



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA BIOMÉDICA

**DISEÑO E IMPLEMENTACIÓN DE UN
CLASIFICADOR PARA LA DETECCIÓN DE
LA ENFERMEDAD DE ALZHEIMER
MEDIANTE EL USO DE TÉCNICAS DE
INTELIGENCIA ARTIFICIAL EN IMÁGENES
DE RESONANCIA MAGNÉTICA**

AUTORA: Irene Aleixandre Soriano

TUTOR: Ignacio Bosch Roig

COTUTORES: Eduardo Camacho Ramos
José Sánchez García

Curso Académico: 2019-20

Resumen

El Alzheimer es una enfermedad neurodegenerativa considerada como la mayor forma de demencia en el mundo. Esta patología tiene graves consecuencias en el desarrollo diario de la persona, pudiendo afectar su razonamiento y su memoria hasta el punto de perder la capacidad de realizar las tareas más cotidianas. Pese a que, actualmente, no existe una cura para esta enfermedad, sí que existen tratamientos capaces de retrasar o paliar sus síntomas si es detectada a tiempo. Por este motivo, es necesario el desarrollo de herramientas capaces de facilitar su detección precoz.

En este trabajo se pretende abordar el problema desde el campo de la Inteligencia Artificial, concretamente mediante técnicas de aprendizaje profundo. Para ello, y haciendo uso de imágenes de resonancia magnética procedentes de la base de datos ADNI (Alzheimer's Disease Neuroimaging Initiative), se implementará un clasificador basado en redes neuronales convolucionales (CNN). El objetivo principal de esta herramienta será el cribado de casos de imagen de resonancia magnética para diferenciar casos de demencia con respecto a sujetos sanos.

Palabras clave: Redes neuronales; Alzheimer; aprendizaje profundo; resonancia magnética cerebral; clasificador; modelo; atrofia cerebral; inteligencia artificial.

Resum

L'Alzheimer és una malaltia neurodegenerativa considerada com la major forma de demència en el món. Esta patologia té greus conseqüències en el desenvolupament diari de la persona, podent afectar el seu raonament i la seua memòria fins al punt de perdre la capacitat de realitzar les tasques més quotidianes. Malgrat que, actualment, no hi ha una cura per a aquesta malaltia, sí que hi ha tractaments capaços de retardar o pal·liar els seus símptomes si és detectada a temps. Per aquest motiu, és necessari el desenvolupament de eines capaces de facilitar la seua detecció precoç.

En aquest treball es pretén abordar el problema des del camp de la Intel·ligència Artificial, concretament per mitjà de tècniques d'aprenentatge profund. Per a això, i fent ús d'imatges de ressonància magnètica procedents de la base de dades ADNI (Alzheimer's Disease Neuroimaging Initiative), s'implementarà un classificador basat en xarxes neuronals convolucionals (CNN). L'objectiu principal d'aquesta eina serà el garbellament de casos d'imatge de ressonància magnètica per a diferenciar casos de demència respecte a subjectes sans.

Paraules clau: Xarxes neuronals; Alzheimer; aprenentatge profund; ressonància magnètica cerebral; classificador; model; atròfia cerebral; intel·ligència artificial.

Abstract

Alzheimer is a neurodegenerative disease considered to be the largest form of dementia in the world. This pathology has serious consequences on the daily's person development, affecting its reasoning and its memory up to the point of losing the ability to carry out everyday tasks. Despite the fact that currently there is no cure for this disease, there are treatments that can delay or alleviate its symptoms if it is detected in time. For this reason, it is necessary to develop tools capable of facilitating its early detection.

The purpose of this project is to tackle the problem from the field of Artificial Intelligence, specifically through deep learning techniques. To this end, making use of magnetic resonance images from the ADNI (Alzheimer's Disease Neuroimaging Initiative) database, a classifier based on convolutional neural networks (CNN) will be implemented. The main objective of this tool will be the screening of magnetic resonance imaging cases to differentiate dementia cases from healthy subjects.

Keywords: Neural networks; Alzheimer; deep learning; brain magnetic resonance imaging; classifier; model; brain atrophy; artificial intelligence.

Índice

Documentos contenidos en el TFG

- Memoria
- Presupuesto

MEMORIA

Diseño e implementación de un clasificador para la detección de la enfermedad de Alzheimer mediante el uso de técnicas de inteligencia artificial en imágenes de resonancia magnética.

Índice de la memoria

CAPÍTULO 1. MARCO TEÓRICO.....	1
1.1. <i>Alzheimer</i>	1
Regiones cerebrales mayormente afectadas.....	1
1.2. <i>Neuroimagen estructural</i>	3
1.3. <i>Inteligencia artificial</i>	5
1.3.1. Neurona artificial.....	6
1.3.2. Red neuronal artificial.....	7
1.3.3. Red neuronal convolucional.....	8
A) Capa convolucional.....	9
B) Función de activación.....	10
C) Capa de reducción.....	11
D) Capa completamente conectada.....	12
E) Arquitecturas CNN.....	12
AlexNet.....	13
VGGNet.....	14
ResNet.....	14
DenseNet.....	15
SqueezeNet.....	16
1.3.4. Entrenamiento de las CNN.....	17
A) Propagación direct.....	17
B) Función de pérdidas.....	17
C) Propagación inversa.....	18
D) Tamaño de lote.....	19
E) Epochs.....	19
Congelación de las primeras capas de la CNN.....	19
1.4. <i>Estructura de la memoria</i>	19
CAPÍTULO 2. OBJETIVOS Y MOTIVACIÓN.....	20
2.1. <i>Objetivos</i>	20
2.2. <i>Motivación</i>	20
CAPÍTULO 3. ESTADO DEL ARTE.....	22
CAPÍTULO 4. MATERIALES Y MÉTODOS.....	24
4.1. <i>Adquisición de los datos</i>	24
4.2. <i>Herramientas software</i>	26
4.2.1. MATLAB.....	26
4.2.2. SPM (Statistical Parametric Mapