slice_113, donde OAS1_0028_MR1 corresponde al ID del paciente.

Se recopila esta información en un *DataFrame* que incluye la ruta de la imagen, su etiqueta y el identificador del paciente. El total de pacientes únicos en el conjunto de datos es de **436**. La división se realiza de forma estratificada en función de la etiqueta diagnóstica de cada paciente, utilizando un **80 % para entrenamiento, 10 % para validación y 10 % para prueba**.

El tamaño final de cada subconjunto de imágenes es el siguiente:

- Entrenamiento: **52,546** imágenes
- Validación: **13,152** imágenes
- Prueba: **17,218** imágenes

Para facilitar el procesamiento del modelo, se normalizan las etiquetas diagnósticas mediante una codificación numérica adecuada.

La evaluación del rendimiento tras la reorganización del conjunto de datos por paciente se lleva a cabo utilizando la arquitectura propuesta en el trabajo de Sivaranjani et al. (*A CNN based framework for classification of Alzheimer's disease*) [43], dado que fue el modelo que mostró mejores resultados en la versión anterior.

Una vez implementada esta organización por paciente, se observa una notable caída en el rendimiento del modelo. La matriz de confusión (Figura 12) muestra una alta confusión

entre las clases *MildDemented* y *VeryMildDemented*, así como la completa ausencia de predicciones para la clase *ModerateDemented*.

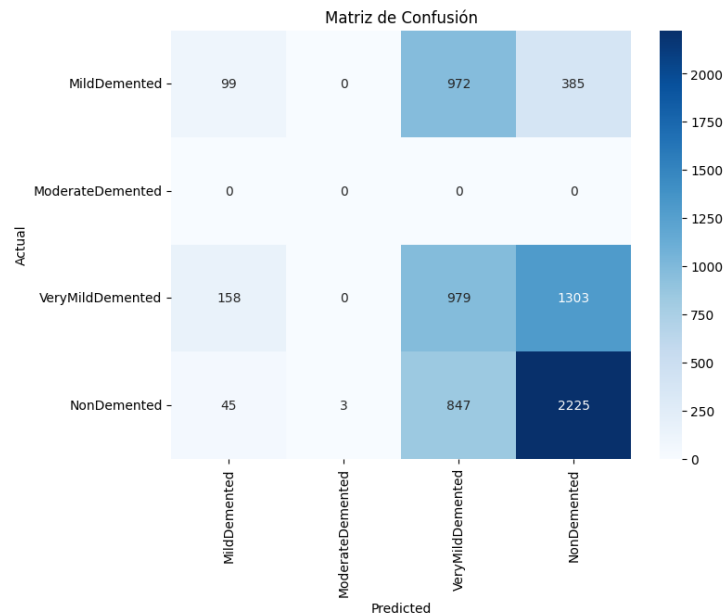


Figura 12: Matriz de confusión tras la organización por paciente del modelo de Sivaranjani et al.

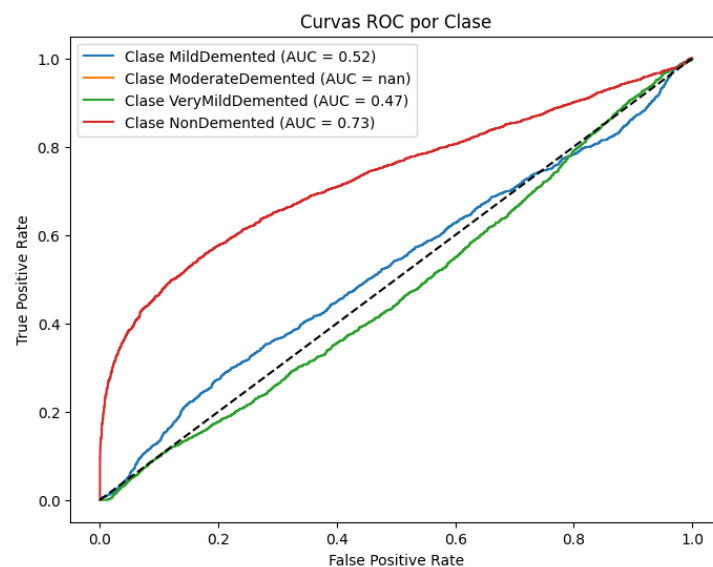


Figura 13: Curva ROC tras la organización por paciente del modelo Sivaranjani et al.

Asimismo, las puntuaciones del área bajo la curva ROC (AUC) obtenidas para cada clase son considerablemente más bajas en comparación con configuraciones anteriores como se observa en la figura 13.

Estos resultados evidencian la complejidad del problema y la importancia de una partición adecuada por paciente para evaluar de manera fiable la capacidad de generalización de los modelos en el ámbito del diagnóstico médico mediante imágenes.

Para de preservar el mejor estado del modelo durante el entrenamiento, se implementa la técnica de **early stopping** con una paciencia de 7 épocas, que permite guardar los pesos correspondientes a la época con menor pérdida de validación (*val_loss*). , ya en la primera época se alcanza el valor mínimo de *val_loss* (0.7928) y una precisión de validación del 75.73 %. A pesar de continuar el entrenamiento durante varias épocas más, los valores de pérdida y precisión en validación no mejoraron, lo cual no resulta habitual y sugiere posibles problemas de sobreajuste temprano o falta de capacidad de generalización del modelo bajo la nueva partición por paciente. Finalmente, el modelo se detuvo automáticamente en la época 8 tras no observar mejoras, y se guardaron los pesos correspondientes a la primera época.

2. Eliminación de la clase minoritaria

Ante el comportamiento inusual del modelo, se consideró que la causa podría estar relacionada con un desbalance significativo entre las clases presentes en el conjunto de datos. El recuento de pacientes por categoría diagnóstica revela una distribución altamente desigual:

- **NonDemented (2):** 336 pacientes
- **VeryMildDemented:** 70 pacientes
- **MildDemented:** 28 pacientes
- **ModerateDemented:** 2 pacientes

Esta última categoría, al contar con un número tan reducido de pacientes, imposibilita una partición adecuada en entrenamiento, validación y prueba, dificultando la capacidad del modelo para aprender patrones representativos de dicha clase. Por este motivo, **se decide eliminar la clase *ModerateDemented* del problema de clasificación.**

Una vez ajustado el conjunto de datos, se reutiliza la arquitectura propuesta en *A CNN based framework for classification of Alzheimer's disease* [43] para evaluar el rendimiento del modelo bajo esta nueva configuración triclase. El mejor modelo se obtiene en la época 3, con una pérdida de validación de 0.4310 y una precisión de prueba del 75.43 %. La curva ROC para cada clase también muestra mejoras respecto a la configuración con cuatro clases 14, con un AUC de 0.85 para *MildDemented*, 0.76 para *VeryMildDemented* y 0.86 para *NonDemented*. No obstante, persiste la tendencia observada previamente: **el**

modelo alcanza su mejor rendimiento en etapas tempranas del entrenamiento, lo cual podría indicar problemas de capacidad de generalización o sobreajuste rápido a los datos de entrenamiento.

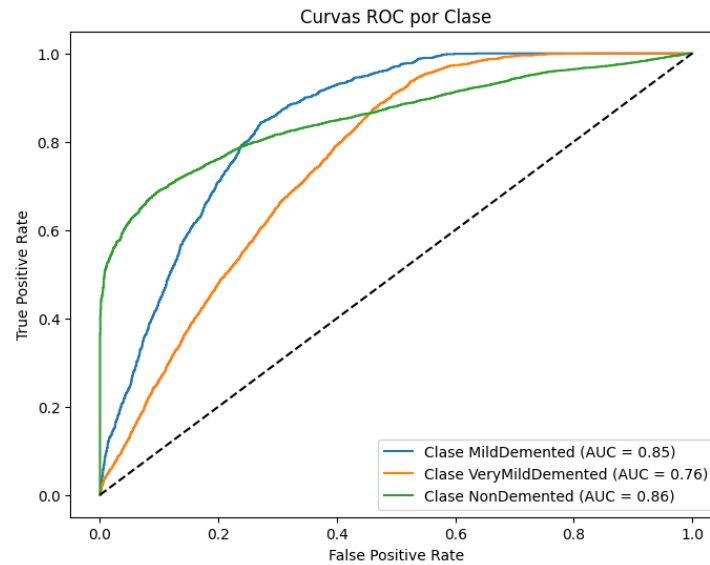


Figura 14: Curva ROC tras la eliminación de la clase *ModerateDemented* del modelo Sivaranjani et al.

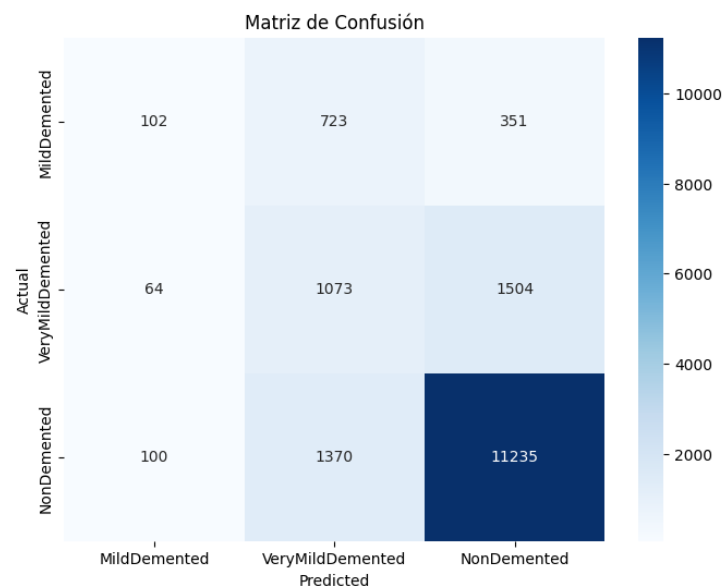


Figura 15: Matriz de confusión tras la eliminación de la clase *ModerateDemented* del modelo Sivaranjani et al.

También se exploró el uso de la arquitectura propuesta en el trabajo *A Novel Framework for Classification of Different Alzheimer's Disease Stages Using CNN Model* [45]. Este modelo está diseñado específicamente para la clasificación de las distintas etapas del

Alzheimer y propone una red convolucional personalizada cuya estructura se muestra en la Figura 8.

Cuadro 8: Estructura del modelo propuesto en Dar et al. [45].

Layer	Output Shape	Parameters
Input (Input Layer)	(None, 224, 224, 3)	0
Conv1 (Conv2D)	(None, 112, 112, 32)	864
BatchNorm	(None, 112, 112, 32)	128
ReLU	(None, 112, 112, 32)	0
DepthwiseConv2D	(None, 112, 112, 32)	288
BatchNorm	(None, 112, 112, 32)	128
ReLU	(None, 112, 112, 32)	0
PointwiseConv2D	(None, 112, 112, 64)	2048
BatchNorm	(None, 112, 112, 64)	256
ReLU	(None, 112, 112, 64)	0
ZeroPadding2D	(None, 113, 113, 64)	0
DepthwiseConv2D	(None, 56, 56, 64)	576
BatchNorm	(None, 56, 56, 64)	256
ReLU	(None, 56, 56, 64)	0
PointwiseConv2D	(None, 56, 56, 128)	8192
BatchNorm	(None, 56, 56, 128)	512
ReLU	(None, 56, 56, 128)	0
DepthwiseConv2D	(None, 56, 56, 128)	1152
Flatten	(None, 50176)	0
Dense	(None, 512)	25,690,624
BatchNorm	(Batch, 512)	2048
Dense	(None, 512)	262,656
Dropout	(None, 512)	0
Dense	(None, 5)	2565
Total params: 29,190,853 (25,958,917 trainables)		

En este trabajo, se replicó dicha arquitectura utilizando como base la arquitectura MobileNetV2 preentrenada sobre ImageNet. Esta adaptación se implementó con el objetivo de conservar la eficiencia y ligereza del modelo original. Se mantuvieron las capas convolucionales de MobileNetV2 como extractor de características, sustituyendo únicamente la parte final del clasificador por un bloque de capas totalmente conectadas inspirado en la estructura descrita en el artículo. Se aplicaron ligeras modificaciones en los tamaños

de entrada y dimensiones intermedias para ajustarse a las características del conjunto de datos utilizado y al número de clases definidas en esta experimentación (tres clases en lugar de cinco).

Durante la evaluación inicial de este modelo con tres clases, se observó nuevamente el patrón anómalo en el que la mejor validación se daba en la primera época: `val_loss: 0.5698`, con una precisión de validación del 78.08 %. A partir de esa época, las pérdidas de validación fueron aumentando progresivamente, y aunque se ejecutaron 14 épocas, se decidió detener la prueba completa debido a la alta duración del entrenamiento y al estancamiento en la mejora.

3. Data augmentation

Con el objetivo de analizar si el comportamiento observado en los entrenamientos anteriores —donde el mejor rendimiento se producía en las primeras épocas— podía estar relacionado con el marcado desequilibrio entre clases, se procedió a aplicar técnicas de *data augmentation* de manera selectiva. En concreto, se aumentaron artificialmente las clases *MildDemented* y *VeryMildDemented* como se observa en el cuadro 9. La clase *NonDemented*, con 63.560 imágenes, se mantuvo sin modificaciones. El conjunto de datos se dividió en entrenamiento (55.458 imágenes), validación (13.397) y prueba (17.705), buscando una mayor uniformidad entre clases sin introducir imágenes sintéticas en exceso.

Cuadro 9: Comparación del número de imágenes por clase antes y después del *data augmentation*.

Clase	Antes del aumento	Después del aumento
<i>NonDemented</i>	63.560	63.560
<i>VeryMildDemented</i>	13.796	15.000
<i>MildDemented</i>	5.184	8.000

Para esta fase experimental se continuó utilizando la arquitectura basada en el trabajo de Sivaranjani et al. [43], dado que había mostrado un rendimiento más estable en pruebas anteriores.

El modelo fue entrenado nuevamente bajo las mismas condiciones experimentales, observándose que, aunque se produjo una ligera mejora en las primeras épocas (con mejor modelo en la época 2 con una *validation loss* mínima de 0.5247 y una *validation accuracy* de 0.7582), el comportamiento general persistía: las mejores métricas de validación se obtenían de forma temprana, y el sobreajuste aumentaba progresivamente en épocas posteriores.

Con el objetivo de seguir mejorando la capacidad del modelo para generalizar en presencia de clases desbalanceadas, se optó por continuar usando la arquitectura empleada en el experimento anterior con *data augmentation*, incorporando ahora una función de **pérdida ponderada**. Para ello, se asignaron pesos inversamente proporcionales al número de muestras por clase en el conjunto de entrenamiento, con la intención de penalizar con mayor intensidad los errores en las clases minoritarias (*MildDemented* y *VeryMildDemented*).

Los resultados no mostraron una mejora sustancial con respecto al experimento sin pesos. Por ejemplo, aunque el mejor modelo se obtuvo también en la época 2, con una *validation loss* de 0.5811 y una precisión de validación del 70.77 %, estos valores no superaron a los obtenidos anteriormente con solo *data augmentation* (donde se alcanzó una *validation accuracy* de 75.82 %). Esto indica que, si bien la ponderación ayudó a atenuar el desbalance en las primeras etapas del entrenamiento, no fue suficiente por sí sola para corregir el patrón observado de sobreajuste temprano. Por tanto, se refuerza la hipótesis de que el desequilibrio de clases no es el único desafío presente y que podrían ser necesarias estrategias adicionales.

4. Clasificación binaria

Para simplificar el problema y explorar si una reformulación binaria podía mejorar el rendimiento del modelo, se entrenó nuevamente la arquitectura del artículo de *A CNN based framework for classification of Alzheimer's disease* [43] bajo un enfoque de clasificación binaria. Para ello, se agruparon las clases *MildDemented*, *VeryMildDemented* y *ModerateDemented* bajo la categoría “Alzheimer”, mientras que la clase *NonDemented* se mantuvo como “NonAlzheimer”. Esta nueva configuración permitió evaluar la capacidad del modelo para distinguir entre presencia y ausencia de signos de demencia, sin diferenciar entre niveles de severidad.

El entrenamiento se realizó durante 25 épocas, con un comportamiento notablemente similar al observado en los experimentos anteriores: el modelo alcanzó su mejor rendimiento en validación de forma muy temprana, concretamente en la época 1, con una *validation loss* de 0.4032 y una precisión del 78.41 %. A pesar de que la pérdida de entrenamiento continuó mejorando en las épocas sucesivas (hasta alcanzar un valor de 0.0485 con un *training accuracy* de 98.16 % en la época 25), la pérdida de validación aumentó progresivamente, alcanzando un valor final de 1.1283. Esta divergencia entre entrenamiento y validación evidencia nuevamente un patrón de sobreajuste como se observa en la figura 16.

El rendimiento en el conjunto de prueba fue razonablemente bueno, con una precisión de 85.25 % y un *AUC* sobre la curva ROC de 0.92 (Figura 17), por lo que el modelo logra una

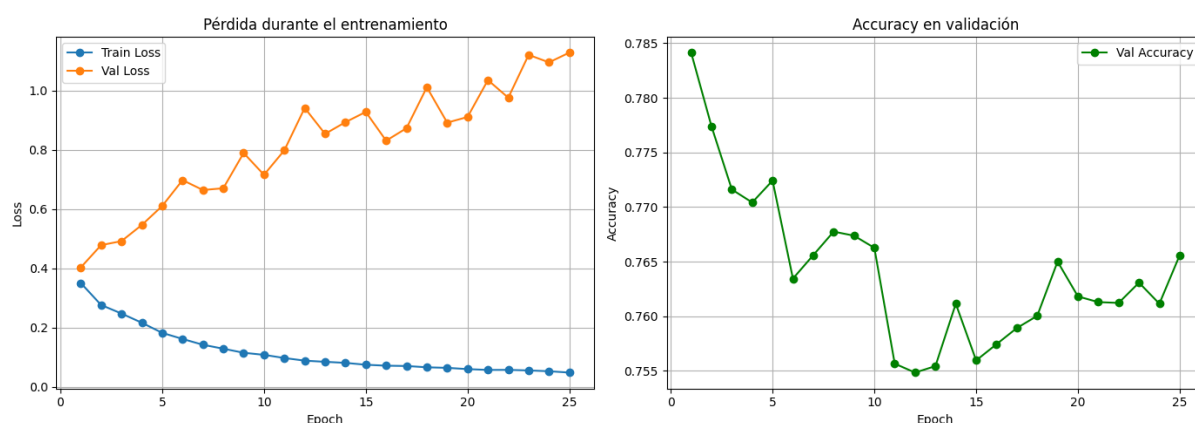


Figura 16: Gráficas de pérdida y precisión del modelo binario (*Alzheimer* vs *NonAlzheimer*) del artículo de Sivaranjani et al.[43]

buena capacidad de discriminación entre ambas clases a pesar del sobreajuste observado en validación.

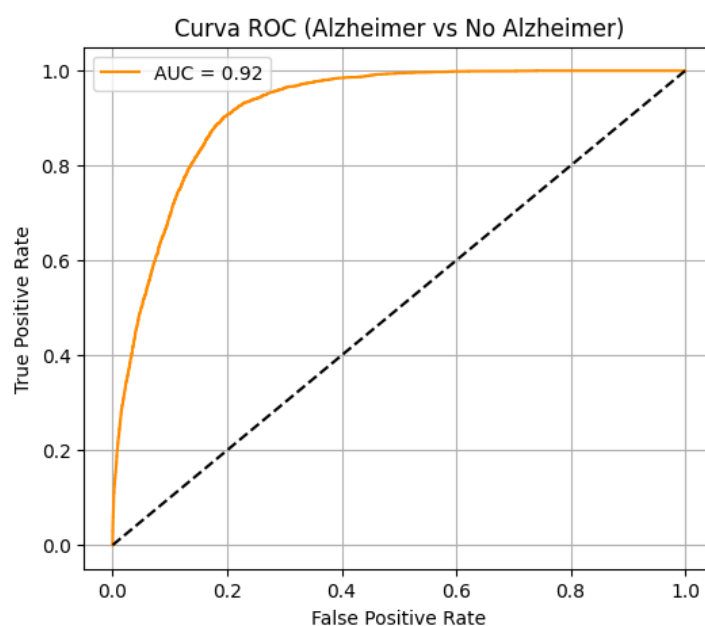


Figura 17: Curva ROC del modelo binario (*Alzheimer* vs *NonAlzheimer*) basado en el artículo de Sivaranjani et al. [43].

La matriz de confusión presentada en la Figura 18 muestra una mayor tasa de aciertos para la clase “NonAlzheimer” (11,490 verdaderos negativos) en comparación con la clase “Alzheimer” (2,674 verdaderos positivos), aunque también se observa una cantidad no despreciable de falsos negativos (1,143), lo cual es crítico en este tipo de tareas clínicas.

Aunque la reformulación binaria permitió obtener métricas de clasificación más eleva-

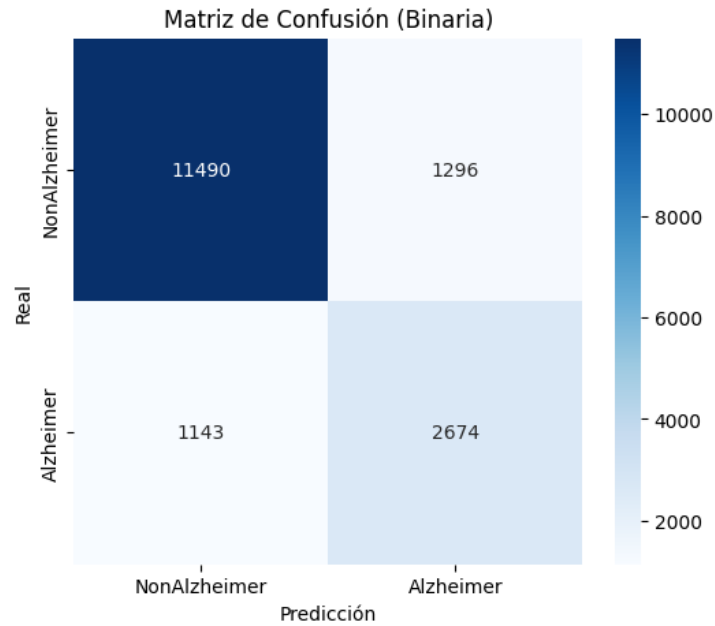


Figura 18: Matriz de confusión del modelo binario (*Alzheimer* vs *NonAlzheimer*) basado en el artículo de Sivaranjani et al. [43].

das en términos absolutos, especialmente en precisión de prueba y AUC, el patrón de sobreajuste persistió. Esto sugiere que el problema principal no reside únicamente en la cantidad de clases, sino posiblemente en aspectos como la calidad de los datos, el desbalance residual o la necesidad de técnicas de regularización adicionales.

4.3. Versión 2 – Reducción del dataset y modelos base

Con el fin de explorar nuevas estrategias para mejorar la generalización y reducir el sobreajuste observado en los experimentos anteriores, se propuso una segunda versión del enfoque experimental basada en dos pilares fundamentales: (i) la reducción controlada del tamaño del conjunto de datos y (ii) la utilización de arquitecturas de referencia ampliamente contrastadas en la literatura.

Por un lado, la reducción del dataset se planteó como una forma de forzar un escenario más equilibrado y manejable, permitiendo realizar experimentos más rápidos y homogéneos. Para ello, se realizaron **dos configuraciones** diferentes: una primera con un máximo de 1,000 imágenes por clase y una segunda con hasta 5,000 imágenes por clase.

Por otro lado, **se introdujeron como modelos base varias arquitecturas conocidas y preentrenadas en ImageNet: ResNet-18, MobileNetV2 y AlexNet;** y también se incluyó como referencia el modelo propio desarrollado en versiones anteriores (modelo del artículo *A CNN based framework for classification of Alzheimer’s disease* [43] de Sivaranjani et al.). Con este

enfoque se evalúa la capacidad de estos modelos para transferir conocimiento a la tarea de clasificación de Alzheimer a partir de imágenes de resonancia magnética, así se compara su comportamiento respecto a la arquitectura personalizada utilizada en las versiones anteriores.

1. Experimento con 1000 imágenes por clase.

En esta primera variante, se seleccionaron hasta un máximo aproximado de **1000 imágenes por clase**, respetando la estructura por paciente. Para ello, se implementó un procedimiento iterativo que priorizaba la inclusión completa de imágenes correspondientes a pacientes individuales, evitando que un mismo paciente quedara fraccionado entre conjuntos de entrenamiento, validación o prueba.

Concretamente, el proceso consistía en recorrer de manera secuencial los pacientes dentro de cada clase y seleccionar sus imágenes completas siempre que no se excediera el límite de imágenes deseado.

Tras aplicar este criterio de selección, el conjunto resultante quedó distribuido de la siguiente manera:

- **Tamaño de entrenamiento:** 1544 imágenes
- **Tamaño de validación:** 728 imágenes
- **Tamaño de prueba:** 488 imágenes

La distribución por clase obtenida fue la siguiente:

Cuadro 10: Distribución del número de imágenes por clase con máximo de 1000 imágenes.

Clase	Número de imágenes
MildDemented	952
NonDemented	936
VeryMildDemented	872

Para garantizar consistencia en las comparaciones, se empleó un **batch size de 128** en todos los modelos evaluados en esta etapa.

Al analizar la evolución de la *train accuracy* (Figura 19), se observa que los modelos *MobileNet* y *ResNet* alcanzan rápidamente una precisión cercana al 100 %, lo cual sugiere una alta capacidad de ajuste sobre los datos de entrenamiento. Por el contrario, el modelo Sivaranjani-CNN (modelo del artículo de Sivaranjani et al.) y *AlexNet* presentan un aprendizaje más progresivo, con precisiones finales notablemente inferiores.

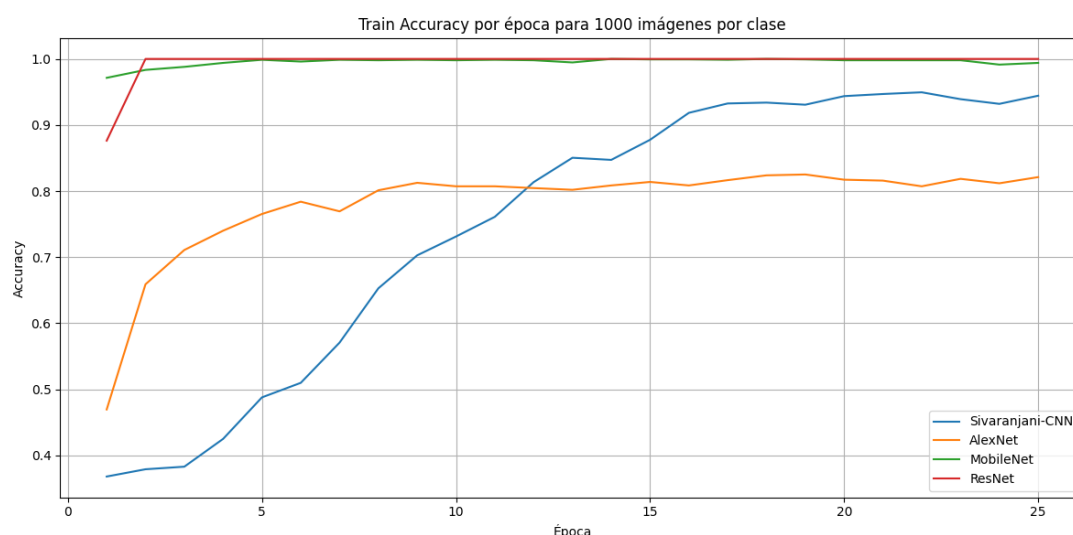


Figura 19: Evolución de la *train accuracy* durante el entrenamiento para los distintos modelos evaluados con 1000 imágenes por clase.

Sin embargo, al examinar la *validation accuracy* (Figura 20), se aprecia un comportamiento diferente. Mientras que *Sivanranjani-CNN* y *ResNet* logran mejores valores de precisión en validación, *MobileNet* y especialmente *AlexNet* muestran un desfase considerable entre precisión en entrenamiento y validación.

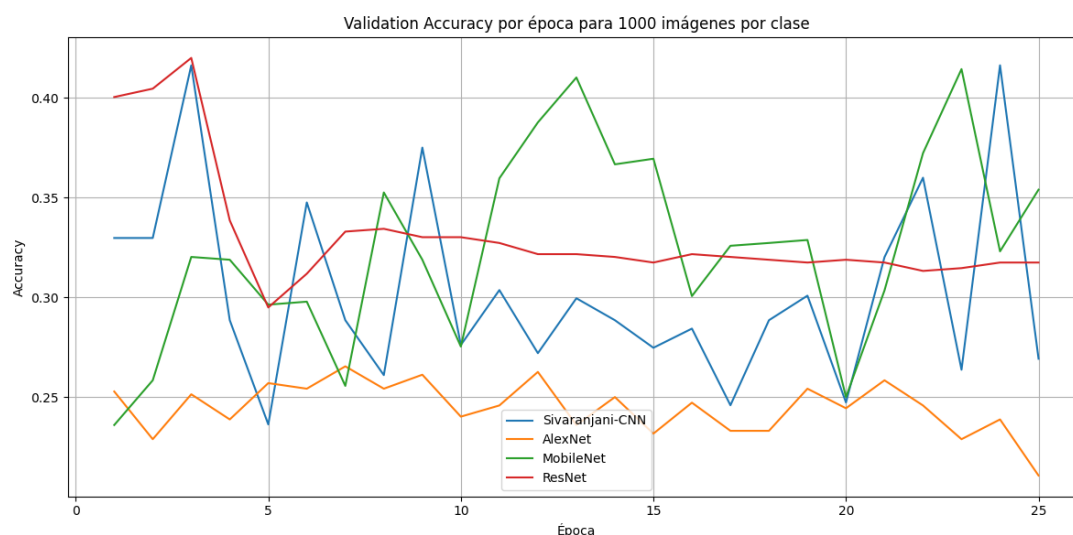


Figura 20: Evolución de la *validation accuracy* durante el entrenamiento para los distintos modelos evaluados con 1000 imágenes por clase.

El análisis del *F1-score* en validación (Figura 21) refuerza estas conclusiones. Los mejores valores de *F1-score* se alcanzan nuevamente con *Sivanranjani-CNN* y con *ResNet* (aunque alcanzado en época temprana), mientras que *MobileNet*, a pesar de su elevada precisión

en entrenamiento, muestra resultados más modestos en este aspecto. *AlexNet*, por su parte, obtiene los valores de *F1-score* más bajos entre los modelos evaluados.

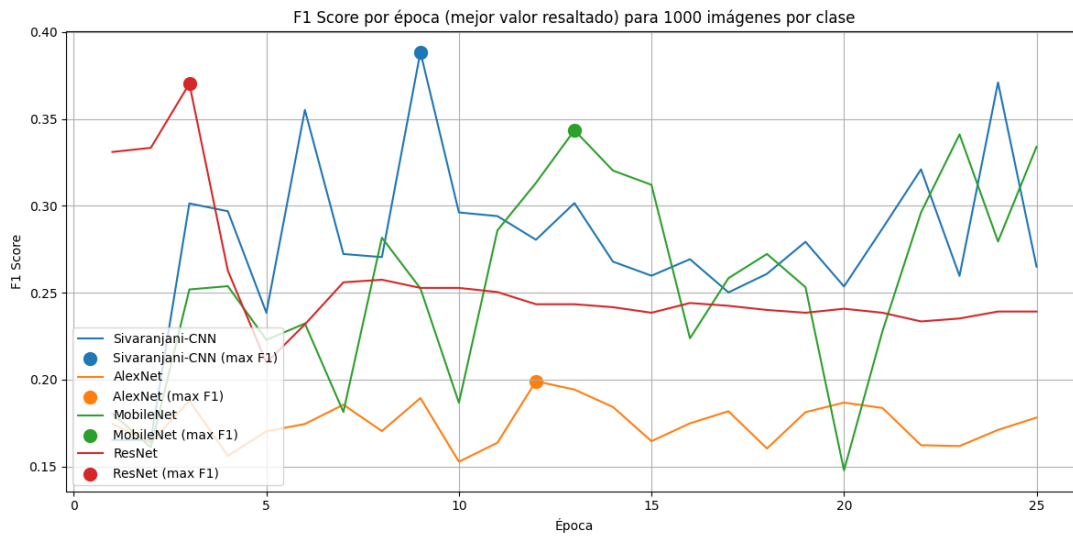


Figura 21: Evolución del *F1-score* en validación para los distintos modelos evaluados. Se destaca el mejor valor alcanzado por cada modelo.

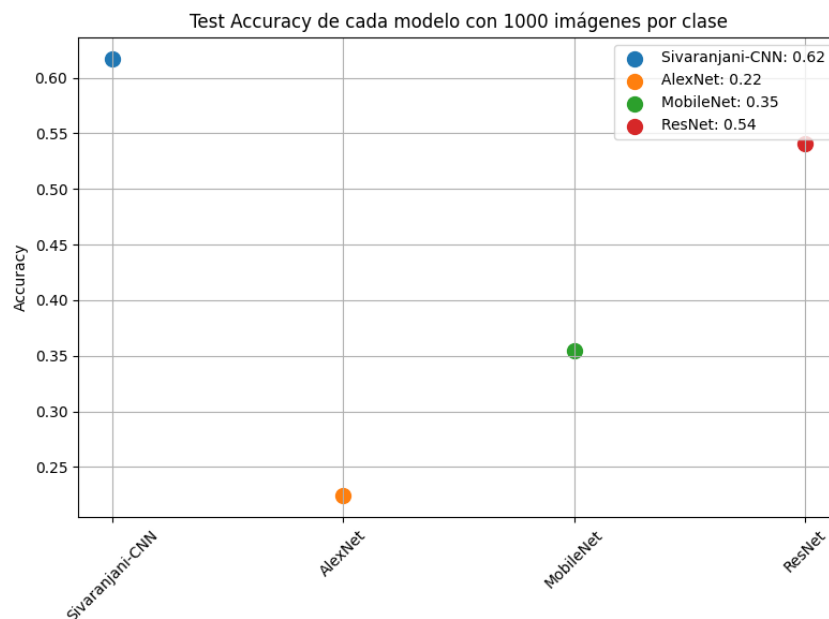


Figura 22: Precisión en el conjunto de test para cada modelo evaluado con 1000 imágenes por clase.

Finalmente, la Figura 22 resume el rendimiento global en el conjunto de test. El modelo *Sivaranjani-CNN* obtiene la mayor precisión en test, seguido de *ResNet*. Los modelos *MobileNet* y *AlexNet* muestran un rendimiento claramente inferior en este conjunto, confirmando las observaciones realizadas anteriormente.

En conjunto, estos resultados sugieren que *Sivaranjani-CNN* y *ResNet* presentan un mejor compromiso entre capacidad de aprendizaje y generalización en este escenario reducido de 1000 imágenes por clase, mientras que *MobileNet* y *AlexNet* tienden a sobreajustarse o a mostrar un aprendizaje subóptimo en validación y test.

2. Experimento con 5000 imágenes por clase

En esta segunda parte de la experimentación, se incrementó el número de imágenes por clase a un máximo de 5000. Al igual que en el caso anterior, la selección de imágenes se realizó respetando la separación por paciente, es decir, no se fragmentaron los pacientes entre los distintos conjuntos. De este modo, cada paciente completo se asigna únicamente a *train*, *validation* o *test*.

El conjunto final de datos quedó distribuido del siguiente modo:

- **Tamaño de entrenamiento:** 9430 imágenes
- **Tamaño de validación:** 2380 imágenes
- **Tamaño de prueba:** 3158 imágenes

Distribución por clase:

Cuadro 11: Distribución del número de imágenes por clase con máximo de 5000 imágenes.

Clase	Número de imágenes
MildDemented	5000
NonDemented	5000
VeryMildDemented	4968

Como en la experimentación anterior, se empleó un *batch size* de 128 en todos los entrenamientos. Se evaluaron los mismos cuatro modelos: *Sivaranjani-CNN*, *AlexNet*, *MobileNet* y *ResNet*, estos tres últimos ya **preentrenados**.

En la gráfica de **train accuracy** (23), se observa que todos los modelos son capaces de alcanzar altos niveles de precisión sobre el conjunto de entrenamiento. El modelo *Sivaranjani-CNN* presenta una progresión constante, alcanzando más del 90 % al final del entrenamiento. *AlexNet* muestra una estabilización en torno al 58–60 %, mientras que *MobileNet* y *ResNet* logran una precisión prácticamente perfecta (más del 99 % desde las primeras épocas).

La gráfica de precisión en validación (24) permite evaluar la capacidad de generalización de los modelos. *Sivaranjani-CNN* mantiene una tendencia estable, logrando precisiones superiores al 50 %. *AlexNet* también muestra una buena progresión y termina alcanzando

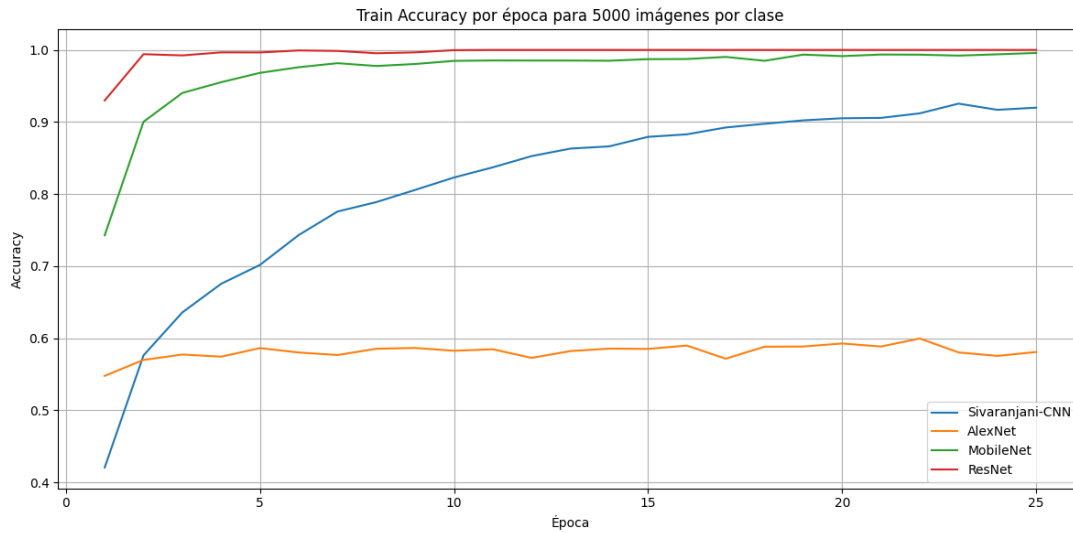


Figura 23: Evolución de la *train accuracy* durante el entrenamiento para los distintos modelos evaluados con 5000 imágenes por clase.

valores cercanos al 60 %. En cambio, *MobileNet* y *ResNet* presentan un comportamiento más inestable, con oscilaciones significativas en la precisión de validación, indicando dificultades para generalizar a datos no vistos.

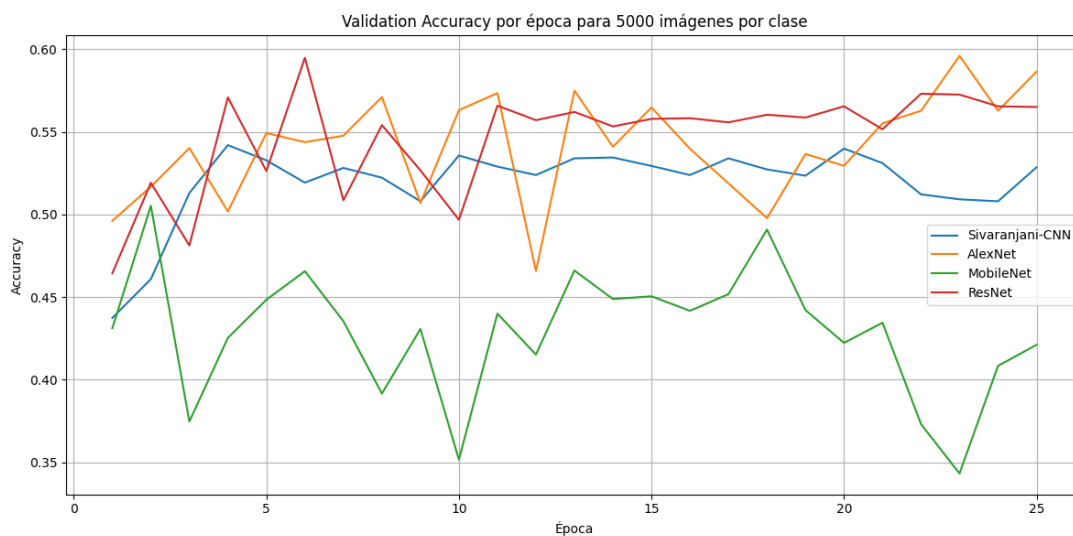


Figura 24: Evolución de la *validation accuracy* durante el entrenamiento para los distintos modelos evaluados con 5000 imágenes por clase.

En la gráfica de F1 score en validación (25), se observa un comportamiento coherente con la *validation accuracy*. *Sivanranjani-CNN* alcanza un *f1-score* máximo en torno a 0.54, mostrando una evolución consistente. *AlexNet* también presenta una progresión ascendente, con un *f1-score* que supera el 0.55 en las mejores épocas. Por el contrario,

MobileNet y ResNet evidencian grandes fluctuaciones en su $f1$ -score.

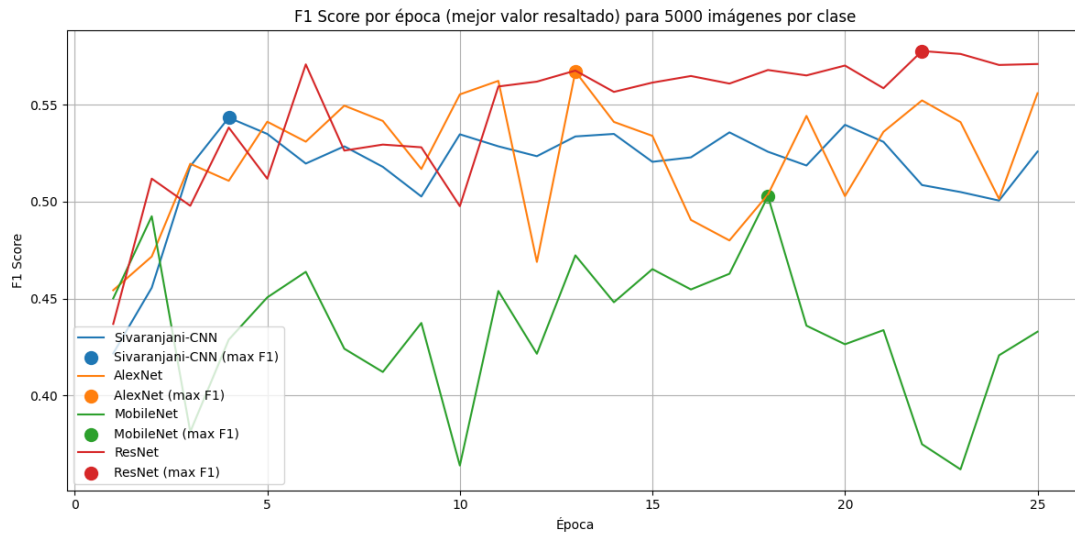


Figura 25: Evolución del $F1$ -score en validación para los distintos modelos evaluados. Se destaca el mejor valor alcanzado por cada modelo con 5000 imágenes por clase.

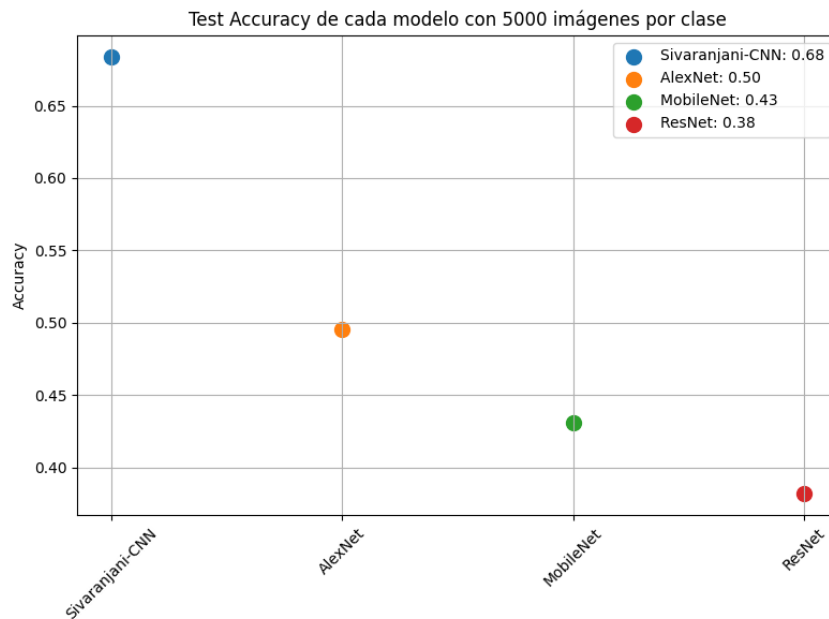


Figura 26: Precisión en el conjunto de test para cada modelo evaluado con 5000 imágenes por clase.

Por último, en el gráfico de test (26), que compara el rendimiento final en el conjunto de prueba, *Sivanranjani-CNN* destaca claramente con una precisión de **0.6837** y un $f1$ -score de **0.6799**, consolidándose como el mejor modelo en este escenario. *AlexNet* mejora sustancialmente respecto al experimento anterior, alcanzando una *test accuracy* de **0.4955**.

En cambio, *MobileNet* y *ResNet* no logran trasladar su alto rendimiento en entrenamiento al conjunto de prueba, obteniendo precisiones de **0.4311** y **0.3819** respectivamente.

Los resultados muestran que, en general, el incremento en el número de imágenes por clase permite obtener un mejor aprendizaje en comparación con el experimento anterior.

- **Sivaranjani-CNN**: presenta una evolución consistente, alcanzando una *test accuracy* de **0.6837** y un *f1-score* en test de **0.6799**, lo que representa una mejora notable frente al experimento con menos imágenes. Las curvas de entrenamiento y validación muestran también un comportamiento estable, con valores que se mantienen próximos entre sí.
- **AlexNet**: aunque la *train accuracy* se estabiliza en torno al 58–60 %, el modelo logra en test una *accuracy* de **0.4955** y un *f1-score* de **0.5003**. Las curvas de validación son más estables que en el primer experimento, aunque siguen mostrando cierta fluctuación.
- **MobileNet**: a pesar de un excelente ajuste en el conjunto de entrenamiento, el modelo muestra problemas de generalización. La *validation accuracy* y el *f1-score* presentan bastante variabilidad durante el entrenamiento, y el rendimiento en test cae a **0.4311** en *accuracy* y **0.4229** en *f1-score*, indicando sobreajuste.
- **ResNet**: el modelo alcanza un sobreajuste más evidente, con una precisión en entrenamiento perfecta, pero en test de solo **0.3819** y un *f1-score* de **0.3891**. Las curvas de validación son más altas que en el primer experimento, pero no logran traducirse en un buen rendimiento en el conjunto de prueba.

El aumento del tamaño del conjunto de datos a 5000 imágenes por clase ha permitido mejorar la estabilidad y el rendimiento de algunos modelos. *Sivaranjani-CNN* y *AlexNet* se benefician especialmente de este incremento, mostrando mejores curvas de validación y resultados en test. Sin embargo, *MobileNet* y *ResNet* continúan mostrando un fuerte sobreajuste.

A pesar de las mejoras observadas, es importante destacar que persiste una marcada diferencia entre la evolución de la precisión durante el entrenamiento y en la validación. Mientras que en entrenamiento las curvas muestran un avance progresivo y estable, **en la validación los valores presentan una alta variabilidad, con numerosos picos y caídas abruptas que resultan poco habituales en modelos de clasificación bien ajustados**. Este comportamiento sugiere que sigue habiendo margen de optimización en términos de regularización y robustez de los modelos evaluados.

4.4. Versión 3 – Imagen central por paciente

Los resultados observados en los experimentos anteriores (tanto con 1000 como con 5000 imágenes por clase) muestran patrones de sobreajuste temprano, así como una elevada variabilidad en las métricas de validación. Este comportamiento resulta llamativo dado que el número de imágenes por clase se mantuvo equilibrado y controlado en ambas versiones, lo cual sugiere que otras posibles causas podrían estar afectando a la capacidad de generalización de los modelos.

Una hipótesis planteada es que la propia naturaleza de las imágenes utilizadas podría estar contribuyendo a este fenómeno. Al tratarse de cortes axiales (*slices*) extraídos de estudios de resonancia magnética estructural tridimensional (MRI 3D), es probable que **múltiples imágenes provenientes de un mismo paciente sean visualmente muy similares entre sí**. De ser así, el conjunto de entrenamiento incluiría un alto grado de redundancia, limitando la diversidad efectiva de las muestras y facilitando el sobreajuste.

En la Figura 27, se muestra un ejemplo representativo de 25 *slices* consecutivos pertenecientes al mismo paciente. Se puede observar que las imágenes presentan una gran similitud visual, lo que refuerza la hipótesis de que esta redundancia podría estar afectando negativamente al aprendizaje de los modelos.

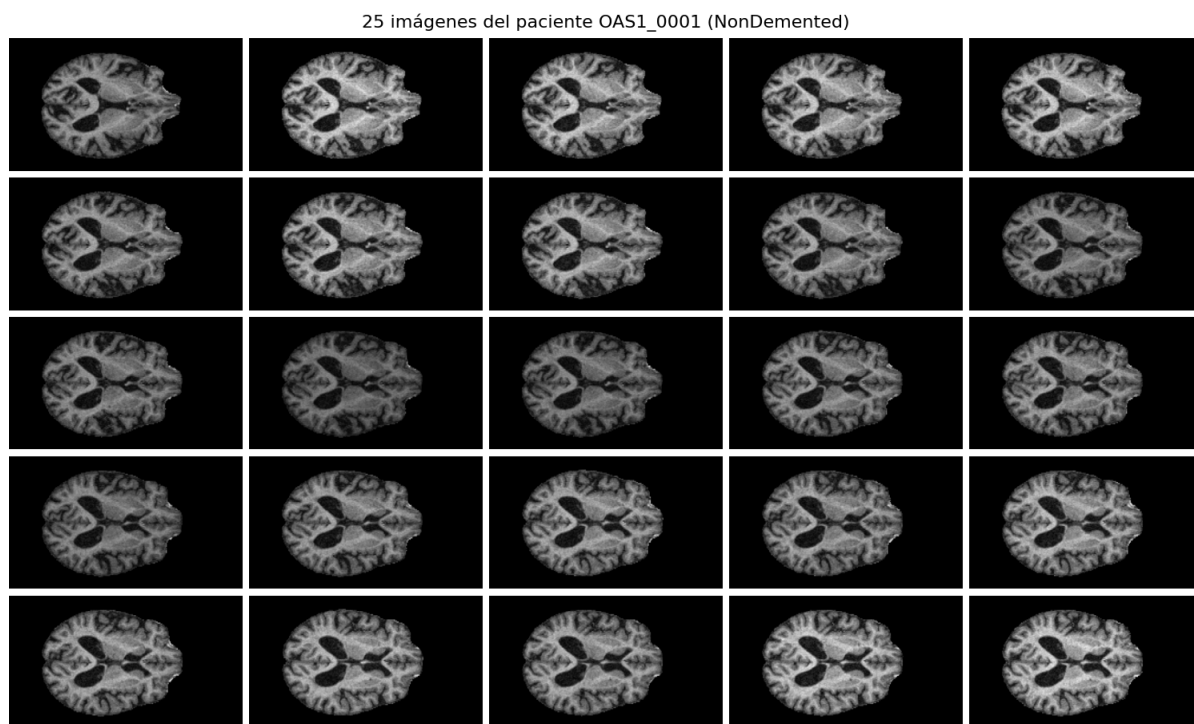


Figura 27: Ejemplo de 25 *slices* consecutivos del paciente OASIS_001. Se aprecia la alta similitud entre los cortes, lo que puede introducir redundancia en el conjunto de datos.

Para mitigar este problema, en esta tercera versión se ha adoptado una estrategia dife-

rente: seleccionar únicamente el **slice central** de cada paciente. Esta decisión se basa en la consideración de que los cortes más centrales suelen contener mayor cantidad de información anatómica relevante, mientras que los cortes más extremos (al principio y al final del volumen 3D) a menudo muestran áreas con escaso o nulo contenido cerebral. De este modo, se busca aumentar la diversidad del conjunto de datos y reducir la redundancia derivada de la inclusión de múltiples imágenes muy similares de un mismo individuo.

La Tabla 12 muestra el número de imágenes por clase y por conjunto (entrenamiento, validación y prueba):

Set	MildDemented	NonDemented	VeryMildDemented
Train	17	215	45
Validation	5	54	11
Test	6	67	14

Cuadro 12: Número de imágenes por clase y conjunto en la versión con imagen central por paciente.

En la Figura 28, se muestran ejemplos ilustrativos de **5 imágenes centrales** correspondientes a pacientes diferentes de cada clase. Esta visualización permite apreciar visualmente la variabilidad presente entre las clases.

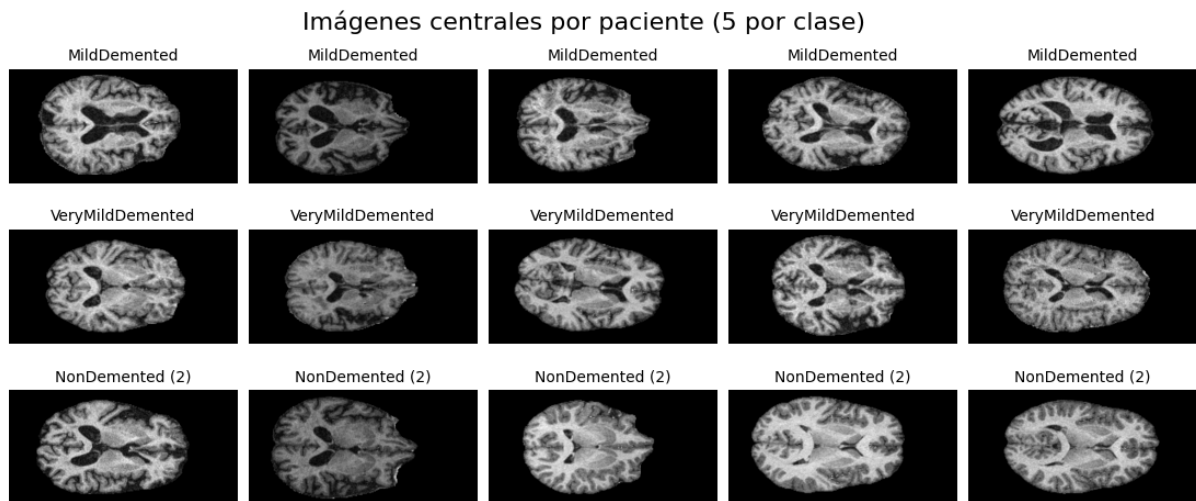


Figura 28: Ejemplo de 5 *slices* centrales de pacientes diferentes por clase.

Como en las versiones anteriores, en esta configuración se evaluaron los modelos de ImageNet (*AlexNet*, *MobileNet*, *ResNet*) y el modelo *Sivaranjani-CNN* probado hasta ahora.

En la gráfica de precisión durante el entrenamiento (Figura 29) se puede observar que *Sivaranjani-CNN*, *MobileNet* y *ResNet* muestran una curva ascendente de *train accuracy* durante las primeras épocas, mientras que *AlexNet* se mantiene completamente constante en todo

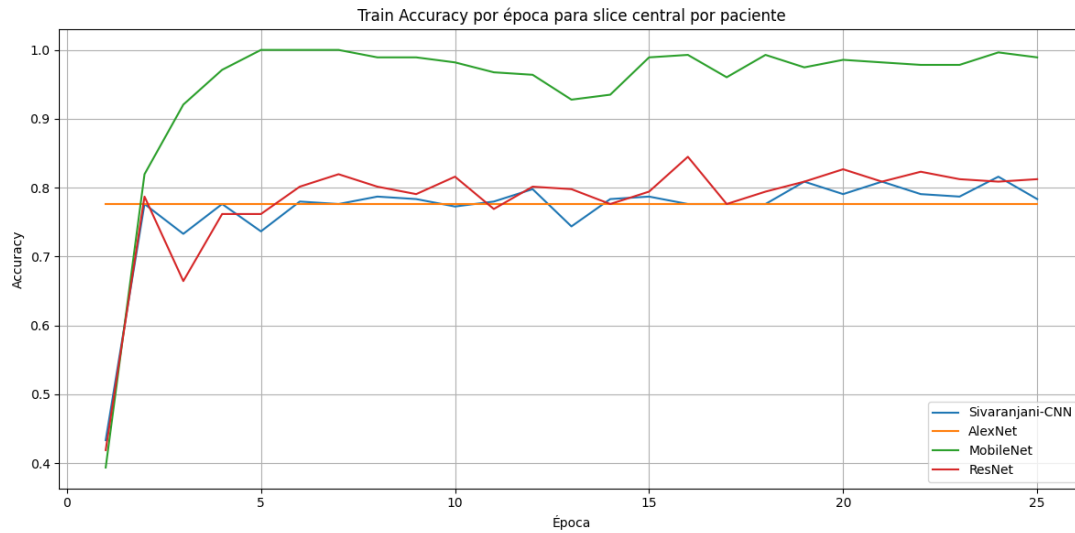


Figura 29: Evolución de la *train accuracy* durante el entrenamiento para los distintos modelos evaluados con una imagen central por paciente.

el entrenamiento. *MobileNet* logra alcanzar un *train accuracy* cercano al 100 %, indicando un claro sobreajuste al conjunto de entrenamiento. *ResNet* y *Sivanranjani-CNN* alcanzan valores elevados pero sin llegar a saturarse.

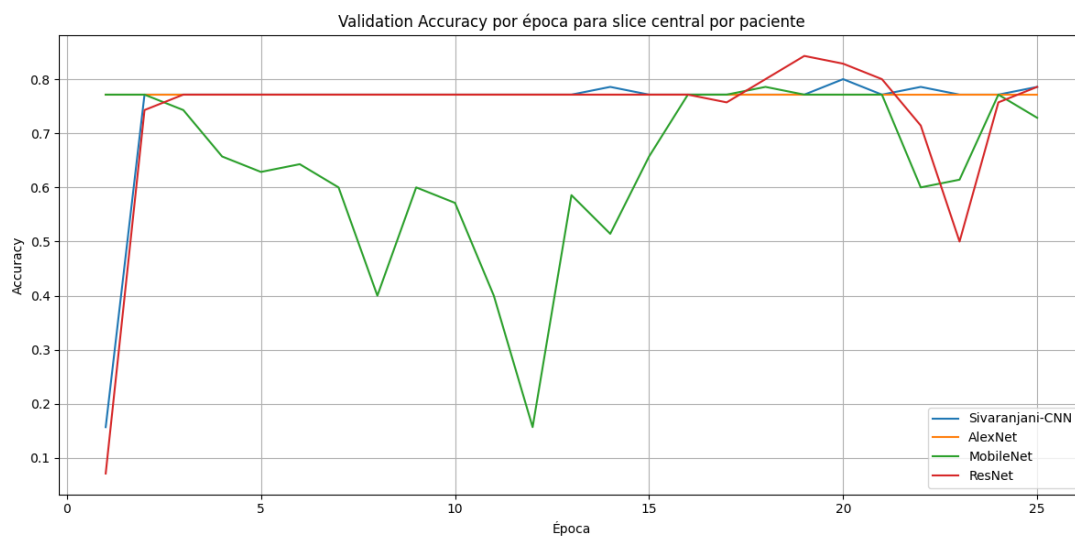


Figura 30: Evolución de la *validation accuracy* durante el entrenamiento para los distintos modelos evaluados con una imagen central por paciente.

En gráfica de precisión durante validación que se observa en la figura 30 se aprecia nuevamente un comportamiento particular en *AlexNet*, que permanece constante durante todo el entrenamiento, lo que sugiere que no está aprovechando la nueva configuración del conjunto de datos. *ResNet* muestra la mejor progresión en validación, alcanzando valores cercanos al

85 %, aunque con cierta oscilación en las últimas épocas. *Sivaranjani-CNN* logra mantener una *validation accuracy* bastante estable en torno al 77-78 %. *MobileNet*, a pesar de sobreajustarse en *train*, presenta una validación muy inestable, con fuertes oscilaciones y valores que bajan hasta el 15 % en ciertas épocas.

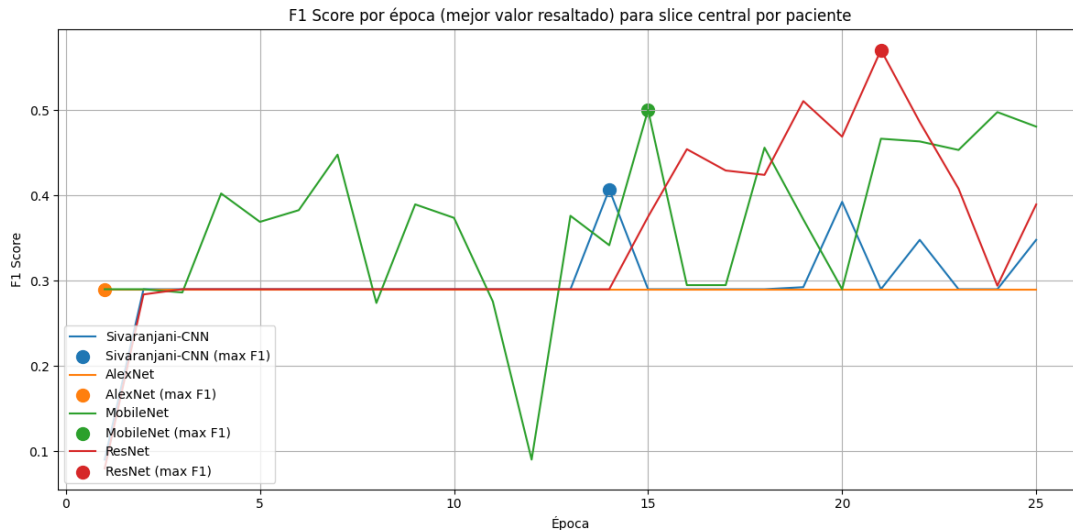


Figura 31: Evolución del *F1-score* en validación para los distintos modelos evaluados. Se destaca el mejor valor alcanzado por cada modelo evaluados con una imagen central por paciente.

En la evolución del *F1 Score* (Figura 31) se observa claramente que *ResNet* es el modelo que consigue los valores más altos, alcanzando un *F1* de aproximadamente 0.57 en su mejor época. *Sivaranjani-CNN* y *MobileNet* presentan valores de *F1* notablemente más bajos y estables. *AlexNet* permanece estancado en un valor bajo constante, en línea con lo observado en las gráficas anteriores.

En el conjunto de test (Figura 32), *ResNet* es el modelo que mejor rendimiento obtiene, con una *test accuracy* de 78 %. *Sivaranjani-CNN* se comporta razonablemente bien con un 80 %, pero con un *F1 Score* bajo (0.40), reflejando una baja capacidad para balancear las clases correctamente. *MobileNet*, aunque había mostrado un sobreajuste fuerte en entrenamiento, no logra generalizar correctamente, con una *test accuracy* de tan solo 64 % y un *F1* de 0.39. Por último, *AlexNet* muestra un rendimiento pobre y constante, con un 77 % de *test accuracy* y un *F1* de 0.29.

La estrategia de utilizar el *slice* central por paciente parece haber ayudado a reducir en parte el sobreajuste que se observaba en las versiones anteriores, aunque aún se aprecia un comportamiento inestable en ciertos modelos, especialmente en *MobileNet*. *ResNet* es claramente el modelo que mejor se adapta a esta nueva configuración, mostrando tanto una progresión sólida en validación como un buen equilibrio en *F1 Score*. No obstante, las oscilaciones observadas en *validation accuracy* y *F1 Score*, incluso con esta reducción de redundancia, indican que la

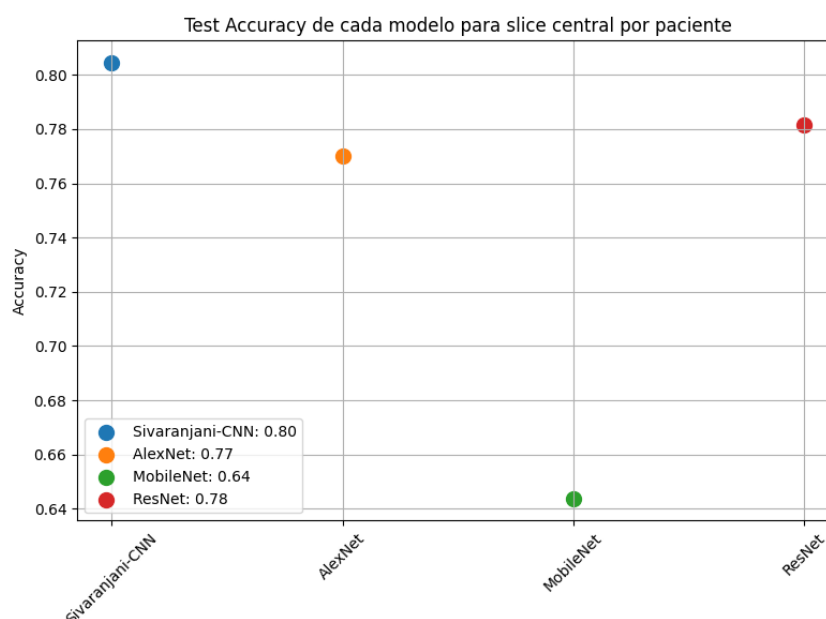


Figura 32: Precisión en el conjunto de test para cada modelo evaluado con una imagen central por paciente.

combinación de bajo tamaño de conjunto de datos y desbalance entre clases sigue afectando al rendimiento y estabilidad de los modelos.

Data Augmentation sobre *slice central* por paciente

Los resultados obtenidos en la configuración anterior, aunque mejoraban la estabilidad en algunos modelos, seguían mostrando síntomas de sobreajuste y curvas de validación poco regulares. Además, el reducido número de imágenes por clase que resultó de tomar únicamente el *slice central* por paciente aumentaba el riesgo de que los modelos memorizaran patrones específicos del conjunto de entrenamiento.

Por ello, se explora la aplicación de técnicas de *data augmentation* para intentar mejorar la capacidad de generalización de los modelos y aumentar la diversidad del conjunto de entrenamiento de manera artificial.

Clase	Distribución original	Distribución tras Data Augmentation
NonDemented (2)	336	336
VeryMildDemented	70	240
MildDemented	28	226

Cuadro 13: Distribución de imágenes por clase antes y después de aplicar Data Augmentation

Dado el bajo rendimiento mostrado por *MobileNet* en esta configuración, se ha decidido prescindir de este modelo en esta fase de experimentación y centrar el análisis en los tres

modelos restantes: *Sivaranjani-CNN*, **AlexNet** y **ResNet**.

El objetivo de este subapartado no es únicamente observar las métricas finales, sino analizar cómo cambian las curvas de aprendizaje de cada modelo tras aplicar *data augmentation*.

Conjunto	Clase	Número de imágenes
Test	NonDemented (2)	67
	VeryMildDemented	14
	MildDemented	6
Entrenamiento	MildDemented	215
	NonDemented (2)	215
	VeryMildDemented	215
Validación	NonDemented (2)	54
	VeryMildDemented	11
	MildDemented	5

Cuadro 14: Distribución de imágenes por clase y conjunto tras *Data Augmentation* y el *split* del dataset.

Además, para ilustrar visualmente el efecto del *data augmentation*, en la Figura 33 se muestran cinco imágenes por clase seleccionadas tras aplicar las transformaciones correspondientes a distintos pacientes.

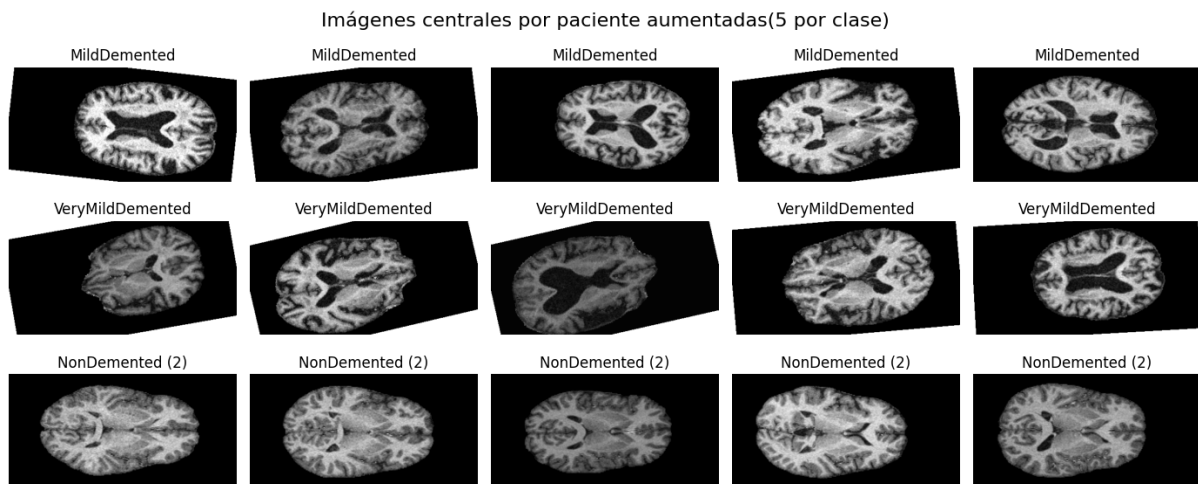


Figura 33: Ejemplos de imágenes tras aplicar *data augmentation*. Se muestran cinco imágenes por clase.

El proceso de *data augmentation* aplicado consistió en generar nuevas imágenes para las clases minoritarias (*MildDemented* y *VeryMildDemented*) con el fin de igualar su tamaño al

de la clase mayoritaria. Para cada imagen aumentada se aplicaron combinaciones aleatorias de las siguientes transformaciones:

- **Volteo horizontal** (*flip left-right*)
- **Ajuste de contraste** con un factor aleatorio entre 0.8 y 1.2
- **Ajuste de brillo** con un factor aleatorio entre 0.8 y 1.2
- **Rotación aleatoria** entre -15 y +15 grados

Estas transformaciones se aplicaron de manera estocástica para cada nueva imagen generada, permitiendo así introducir variaciones sobre las imágenes originales y aumentando de forma efectiva la diversidad del conjunto de entrenamiento..

Con el objetivo de analizar el impacto de las técnicas de *data augmentation* sobre la capacidad de generalización de los modelos, se han comparado las curvas de **exactitud en validación**, **F1-score** y **exactitud en test** para los tres modelos considerados: Sivananjani-CNN, AlexNet y ResNet.

En cada caso, se compara la versión original del modelo (curva en color intenso) frente a su versión entrenada con *data augmentation* (curva en el mismo color pero más claro).

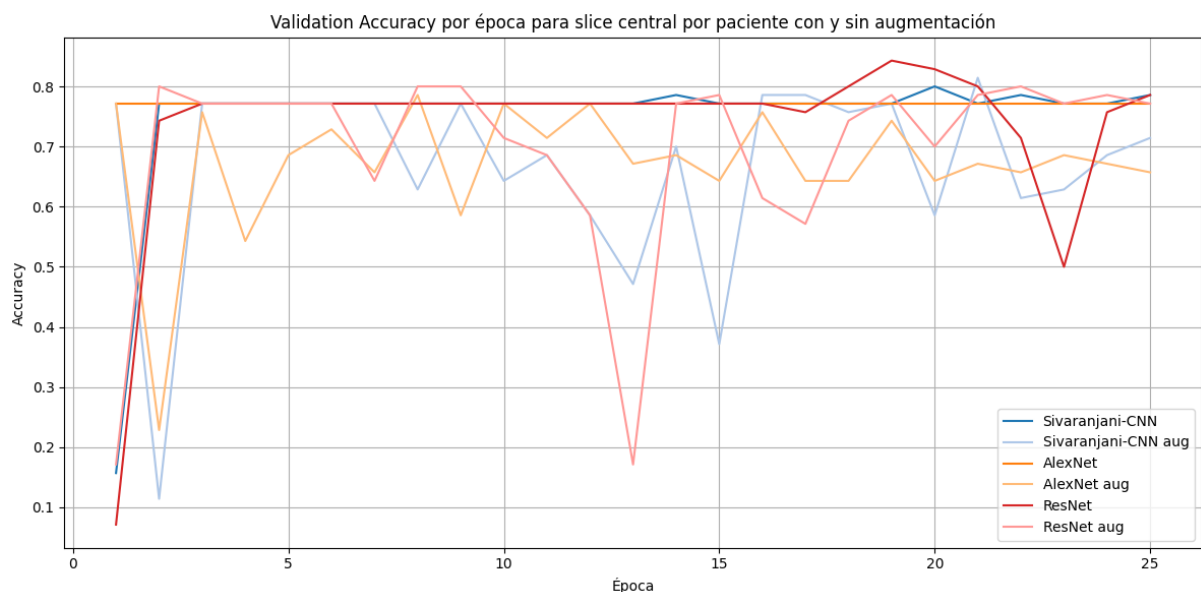


Figura 34: Evolución de la exactitud en validación para los modelos con y sin *data augmentation*.

El análisis conjunto de estas gráficas permite extraer varias conclusiones relevantes:

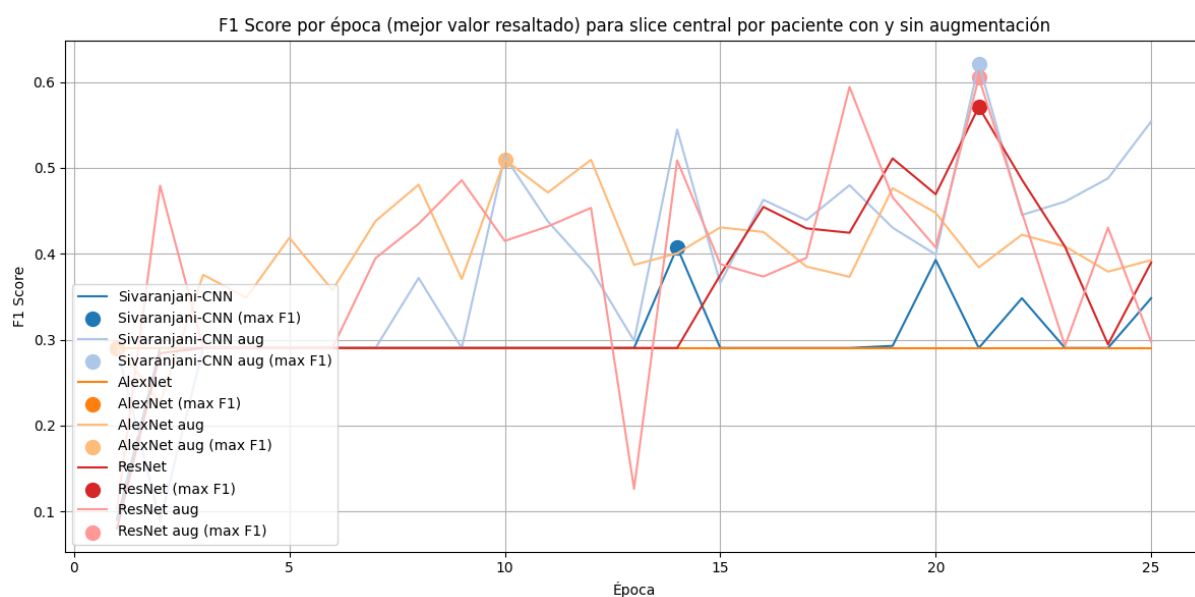


Figura 35: Evolución del F1-score durante la validación para los modelos con y sin *data augmentation*. Se resalta el mejor resultado obtenido en validación.

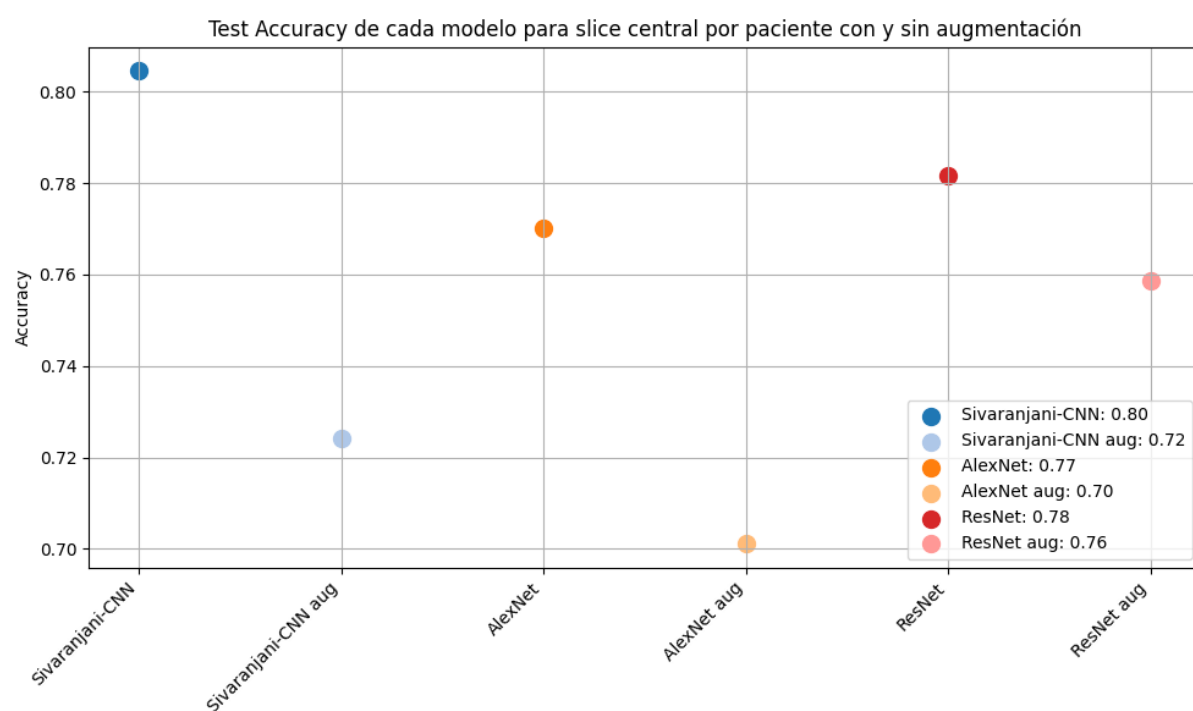


Figura 36: Evolución de la exactitud en entrenamiento para los modelos con y sin *data augmentation*.

- **Sivaranjani-CNN:** en este caso, la aplicación de *data augmentation* no ha resultado demasiado beneficiosa. Mientras que exactitud en test (**0.7241** frente a 0.8046) ha disminuido, el F1-score (**0.4286** frente a 0.4089) han aumentado levemente respecto al modelo sin *augmentation*. Además, las curvas de validación y F1-score muestran un comportamiento claramente inestable, con largos periodos sin mejora y fluctuaciones muy marcadas en las últimas épocas.
- **AlexNet:** el aumento de datos ha sido especialmente beneficioso en este caso. La versión original presentaba métricas planas y sobreajustadas, mientras que la versión con *augmentation* muestra curvas de validación y F1-score más dinámicas y con mejor rendimiento final (**F1-score de 0.4081** frente a 0.2900). No obstante, también en este caso las curvas resultantes presentan una mayor inestabilidad y variaciones pronunciadas entre épocas.
- **ResNet:** aunque el modelo original ya ofrecía un buen rendimiento sin necesidad de *augmentation* (**F1-score de 0.5399**), la versión aumentada no consigue superar este resultado y presenta una mayor variabilidad en las curvas. En algunas épocas se observan picos de F1 y test acc muy elevados, pero también descensos súbitos, lo que sugiere que el *augmentation* podría estar introduciendo ruido adicional en un modelo que ya era suficientemente robusto.

En resumen, la aplicación de técnicas de *data augmentation* en esta configuración ha producido resultados mixtos. En el caso de *AlexNet* ha permitido mejorar claramente el F1-score y dinamizar el aprendizaje. Por otro lado, en *ResNet* el *augmentation* no aporta mejoras claras y en algunos casos degrada la estabilidad del entrenamiento. En general, el efecto colateral común en todos los casos es una **mayor inestabilidad en las curvas de aprendizaje**, con fluctuaciones marcadas entre épocas que podrían dificultar la convergencia óptima de los modelos, y en todos los casos la precisión en test se ha visto desfavorecida.

4.5. Versión 4 – Múltiples *slices* por paciente

Para esta cuarta versión del experimento, se decidió continuar con la estrategia de trabajar con el *slice* central de cada paciente, ya que en versiones anteriores esta selección había contribuido a mejorar ligeramente los resultados.

Sin embargo, con el objetivo de aumentar ligeramente el número de imágenes disponibles y proporcionar algo más de variabilidad al conjunto de datos, se optó en esta ocasión por incluir, además del *slice* central, dos imágenes adicionales por paciente:

- Una imagen situada 10 posiciones antes del *slice* central.

- Una imagen situada 10 posiciones después del *slice* central.

Esta elección se realizó con la intención de añadir nuevas imágenes que no fueran excesivamente redundantes respecto a la central (por ello se eligió una separación de 10 posiciones), evitando así un aumento artificial del sobreajuste.

Una vez generados los conjuntos de datos con esta nueva estrategia, se procedió a la división en los conjuntos de entrenamiento, validación y prueba, quedando la distribución de imágenes por clase y conjunto como sigue:

Conjunto	MildDemented	NonDemented	VeryMildDemented
Train	51	645	135
Validation	15	162	33
Test	18	201	42

Cuadro 15: Número de imágenes por clase y conjunto en la Versión 4.

A continuación, se muestran las cinco primeras imágenes de cada clase del conjunto de entrenamiento (Figura 37), permitiendo así verificar visualmente la diversidad de las muestras tras la inclusión de los *slices* adicionales.

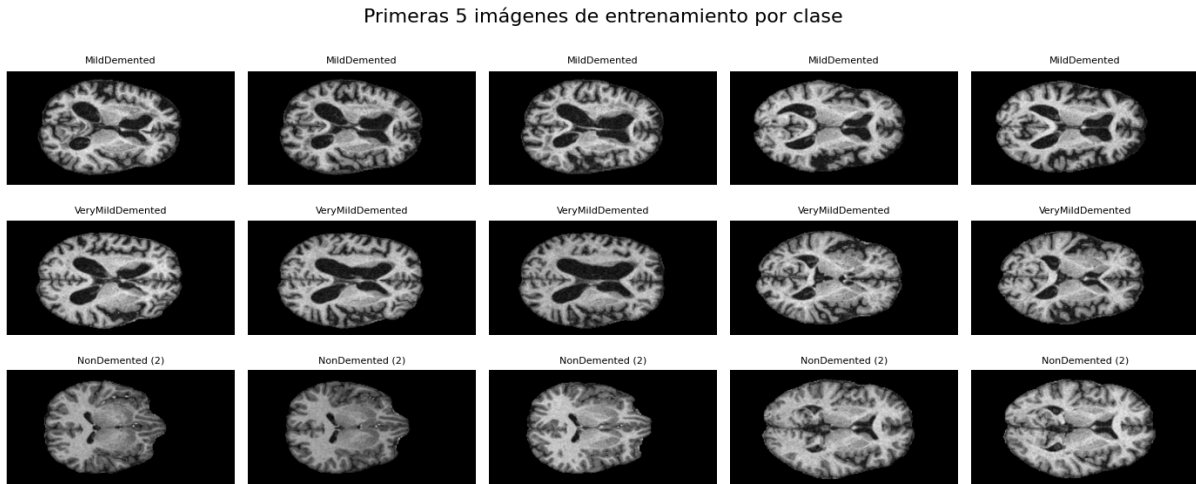


Figura 37: Primeras 5 imágenes por cada clase del conjunto de entrenamiento.

Para complementar el análisis visual, se muestra también el histograma de intensidades en escala de grises (0–255) correspondiente a la primera imagen del conjunto de entrenamiento (Figura 38). En este caso, se ha eliminado previamente el fondo negro de la imagen para facilitar una visualización más clara de la distribución real de intensidades en las regiones relevantes. Este histograma permite así analizar con mayor precisión cómo se reparten los niveles de gris en las zonas anatómicas de interés, sin que los valores bajos (cerca de 0) dominen la gráfica debido al fondo.

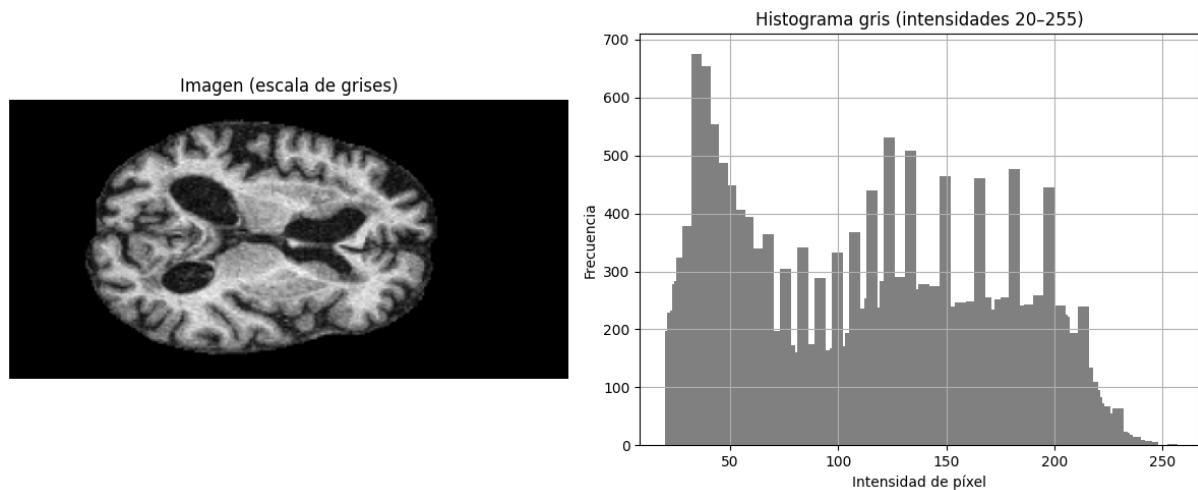


Figura 38: Histograma (escala 0–255) de la primera imagen del conjunto, tras eliminar el fondo negro.

Posteriormente, se procedió a aplicar un preprocesamiento de normalización únicamente sobre el conjunto de entrenamiento. Esta normalización consistió en centrar los datos en torno a la **media 0** y reescalar con una **desviación estándar de 0.5** en cada canal (rojo, verde y azul). Este tipo de normalización permite facilitar el aprendizaje del modelo, al homogenizar las escalas de los datos de entrada.

Tras aplicar esta transformación, se vuelven a mostrar las cinco primeras imágenes de cada clase del conjunto de entrenamiento (Figura 39), lo que permite observar cómo ha cambiado el aspecto visual de las imágenes tras la normalización.

Se muestra de nuevo el histograma de la misma primera imagen que se había analizado previamente, ahora tras la aplicación de la normalización (Figura 40). En este caso, el histograma se calcula sobre el canal rojo, aunque se verificó que el comportamiento es idéntico en los tres canales:

- **Rojo:** min = 0.0000, max = 1.9451
- **Verde:** min = 0.0000, max = 1.9451
- **Azul:** min = 0.0000, max = 1.9451

El histograma resultante, mostrado en la Figura 40, representa la distribución de intensidades de la misma imagen tras aplicar una transformación de normalización con media 0 y desviación estándar 0.5. Debido a esta configuración, los valores de píxel se reescalan a un rango aproximado entre -2 y 2, aunque en la práctica predominan los valores entre 0 y 2 al haberse eliminado previamente el fondo negro de la imagen.

5 imágenes por clase del dataset (transformadas)

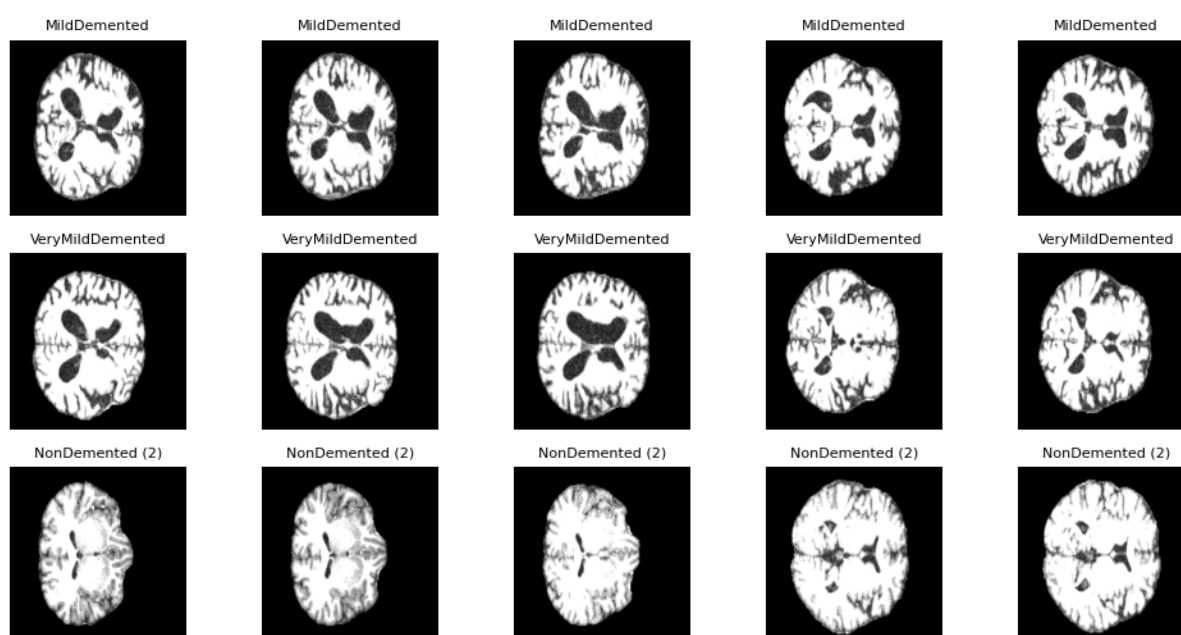


Figura 39: Primeras 5 imágenes por cada clase del conjunto de entrenamiento tras aplicar la transformación ($\text{mean}=0$, $\text{std}=0.5$).

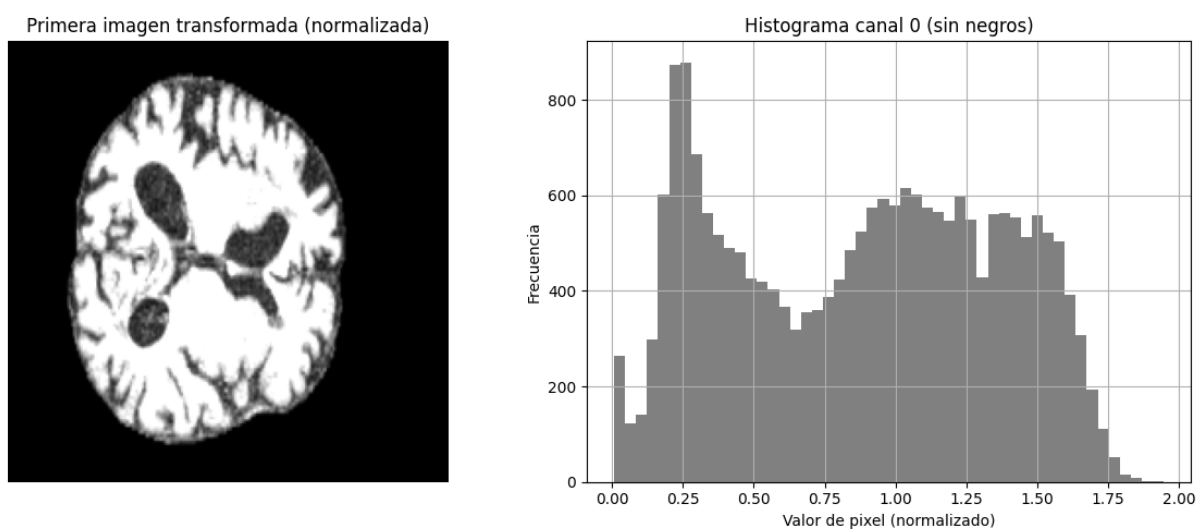


Figura 40: Histograma (escala 0–2) de la primera imagen del conjunto, tras aplicar la transformación ($\text{mean} = 0$, $\text{std} = 0.5$) y eliminar el fondo negro.

Tras entrenar los tres modelos (Sivaranjani-CNN, AlexNet y ResNet) en esta configuración, se presentan los resultados obtenidos, considerando las métricas de **accuracy** y **F1-score** en test, así como el comportamiento de las curvas de validación y F1 a lo largo de las épocas.

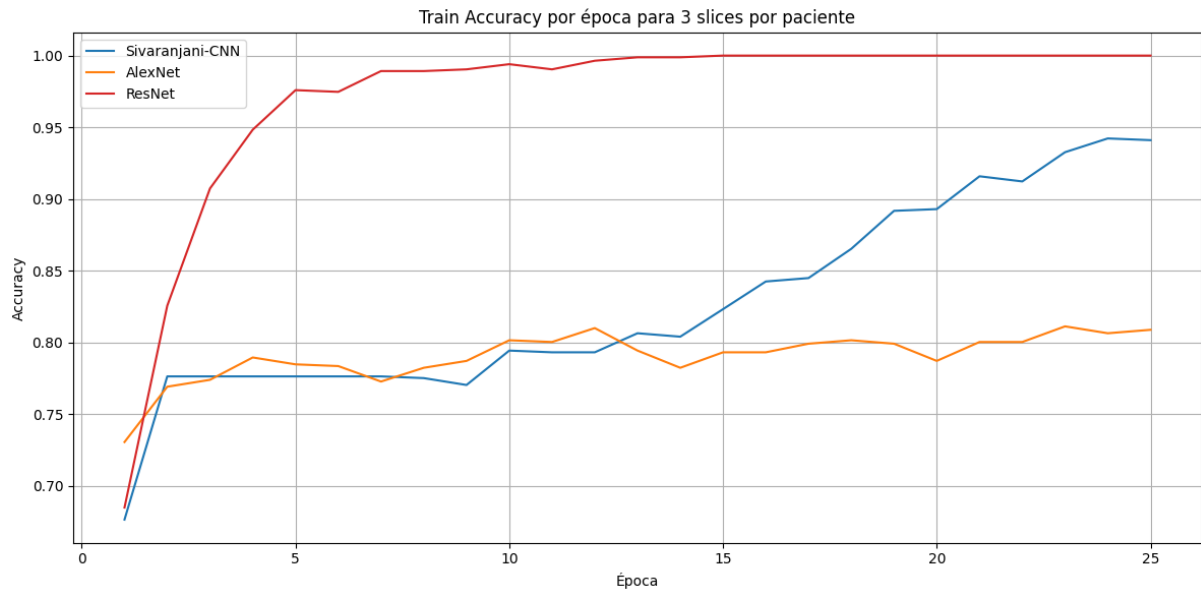


Figura 41: Evolución de la *training accuracy* para los tres modelos en la versión 4.

- **Sivaranjani-CNN**: este modelo mantiene un rendimiento sólido en test, con una **accuracy de 0.7931** y un **F1-score de 0.5057**, resultados prácticamente idénticos a los obtenidos en la versión anterior. Las curvas de validación y F1 muestran un comportamiento relativamente estable, sin grandes fluctuaciones entre épocas.
- **AlexNet**: en esta versión, AlexNet ha mostrado una **ligera mejora respecto a la versión anterior**, alcanzando una **accuracy de 0.7510** y un **F1-score de 0.3621** en test. Además, las curvas de validación y F1 han presentado una evolución más suave y consistente que en versiones previas, sin las grandes oscilaciones observadas anteriormente.
- **ResNet**: el modelo ResNet mantiene un rendimiento competitivo, con una **accuracy de 0.7548** y un **F1-score de 0.4866** en test. Las curvas de validación sin embargo se han vuelto un poco más inestables que en la versión anterior.

En resumen, los resultados de esta cuarta versión muestran que la estrategia de incluir *slices* adicionales y aplicar una normalización ha tenido un efecto positivo en la estabilidad del aprendizaje general, especialmente para los modelos AlexNet y Sivaranjani-CNN, que en versiones anteriores sufrían mayores fluctuaciones.

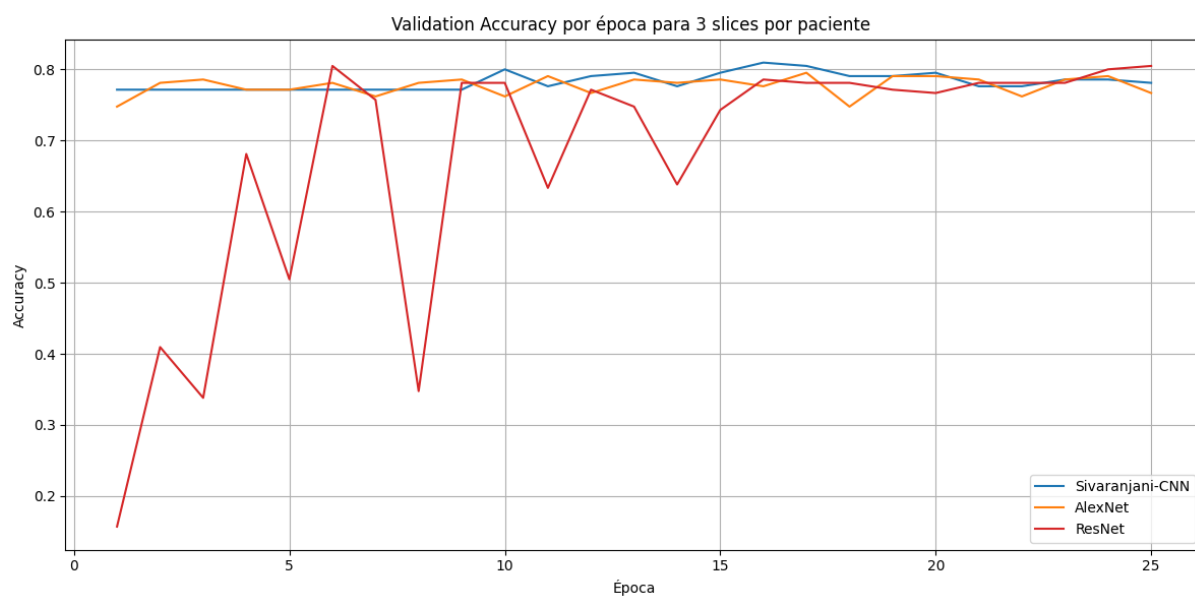


Figura 42: Evolución de la *validation accuracy* para los tres modelos en la versión 4.

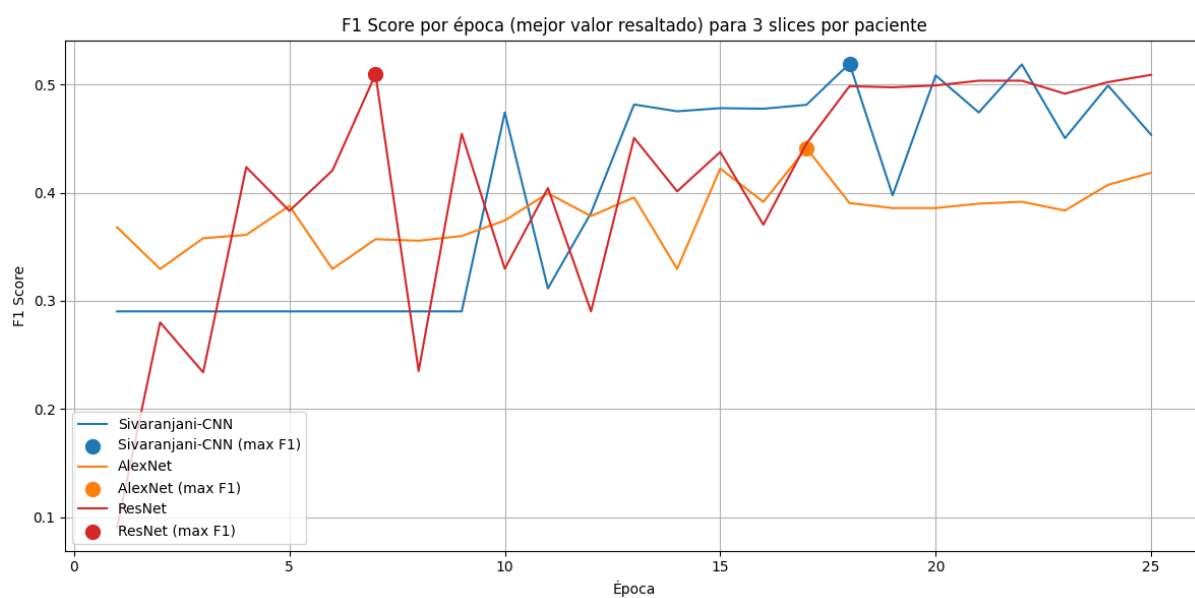


Figura 43: Evolución del *F1-score* en validación para los tres modelos en la versión 4.

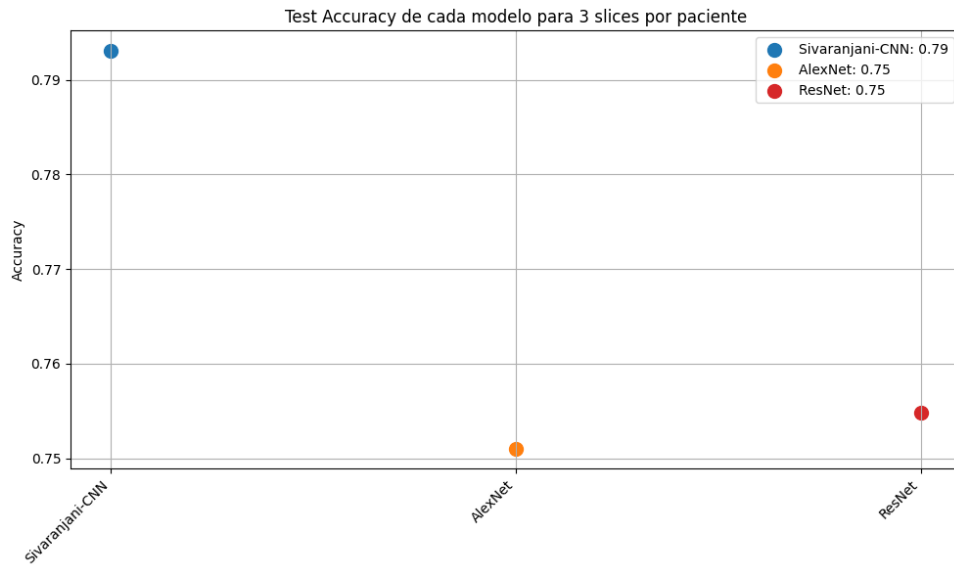


Figura 44: Comparativa de *test accuracy* para los tres modelos en la versión 4.

Prueba de *Data Augmentation*

Como prueba adicional, se ha explorado la posibilidad de aplicar técnicas de *data augmentation* para intentar mejorar el rendimiento de los modelos. En concreto, se ha aplicado *data augmentation* únicamente sobre los dos modelos que mejores resultados habían obtenido: el **Sivaranjani-CNN** y la **ResNet**.

Las Figuras 45 y 46 muestran la evolución de la *validation accuracy* y del *F1-score* en validación para los modelos con y sin *data augmentation*.

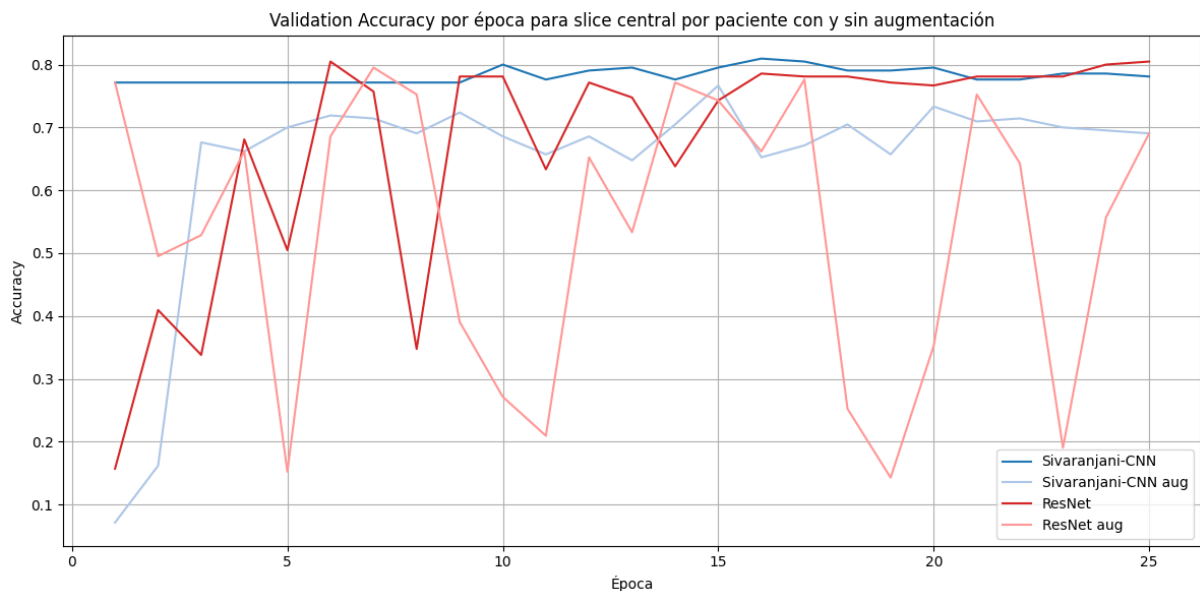


Figura 45: Evolución de la *validation accuracy* con y sin *data augmentation*.

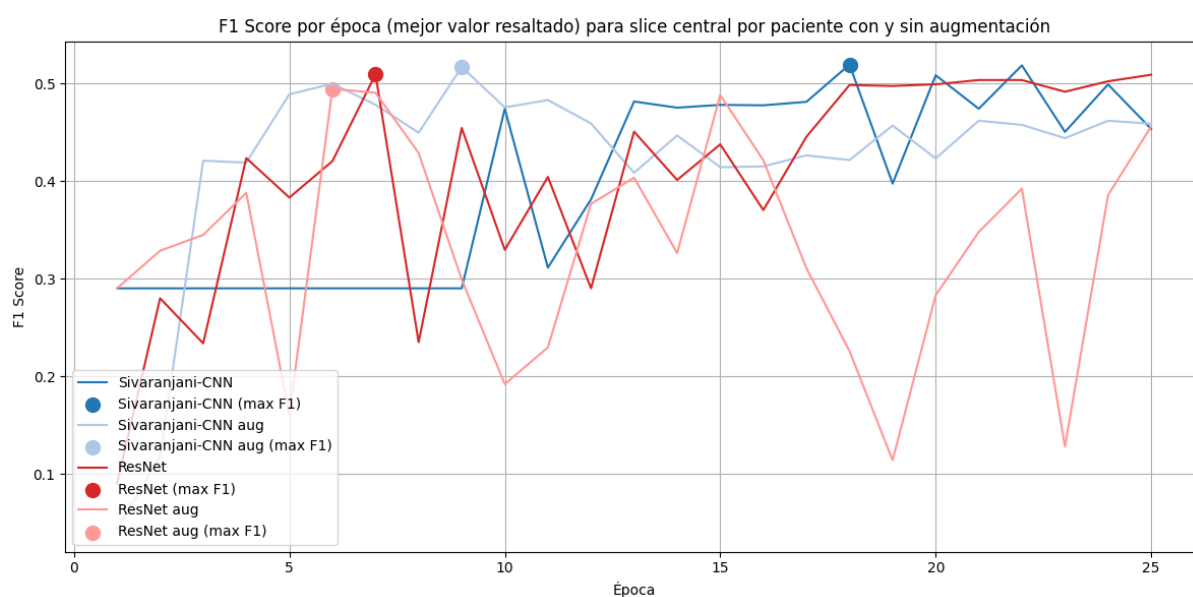


Figura 46: Evolución del $F1$ -score en validación con y sin *data augmentation*.

Sin embargo, como puede observarse en dichas gráficas, el efecto del *data augmentation* en este caso no ha sido positivo. En ambos modelos, **las curvas de validación muestran un comportamiento considerablemente más inestable**, con oscilaciones pronunciadas a lo largo de las épocas, tanto en *validation accuracy* como en $F1$ -score.

Además, los resultados finales en el conjunto de *test* confirman esta tendencia:

- Para **Sivaranjani-CNN**, el *test accuracy* disminuye de **0.7931** a **0.7471**, y el $F1$ -score se mantiene prácticamente igual de **0.5057** a **0.5096**.
- Para la **ResNet**, aunque el *test accuracy* disminuye de **0.7548** a **0.7356**, el $F1$ -score sube de **0.4866** a **0.5745** pero se alcanza en una época muy temprana, y la curva de validación muestra menor estabilidad.

Por tanto, el análisis de las curvas de validación y la falta de mejora consistente en el $F1$ -score nos llevan a concluir que, en esta versión del problema, la aplicación de *data augmentation* no resulta beneficiosa. Es preferible utilizar los modelos sin dicha técnica.

4.6. Versión 5 – Ajustes de normalización y optimización

En esta nueva versión, se introduce un cambio en la estrategia de normalización aplicada a las imágenes. En la versión anterior se había utilizado una normalización con media 0 y desviación típica 0.5, lo cual producía un reescalado más agresivo de los valores de píxel.

En esta versión se opta por una normalización más estándar, con **media 0 y desviación típica 1**. Esta elección es común en el entrenamiento de redes convolucionales, ya que:

- Centra los datos en torno a cero, mejorando la estabilidad numérica durante el entrenamiento.
- Asegura una escala uniforme entre las diferentes características de entrada.
- Facilita la convergencia de los algoritmos de optimización y la correcta inicialización de los pesos.

Tras aplicar esta normalización, se visualizan nuevamente las imágenes del conjunto de entrenamiento para comprobar el efecto visual del preprocesado. En la Figura 47 se muestra un conjunto de las **5 primeras imágenes de cada clase** tras la normalización.

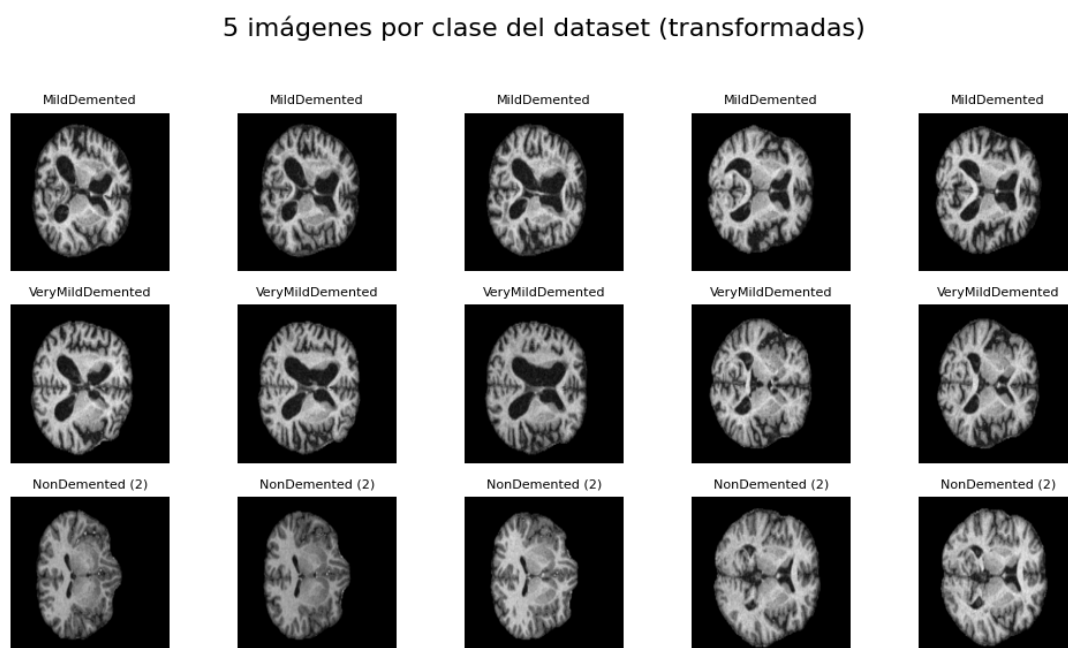


Figura 47: Primeras 5 imágenes de cada clase tras la normalización con media 0 y desviación típica 1.

Además, se representa el histograma de la primera imagen del conjunto de entrenamiento (Figura 48). En este caso, se ha eliminado el fondo negro de la imagen para visualizar con mayor claridad la distribución de intensidades relevantes. Tras aplicar la normalización (con media 0 y desviación estándar 1), los valores de píxel quedan distribuidos en torno a 0.

En esta versión se han evaluado nuevamente los dos modelos más competitivos de las versiones anteriores: *Sivaranjani-CNN* y una **ResNet** sin preentrenar.

En este caso, se ha **descartado el uso de pesos preentrenados en la ResNet**, ya que el preentrenamiento en datasets de imágenes naturales (como ImageNet) no aporta ventajas en este tipo de imágenes biomédicas, que presentan características morfológicas muy distintas.

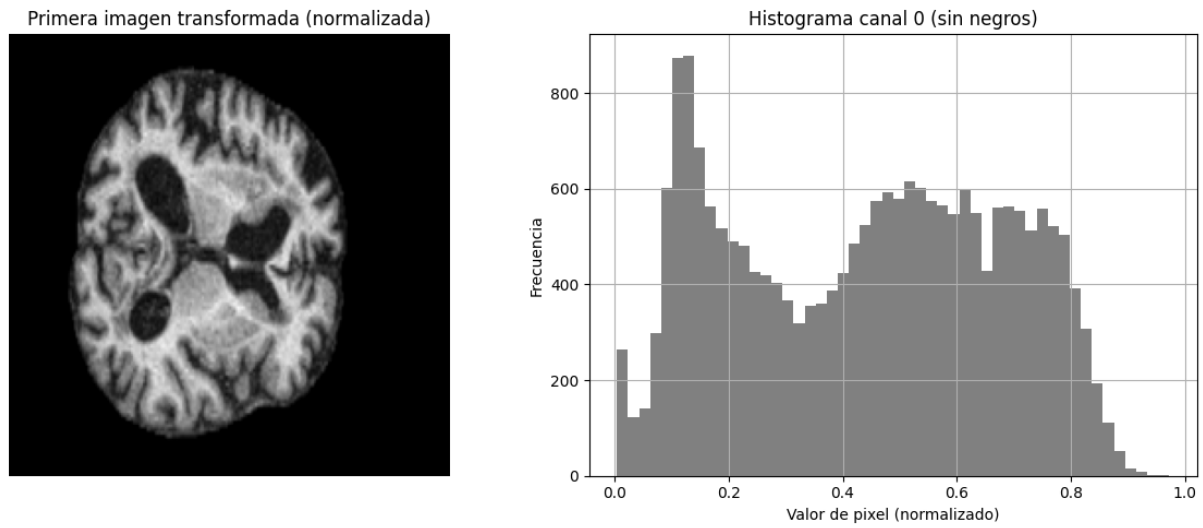


Figura 48: Histograma de la primera imagen del conjunto de entrenamiento tras la normalización.

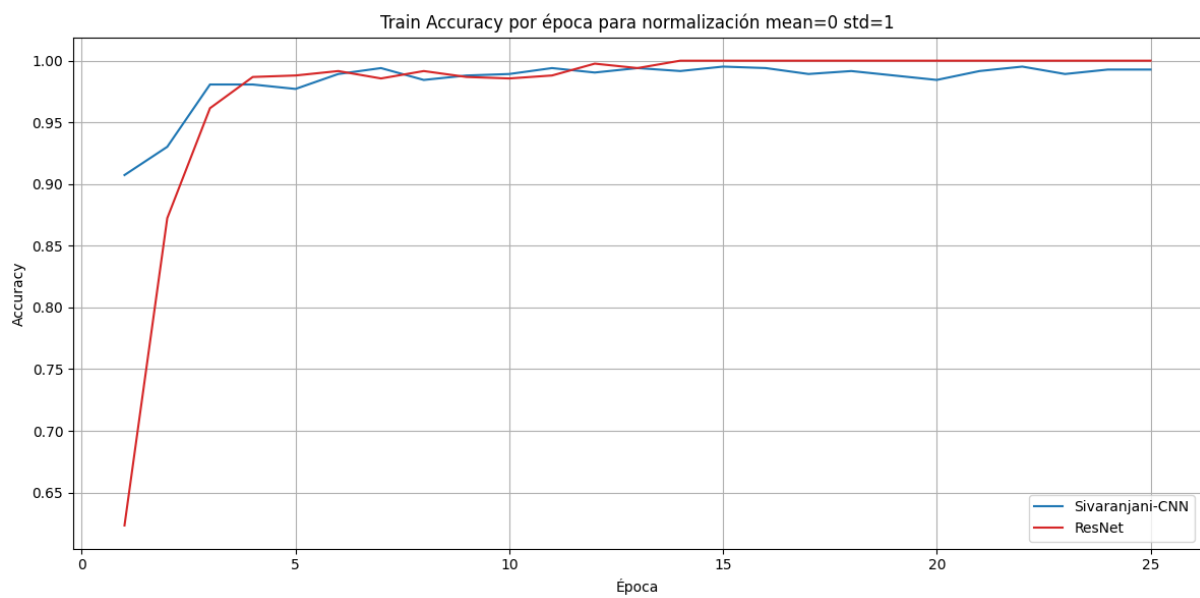


Figura 49: Evolución del *Train Accuracy* por época para normalización $mean=0$, $std=1$.

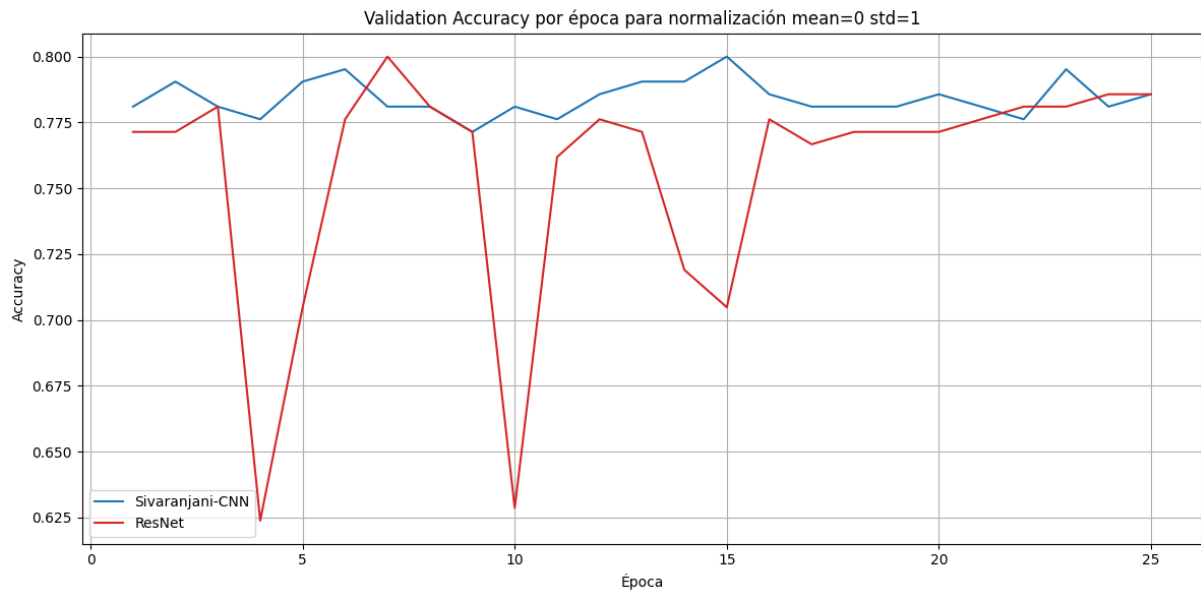


Figura 50: Evolución del *Validation Accuracy* por época para normalización $mean=0$, $std=1$.

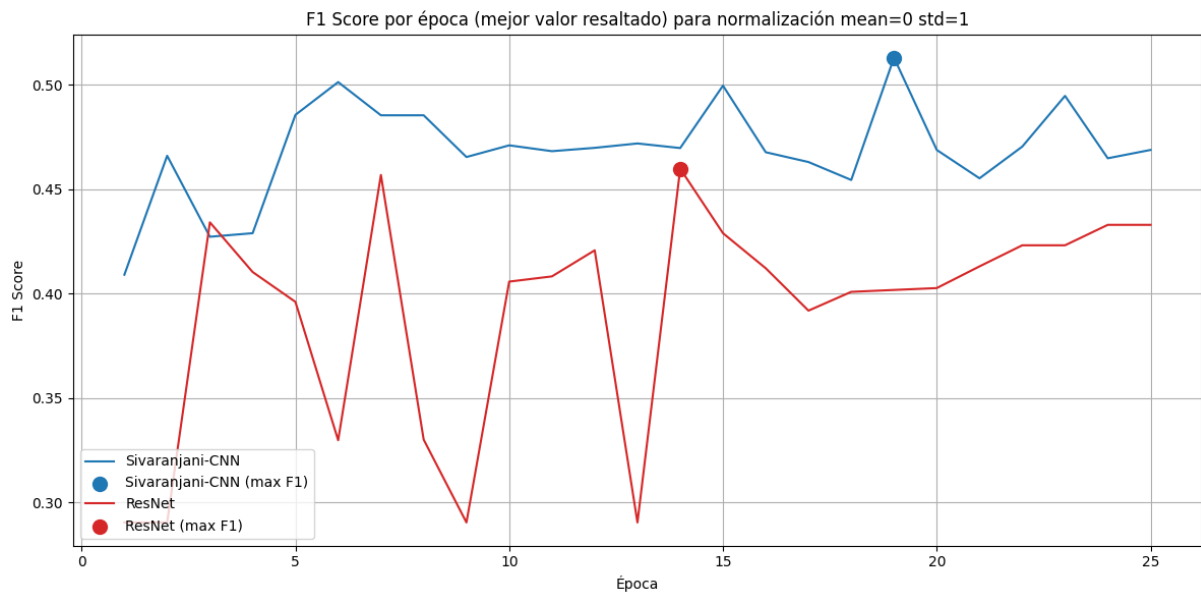


Figura 51: Evolución del *F1 Score* por época (con el mejor valor destacado) para normalización $mean=0$, $std=1$.

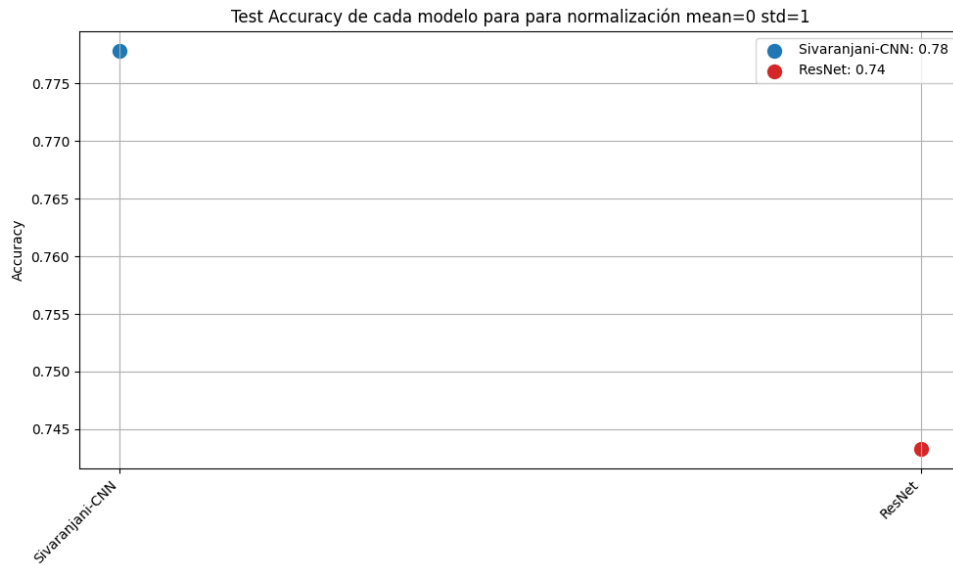


Figura 52: *Test Accuracy* de cada modelo para normalización $mean=0$, $std=1$.

Con la nueva normalización aplicada (media 0 y desviación típica 1), los modelos han mostrado los siguientes comportamientos:

- **Sivaranjani-CNN** mantiene un aprendizaje estable, con una evolución progresiva del *accuracy* en validación y del *F1-score*. Se alcanza un **F1-score en test de 0.4902** y un **accuracy en test de 0.7739**.
- La **ResNet sin preentrenar** muestra picos en algunas de las épocas, y no logra superar en F1 al modelo del artículo, alcanzando un **F1-score en test de 0.4067**, y un **accuracy en test de 0.7471**.

En conclusión, el cambio de normalización no perjudica el rendimiento, y en el caso de la ResNet parece haber contribuido a una leve mejoría en la capacidad de generalización (especialmente en términos de *F1-score*). Las gráficas confirman que ambos modelos logran estabilizar su *accuracy* y *F1-score* en las últimas épocas, mostrando un comportamiento robusto.

Prueba de optimizadores

Para evaluar el impacto del optimizador sobre el rendimiento de los modelos, se compararon tres alternativas: **Adam**, **AdamW** y **SGD**, aplicándolos tanto a *Sivaranjani-CNN* como a *ResNet*. Los resultados obtenidos muestran que el optimizador **Adam** proporciona el mejor desempeño general. En particular, logra un equilibrio sólido entre *accuracy* y *F1-score* en ambos modelos. Por ejemplo, en *Sivaranjani-CNN*, Adam alcanza un *F1-score* en test de **0.4902**, superando a AdamW (**0.4026**) y SGD (**0.2900**). En *ResNet*, Adam mantiene también un rendimiento competitivo, con un *F1-score* de **0.4067**, solo ligeramente inferior al obtenido con

AdamW (**0.4956**), pero mostrando una mayor estabilidad y menor variabilidad en la validación. En contraste, el uso de SGD dio lugar a un F1-score prácticamente constante e inferior (**0.2900** en ambos modelos), lo que evidencia su limitada capacidad de adaptación al desbalance de clases. Por ello, se concluye que **Adam es la opción más adecuada para esta tarea**, al ofrecer un rendimiento robusto, especialmente en contextos con distribución desigual de clases.

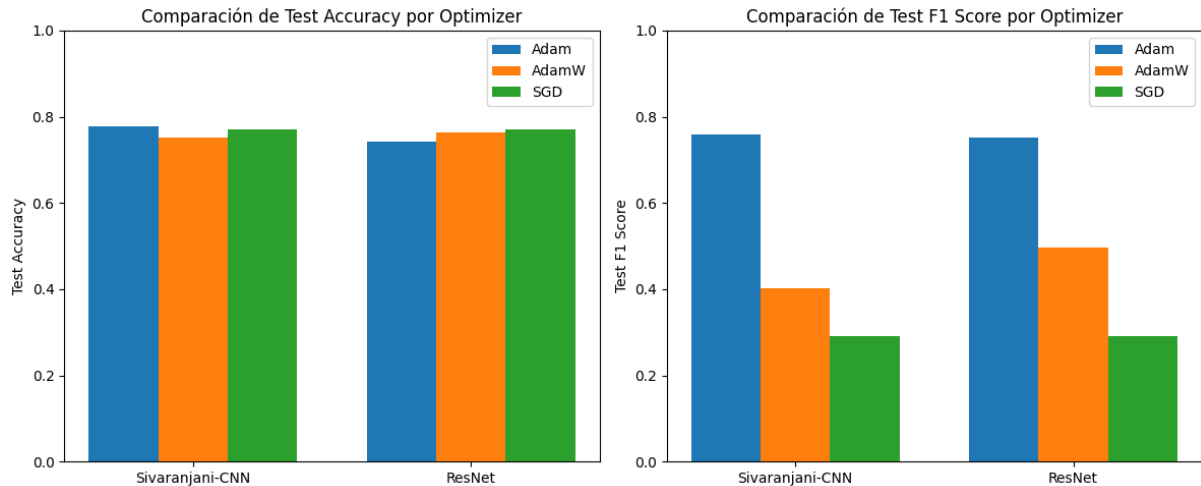


Figura 53: Comparación de optimizadores (**Adam**, **AdamW** y **SGD**) en el *Sivaranjani-CNN* y en *ResNet*. Se muestra el *Test Accuracy* y el *F1 Score de Test* para cada combinación.

5

Discusión

5.1. Discusión

A continuación, se resumen los valores obtenidos de *Test Accuracy* y *Test F1-score* para cada versión del pipeline evaluado, con sus respectivas variantes de preprocesado y arquitectura. Se incluyen tanto los modelos base como los ajustes experimentados (normalización, número de cortes, data augmentation y optimizador), destacando el impacto que cada combinación ha tenido en el rendimiento de las arquitecturas comparadas.

Cuadro 16: Comparativa de **Test Accuracy** entre modelos y versiones

Versión	Configuración	Sivaranjani-CNN	ResNet	AlexNet	MobileNet
Versión 2	1000 imágenes por clase	0.6168	0.5404	0.2243	0.3548
	5000 imágenes por clase	0.6837	0.3819	0.4955	0.4311
Versión 3	Slice central por paciente	0.8046	0.7816	0.7701	0.3977
	+ Data Augmentation	0.7241	0.7586	0.7011	-
Versión 4	3 slices (mean=0, std=0.5)	0.7931	0.7548	0.7510	-
	+ Data Augmentation	0.7471	0.7356	-	-
Versión 5	Adam (mean=0, std=1)	0.7778	0.7433	-	-
	AdamW (mean=0, std=1)	0.7510	0.7625	-	-
	SBG (mean=0, std=1)	0.7701	0.7701	-	-

Los resultados obtenidos en las diferentes versiones muestran con claridad cómo la elección de configuraciones adecuadas puede marcar una diferencia significativa en el rendimiento del sistema. En particular, la combinación de **normalización estándar** (media 0, desviación típica 1) junto con el uso del optimizador **Adam** —implementada en la **Versión 5**— proporciona el **mejor rendimiento global**. Tanto la arquitectura *Sivaranjani-CNN* como *ResNet* alcanzan sus mejores resultados bajo esta configuración, con valores altos de **accuracy** (0,7778 y 0,7433 respectivamente) y, especialmente, con **F1-scores elevados** (0,7583 y 0,7525), lo cual es fundamental en contextos con **clases desbalanceadas**.

Además, esta configuración demostró una mayor **estabilidad durante el entrenamiento y la validación** sobre todo en *Sivaranjani-CNN*, mostrando curvas consistentes y mejor

Cuadro 17: Comparativa de **Test F1-score** entre modelos y versiones

Versión	Configuración	Sivaranjani-CNN	ResNet	AlexNet	MobileNet
Versión 2	1000 imágenes por clase	0.6142	0.5423	0.2291	0.3617
	5000 imágenes por clase	0.6799	0.3891	0.5003	0.4229
Versión 3	Slice central por paciente	0.4089	0.5399	0.2900	0.6437
	+ Data Augmentation	0.4081	0.5463	0.5399	-
Versión 4	3 slices (mean=0, std=0.5)	0.5057	0.4866	0.3621	-
	+ Data Augmentation	0.5096	0.5745	-	-
Versión 5	Adam (mean=0, std=1)	0.7583	0.7525	-	-
	AdamW (mean=0, std=1)	0.4026	0.4956	-	-
	SBG (mean=0, std=1)	0.2900	0.2900	-	-

capacidad de **generalización en el conjunto de test**. En contraste, configuraciones anteriores como la **Versión 3**, que empleaban una única *slice* central por paciente y técnicas de *data augmentation*, aunque lograban buenos valores de **accuracy**, mostraban **menor F1-score y mayor sensibilidad al preprocesamiento aplicado** además de curvas de validación mas inestables. Esto indica que no todas las estrategias que amplían los datos conllevan una mejora real del poder de generalización del modelo, especialmente cuando no se ajustan al contexto clínico del problema.

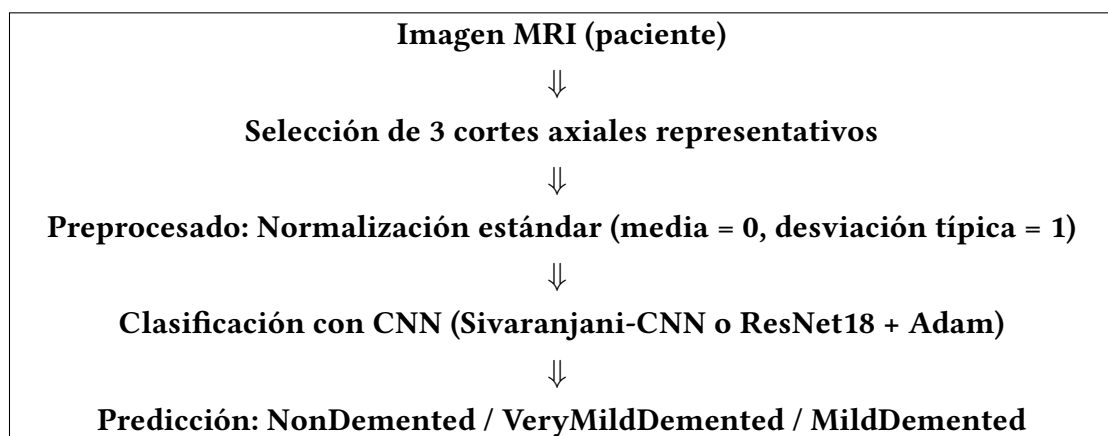


Figura 54: Esquema del flujo de inferencia con los mejores modelos.

El análisis comparativo entre arquitecturas reveló también que redes como *AlexNet* y *MobileNet*, más ligeras y generalistas, muestran limitaciones claras en este tipo de tareas. Su menor profundidad y arquitectura no especializada parecen no adaptarse bien a los patrones sutiles presentes en cortes cerebrales, frente a arquitecturas como *ResNet* o la red propuesta por Sivaranjani et al., mejor adaptadas a esta tarea específica.

También se ha evidenciado que la cantidad de datos es un factor importante, aunque no

absoluto. Las primeras versiones mostraron mejoras notables al pasar de 1000 a 5000 imágenes por clase, pero en fases posteriores se logró rendimiento superior con menos datos pero mejor organizados y preprocesados. Esto resalta que la calidad del procesamiento y la coherencia en el diseño del pipeline pueden compensar ciertas limitaciones del conjunto de datos.

En conjunto, estos resultados muestran que no existe una única técnica que garantice el éxito, sino que es el ajuste fino entre el preprocesamiento, la arquitectura del modelo y la estrategia de entrenamiento lo que determina la calidad final del sistema. Además, este proyecto ha puesto de manifiesto la necesidad de realizar una evaluación crítica de los modelos disponibles en plataformas abiertas: algunos modelos inicialmente prometedores, como el obtenido desde Kaggle, resultaron estar mal validados (por ejemplo, sin estratificación por paciente), lo cual inflaba artificialmente su precisión. Esto resalta la importancia de revisar cuidadosamente la metodología de cualquier modelo antes de considerarlo como referencia o base de comparación.

6

Aplicación Web

Para mostrar el funcionamiento del sistema desarrollado, se ha implementado una interfaz de usuario utilizando la librería *Streamlit*, que permite realizar predicciones sobre imágenes de resonancia magnética cerebral (MRI) directamente desde el navegador.

6.1. Funcionamiento de la aplicación demo

La aplicación demo desarrollada permite realizar una predicción automática del estado clínico del Alzheimer a partir de una imagen de resonancia magnética (MRI). El usuario puede seleccionar entre dos modelos entrenados (*Sivaranjani-CNN* o *ResNet18*) y cargar una imagen médica desde su equipo. A partir de ahí, la aplicación realiza el preprocesamiento necesario, ejecuta la inferencia con el modelo seleccionado y muestra las probabilidades asociadas a cada clase.

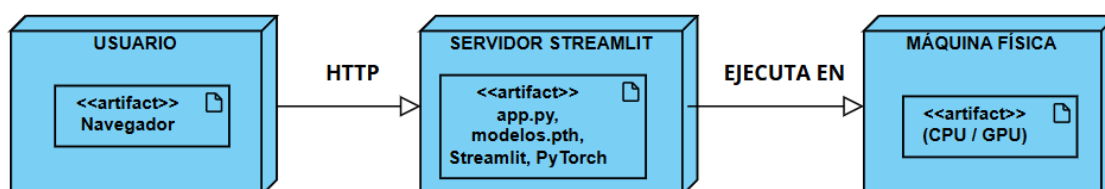


Figura 55: Diagrama de despliegue de la arquitectura de la aplicación demo.

La Figura 55 muestra el diagrama de despliegue, donde se ilustran los componentes físicos y lógicos que intervienen en la ejecución de la demo. En él se representan los nodos implicados (usuario, servidor y entorno de ejecución) y cómo se distribuyen los diferentes elementos del sistema, como el script principal de la aplicación y los modelos entrenados.

La Figura 56 muestra un esquema del flujo de la aplicación.

Para facilitar la comprensión del funcionamiento de la demo, a continuación se muestran capturas de la interfaz web que ilustran el proceso paso a paso.

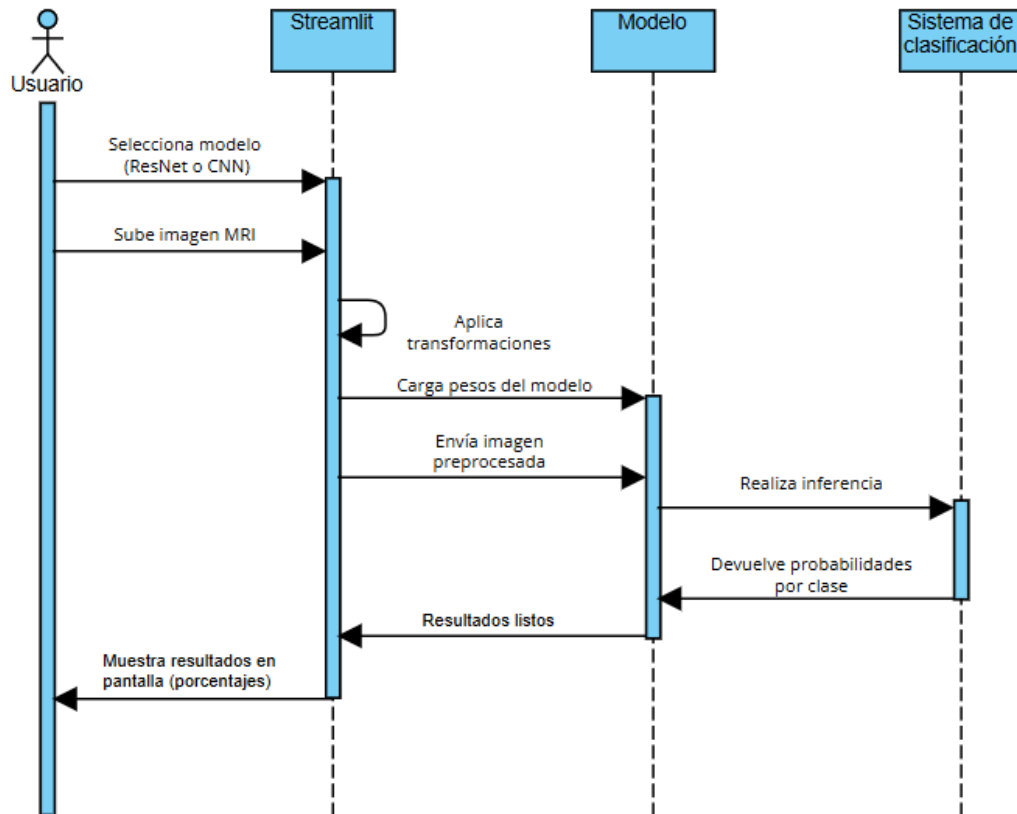


Figura 56: Diagrama de secuencia de la aplicación demo desarrollada.

En primer lugar, como se muestra en la Figura 60, el usuario puede seleccionar el modelo que desea utilizar entre las dos opciones disponibles: *Sivaranjani-CNN* y *ResNet18*.

A continuación, el usuario carga una imagen de resonancia magnética cerebral (MRI), como se muestra en la Figura 61. Esta imagen será preprocesada automáticamente y utilizada como entrada del modelo.

Una vez realizada la inferencia, la aplicación muestra las probabilidades asociadas a cada una de las clases, tal y como se puede ver en la Figura 62. Esta salida permite interpretar el grado estimado de deterioro cognitivo para la imagen suministrada.

Clasificador de Alzheimer por Imágenes

Sube una imagen y elige el modelo para predecir el grado de demencia

Selecciona el modelo que quieres usar:

☒ Sivaranjani-CNN

☐ ResNet18

Elige una imagen...

 Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

Browse files

Figura 57: Selección del modelo que se quiere usar en la demo.

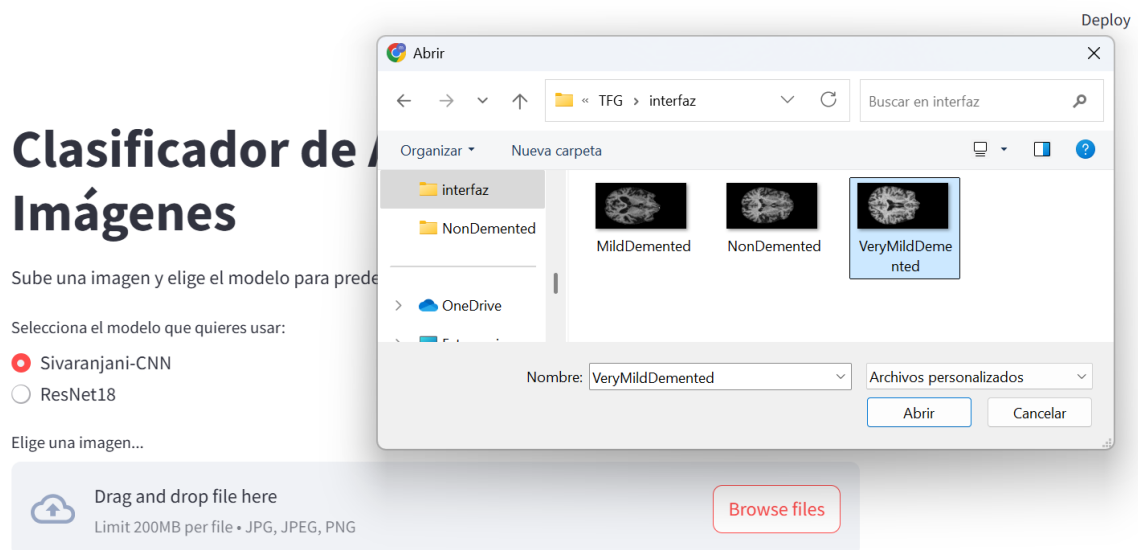


Figura 58: Carga de una imagen MRI desde el navegador.

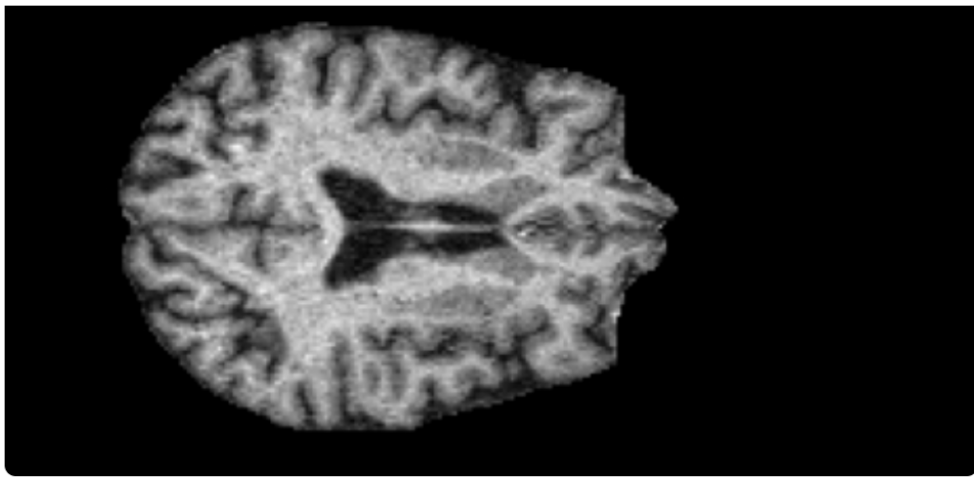


Imagen cargada

Resultados de la predicción

NonDemented: 11.87%

VeryMildDemented: 87.07%

MildDemented: 1.07%

Figura 59: Probabilidades obtenidas tras la predicción.

Conclusiones y Líneas Futuras

7.1. Conclusiones

A lo largo de este trabajo se ha desarrollado un sistema basado en **deep learning** para la **clasificación automática de imágenes de resonancia magnética (MRI)** de pacientes con Alzheimer, diferenciando entre distintos estadios clínicos de la enfermedad. Para ello, se ha llevado a cabo un proceso completo que ha abarcado desde la **recopilación y preprocesamiento de datos** del conjunto público **ADNI**, hasta la **implementación, entrenamiento y evaluación** de diversos modelos de aprendizaje profundo.

Este trabajo ha abordado con éxito los objetivos planteados inicialmente, que incluían la preparación estructurada del conjunto de datos centrado en cortes axiales representativos, así como la experimentación con distintas arquitecturas (*ResNet18*, *AlexNet*, *MobileNet*, entre otras) y el análisis de su rendimiento bajo diferentes configuraciones (variación en el número de imágenes por clase, técnicas de normalización, uso de *data augmentation*, etc.). Además, se han acometido tareas no previstas en el anteproyecto original, como el desarrollo de una **aplicación web funcional** que permite la clasificación de imágenes desde una interfaz accesible, facilitando así la validación práctica del modelo en un entorno más realista.

Desde un punto de vista metodológico, uno de los aprendizajes más relevantes ha sido la importancia de seguir buenas prácticas en la preparación de los datos. El primer modelo, obtenido de un repositorio público (Kaggle), mostraba una precisión cercana al 100 %, pero su validez era cuestionable debido a errores metodológicos graves, como el uso no estratificado de los datos por paciente, lo que provocaba *data leakage*. Esta experiencia pone de manifiesto la necesidad de validar de forma rigurosa cualquier implementación externa.

Asimismo, se ha identificado que comenzar la experimentación con modelos de referencia establecidos en la literatura podría haber sido una estrategia más eficiente. Esto habría permitido construir líneas base comparables desde fases tempranas, en lugar de invertir tiempo inicialmente en modelos con poca documentación o no contrastados.

En relación con los resultados obtenidos, el enfoque más robusto ha sido el que aplica-

ba cortes axiales centrales de resonancia, una normalización estándar (media cero, desviación típica uno) y el uso del optimizador **Adam**. Esta configuración permitió a modelos como *ResNet18* y *Sivaranjani-CNN* alcanzar los valores más altos de **F1-score**, una métrica especialmente significativa en contextos con clases desbalanceadas. Además del buen rendimiento cuantitativo, estos modelos mostraron una mayor estabilidad durante el entrenamiento y una mejor capacidad de generalización frente al conjunto de test.

Por el contrario, arquitecturas como *AlexNet* o *MobileNet* presentaron un rendimiento más irregular, probablemente debido a su menor capacidad de representación para imágenes biomédicas complejas.

A lo largo del trabajo, se ha puesto a prueba no solo la capacidad de diferentes modelos de aprendizaje profundo, sino también la importancia de tomar decisiones bien fundamentadas en cada fase del proceso: desde la elección del conjunto de datos hasta los ajustes de entrenamiento. Más allá de los resultados obtenidos, esta experiencia ha servido para comprender mejor los retos reales de aplicar inteligencia artificial en el contexto biomédico, donde cada decisión puede tener un impacto significativo en la fiabilidad de las conclusiones.

7.2. Líneas Futuras

A partir del desarrollo y los resultados obtenidos en este trabajo, se identifican varias líneas de actuación que podrían explorarse en futuros estudios:

- **Uso del conjunto completo de ADNI:** En lugar de partir de un subconjunto ya estructurado, sería interesante trabajar directamente con el conjunto de datos original de ADNI. Esto permitiría mayor control sobre el preprocesamiento, la selección de cortes por paciente y el balance de clases, con el objetivo de mejorar la generalización de los modelos.
- **Validación cruzada:** Implementar técnicas como la validación cruzada (*k-fold cross-validation*) permitiría evaluar la estabilidad de los modelos frente a diferentes particiones del conjunto de datos, especialmente útil cuando el tamaño de los datos es limitado.
- **Aprovechamiento de metadatos clínicos:** El conjunto de datos incluía un *dataframe* con información adicional sobre los pacientes, como edad, sexo, diagnóstico clínico, etc. Incorporar estas variables como entrada auxiliar o como parte del análisis podría enriquecer los modelos y aportar una dimensión más completa al proceso de clasificación.
- **Aplicación de técnicas de aprendizaje auto-supervisado:** El uso de modelos pre-entrenados mediante técnicas como autoencoders o *contrastive learning* podría mejorar

el rendimiento en tareas de clasificación, especialmente en escenarios con pocos datos etiquetados.

- **Aumento de datos con métodos avanzados:** Experimentar con técnicas de aumento sintético más avanzadas, como las redes generativas adversarias (GANs), para generar ejemplos artificiales que refuercen las clases minoritarias y mitiguen problemas de desbalance.
- **Incorporación de datos multimodales:** Integrar información adicional disponible en ADNI (biomarcadores, datos clínicos, genética, etc.) podría enriquecer los modelos y aumentar la precisión en la clasificación.
- **Estudio de la interpretabilidad del modelo:** Aplicar técnicas como Grad-CAM o LIME para analizar qué zonas de las imágenes influyen más en la decisión del modelo, lo que contribuiría a una mayor transparencia y confianza en aplicaciones clínicas.
- **Pruebas con arquitecturas especializadas:** Explorar nuevas arquitecturas diseñadas específicamente para imágenes médicas (como Vision Transformers adaptados o modelos 3D convolucionales) que podrían capturar mejor la complejidad espacial de los datos de resonancia.

Referencias

- [1] Ingrid Arevalo-Rodriguez et al. “Mini-Mental State Examination (MMSE) for the detection of Alzheimer’s disease and other dementias in people with mild cognitive impairment (MCI)”. En: *Cochrane Database of Systematic Reviews* 5.3 (2015). DOI: [10.1002/14651858.CD010783](https://doi.org/10.1002/14651858.CD010783).
- [2] Anders Wimo y Martin Prince. *World Alzheimer Report 2010: The Global Economic Impact of Dementia*. <https://www.alzint.org/resource/world-alzheimer-report-2010/>. Alzheimer’s Disease International. 2010.
- [3] Kun Hu et al. “Multi-scale features extraction from baseline structure MRI for MCI patient classification and AD early diagnosis”. En: *Neurocomputing* 175 (2016), págs. 132-145. DOI: [10.1016/j.neucom.2015.10.002](https://doi.org/10.1016/j.neucom.2015.10.002).
- [4] Mohammad Ali Ebrahimighahnavieh, Shao Luo y Raymond Chiong. “Deep learning to detect Alzheimer’s disease from neuroimaging: A systematic literature review”. En: *Computer Methods and Programs in Biomedicine* 187 (2020), pág. 105242. DOI: [10.1016/j.cmpb.2019.105242](https://doi.org/10.1016/j.cmpb.2019.105242).
- [5] Mohammed G. Alsubaie, Shao Luo y Kamran Shaukat. “Alzheimer’s Disease Detection Using Deep Learning on Neuroimaging: A Systematic Review”. En: *Machine Learning and Knowledge Extraction* 6.1 (2024), págs. 464-505. DOI: [10.3390/make6010024](https://doi.org/10.3390/make6010024).
- [6] Alzheimer’s Disease Neuroimaging Initiative (ADNI). *ADNI Publications*. https://adni.loni.usc.edu/news-publications/publications/?q=classification&size=10&page_no=1. Accessed: 2025-04-10. 2024.
- [7] Python Software Foundation. *Python*. <https://www.python.org/>. 2023.
- [8] Project Jupyter. *Jupyter Notebook*. <https://jupyter.org/>. 2023.
- [9] Google Research. *Google Colaboratory*. <https://colab.research.google.com>. 2024.
- [10] Charles R. Harris, K. Jarrod Millman y Stéfan J. et al. van der Walt. “Array programming with NumPy”. En: *Nature* 585.7825 (2020), págs. 357-362. DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2).
- [11] Wes McKinney. “Data Structures for Statistical Computing in Python”. En: *Proceedings of the 9th Python in Science Conference*. 2010, págs. 56-61. DOI: [10.25080/Majora-92bf1922-00a](https://doi.org/10.25080/Majora-92bf1922-00a).
- [12] J. D. Hunter. “Matplotlib: A 2D Graphics Environment”. En: *Computing in Science & Engineering* 9.3 (2007), págs. 90-95. DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).

- [13] Michael Waskom. *Seaborn: Statistical Data Visualization*. <https://seaborn.pydata.org>. 2021.
- [14] Alex Clark y Contributors. *Pillow (PIL Fork)*. <https://python-pillow.org/>. 2023.
- [15] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. En: *Advances in Neural Information Processing Systems*. Vol. 32. 2019. URL: <https://pytorch.org>.
- [16] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort et al. “Scikit-learn: Machine Learning in Python”. En: *Journal of Machine Learning Research* 12 (2011), págs. 2825-2830.
- [17] Kaiming He et al. “Deep Residual Learning for Image Recognition”. En: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 2016, págs. 770-778.
- [18] Mark Sandler et al. “MobileNetV2: Inverted Residuals and Linear Bottlenecks”. En: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, págs. 4510-4520.
- [19] Alex Krizhevsky, Ilya Sutskever y Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. En: *Advances in Neural Information Processing Systems*. Vol. 25. 2012.
- [20] Kaggle. *Kaggle: Your Machine Learning and Data Science Community*. <https://www.kaggle.com>. 2025.
- [21] Manuel H Janeiro et al. “Biomarkers in Alzheimer’s disease”. En: *Advances in Laboratory Medicine* 2.1 (2021), págs. 27-50. DOI: [10.1515/almed-2020-0090](https://doi.org/10.1515/almed-2020-0090). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10197496/>.
- [22] Esther E Bron et al. “Standardized evaluation of algorithms for computer-aided diagnosis of dementia based on structural MRI: the CADDementia challenge”. En: *NeuroImage* 111 (2015), págs. 562-579.
- [23] Mohammad Reza Arbabshirani et al. “Single subject prediction of brain disorders in neuroimaging: promises and pitfalls”. En: *NeuroImage* 145 (2017), págs. 137-165.
- [24] Geert Litjens et al. “A survey on deep learning in medical image analysis”. En: *Medical Image Analysis* 42 (2017), págs. 60-88.
- [25] MD.Saúde. *Diferença entre cérebro normal e com Alzheimer*. Accedido el 12 de junio de 2025. 2020. URL: <https://www.mdsaude.com/wp-content/uploads/Alzheimer-cerebro-es.jpg>.
- [26] Adrien Payan y Giovanni Montana. “Predicting Alzheimer’s disease: a neuroimaging study with 3D convolutional neural networks”. En: *Deep learning and representation learning workshop, NIPS*. Vol. 2015. 2015, págs. 1-6.

- [27] Frederik Barkhof, Philip Scheltens y Wiesje M. van der Flier. “Neuroimaging in dementia”. En: *Journal of Neurology* 266.2 (2018), págs. 382-406. DOI: [10.1007/s00415-018-9016-3](https://doi.org/10.1007/s00415-018-9016-3). URL: <https://doi.org/10.1007/s00415-018-9016-3>.
- [28] S. Pandey y et al. “Random Forest Model for Feature-based Alzheimer’s Disease Prediction”. En: *NeuroImage* (2019).
- [29] Alessia Sarica, Antonio Cerasa y Aldo Quattrone. “Random forest model for Alzheimer’s disease classification using neuroimaging data”. En: *Frontiers in Aging Neuroscience* 10 (2018), págs. 13. DOI: [10.3389/fnagi.2018.00013](https://doi.org/10.3389/fnagi.2018.00013).
- [30] Noura Alsubaie et al. “Deep learning and neuroimaging biomarkers for Alzheimer’s disease: a systematic review”. En: *Frontiers in Aging Neuroscience* (2023).
- [31] Hrushi Vyas. *How to Visualize the Layers of a Deep Learning Model*. Accedido el 12 de junio de 2025. 2020. URL: <https://medium.com/@hrushi669/how-to-visualize-the-layers-of-a-deep-learning-model-30d3fd7de7e5>.
- [32] Ehsan Hosseini-Asl, Georgy Gimel’farb y Ayman El-Baz. “Alzheimer’s Disease Diagnostics by a Deeply Supervised Adaptable 3D Convolutional Network”. En: *Frontiers in Bioscience* 23 (2018), págs. 584-596.
- [33] Connor Shorten y Taghi M. Khoshgoftaar. “A survey on Image Data Augmentation for Deep Learning”. En: *Journal of Big Data* 6.60 (2019).
- [34] Thien Nhan Vo, Bac Nam Ho y Thanh Xuan Truong. “3D Brain MRI Classification for Alzheimer’s Diagnosis Using CNN with Data Augmentation”. En: *arXiv* (2025). DOI: [10.48550/arXiv.2505.04097](https://doi.org/10.48550/arXiv.2505.04097).
- [35] Thien Nhan Vo, Bac Nam Ho y Thanh Xuan Truong. “3D Brain MRI Classification for Alzheimer’s Diagnosis Using CNN with Data Augmentation”. En: *arXiv preprint arXiv:2505.04097* (2025).
- [36] Author et al. “The role of generative adversarial networks in brain MRI: a scoping review”. En: *Insights into Imaging* (2022).
- [37] Author et al. “Enhancing magnetic resonance imaging-driven Alzheimer’s disease classification using GAN”. En: *Alzheimer’s Research Therapy* 13 (2021).
- [38] Sarah Oraby et al. “Hybrid of DSR-GAN and CNN for Alzheimer disease detection based on MRI images”. En: *Scientific Reports* 15 (2025).
- [39] Daniel S. Marcus et al. “Open Access Series of Imaging Studies (OASIS): Cross-sectional MRI Data in Young, Middle Aged, Nondemented, and Demented Older Adults”. En: *Journal of Cognitive Neuroscience* 19.9 (2007), págs. 1498-1507. DOI: [10.1162/jocn.2007.19.9.1498](https://doi.org/10.1162/jocn.2007.19.9.1498). URL: <https://www.oasis-brains.org>.

- [40] Yiwei Lu. *Well Documented Alzheimer's Dataset*. <https://www.kaggle.com/datasets/yiweilu2033/well-documented-alzheimers-dataset>. Dataset accessed on 2025-05-13. 2023.
- [41] Naeem Nagori. *Alzheimer Classification*. <https://www.kaggle.com/code/naeemnagori/alzheimer-classification>. Último acceso: mayo de 2025. 2022.
- [42] Sanjiban Sekhar Roy, Raghav Sikaria y Aarti Susan. "A deep learning based CNN approach on MRI for Alzheimer's disease detection". En: *Intelligent Decision Technologies* 13.4 (2019), págs. 495-505. DOI: [10.3233/IDT-190005](https://doi.org/10.3233/IDT-190005).
- [43] S Sivaranjani, V Thavavel y S Kaja Mohideen. "A CNN based framework for classification of Alzheimer's disease". En: *Materials Today: Proceedings* 47.17 (2021), págs. 5765-5769. DOI: [10.1016/j.matpr.2021.06.245](https://doi.org/10.1016/j.matpr.2021.06.245).
- [44] Fanar E. K. Al-Khuzai, Oguz Bayat y Adil D. Duru. "Diagnosis of Alzheimer Disease Using 2D MRI Slices by Convolutional Neural Network". En: *Applied Bionics and Biomechanics* 2021 (2021), págs. 1-9. DOI: [10.1155/2021/6690539](https://doi.org/10.1155/2021/6690539).
- [45] Gowhar Mohi ud din Dar et al. "A Novel Framework for Classification of Different Alzheimer's Disease Stages Using CNN Model". En: *Sensors* 22.24 (2022), pág. 9690. DOI: [10.3390/s22249690](https://doi.org/10.3390/s22249690).

Apéndice A

Manual de Instalación

En esta sección se detalla el proceso para preparar el entorno de trabajo y descargar los datos necesarios para el proyecto.

A.1. Requisitos previos

Es necesario tener instalado Python 3.7 o superior. Se recomienda utilizar un entorno virtual para evitar conflictos entre librerías.

A.2. Descarga del repositorio

Para comenzar, descargue el repositorio desde GitHub ejecutando el siguiente comando en la terminal:

```
git clone https://github.com/martacuevasr/Clasification_AlzheimerImages.git
cd Clasification_AlzheimerImages
```

Este repositorio contiene un archivo `README.md` con todas las instrucciones necesarias para la instalación, preparación de datos y ejecución de la interfaz web.

A.3. Instalación de dependencias

En el directorio raíz del proyecto se encuentra el archivo `requirements.txt` que contiene las librerías necesarias para ejecutar el script de descarga y preparación de los datos.

Para instalar estas dependencias, ejecute el siguiente comando:

```
pip install -r requirements.txt
```

A.4. Descarga y preparación de los datos

Una vez instaladas las dependencias, debe ejecutar el script `setup_data.py` que se encarga de descargar, descomprimir y organizar el conjunto de datos en las carpetas adecuadas.

Ejecute el siguiente comando en la terminal desde el directorio raíz del proyecto:

```
python setup.py
```

Este script realizará las siguientes acciones:

- Descargar el conjunto de datos desde la fuente indicada.
- Descomprimir los archivos si están comprimidos.
- Crear la estructura de carpetas necesaria para almacenar las imágenes y anotaciones.

A.5. Notas adicionales

- El script `setup_data.py` requiere conexión a Internet para descargar los datos.
- Se recomienda ejecutar el script en un entorno con suficiente espacio de almacenamiento.
- Si se desea utilizar la interfaz gráfica para realizar predicciones, consulte la sección *Manual de Uso de la Interfaz*, donde se explican los pasos para instalar las dependencias y ejecutar la aplicación.

A.6. Interfaz de Usuario

La carpeta `interfaz` contiene el archivo principal para ejecutar la aplicación gráfica basada en `Streamlit`: `app.py`.

Además, en esta misma carpeta se encuentran los archivos con los pesos preentrenados de los modelos, necesarios para realizar las predicciones:

- `3slicesPaper2Normalization01.pth` — pesos para el modelo AlzheimerCNN (Sivaranjani-CNN).
- `3slicesResNetNormalization01.pth` — pesos para el modelo ResNet18.

Para ejecutar la interfaz, se debe tener instalado el entorno con las dependencias indicadas en `requirements.txt`, que incluye `streamlit`, `torch`, `torchvision`, entre otras.

Una vez preparado el entorno, para lanzar la aplicación basta con abrir una terminal, situarse en la carpeta `interfaz` y ejecutar:

```
streamlit run app.py
```

Esto abrirá una ventana o pestaña en el navegador desde donde se puede subir una imagen, seleccionar el modelo deseado (Sivaranjani-CNN o ResNet18) y obtener la predicción sobre el grado de demencia.

Es importante que los archivos de pesos permanezcan en la misma carpeta que `app.py`, o en la ruta especificada en el código para poder cargarlos correctamente.

Apéndice B

Manual de Usuario

A continuación se detallan los pasos a seguir para ejecutar cada una de las funcionalidades de la interfaz web.

En primer lugar, el usuario debe seleccionar el modelo de clasificación que desea emplear entre las dos opciones disponibles: *Sivaranjani-CNN* y *ResNet18*. Esta selección se realiza mediante un botón de opción (radio button), como se muestra en la Figura 60.

Clasificador de Alzheimer por Imágenes

Sube una imagen y elige el modelo para predecir el grado de demencia

Selecciona el modelo que quieres usar:

- ☒ Sivaranjani-CNN
☐ ResNet18

Elige una imagen...

 Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

Browse files

Figura 60: Selección del modelo a utilizar en la interfaz.

A continuación, el usuario debe cargar una imagen de resonancia magnética cerebral (MRI) para ser analizada. La carga de la imagen se realiza desde el navegador mediante el botón “*Elige una imagen...*”, el cual permite seleccionar un archivo con formato JPG, JPEG o PNG. La imagen cargada se muestra en pantalla para su confirmación, como se ilustra en la Figura 61.

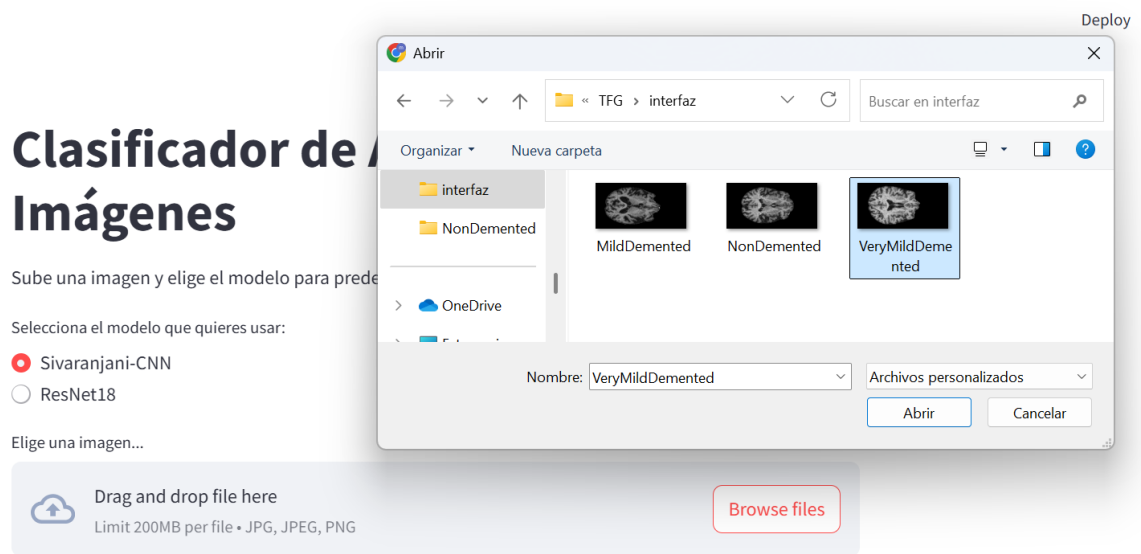


Figura 61: Carga de imagen MRI a través de la interfaz web.

Una vez seleccionados el modelo y la imagen, la aplicación procesa la imagen mediante las transformaciones definidas previamente, y ejecuta la inferencia con el modelo escogido. Finalmente, la interfaz muestra las probabilidades obtenidas para cada clase de demencia, facilitando así la interpretación del grado estimado de deterioro cognitivo para la imagen suministrada, como se muestra en la Figura 62.

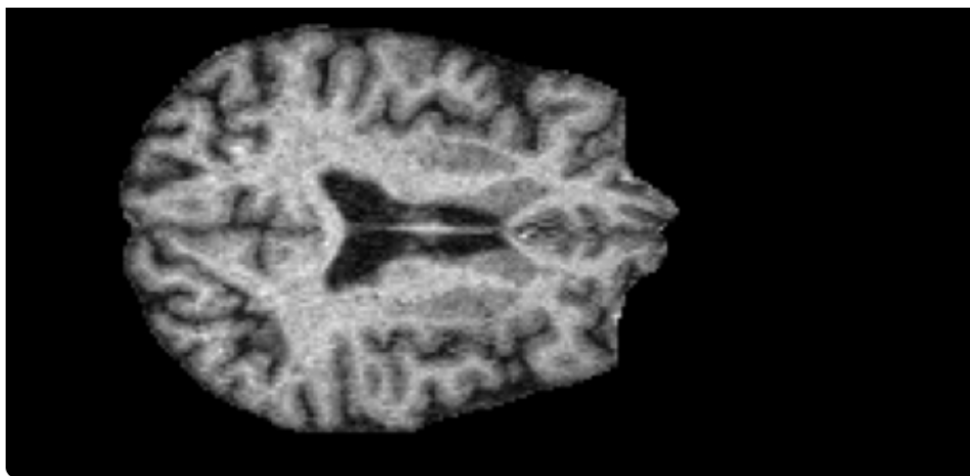


Imagen cargada

Resultados de la predicción

NonDemented: 11.87%

VeryMildDemented: 87.07%

MildDemented: 1.07%

Figura 62: Probabilidades predichas para cada clase tras la inferencia.



UNIVERSIDAD
DE MÁLAGA

| **uma.es**

E.T.S. DE INGENIERÍA INFORMÁTICA

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga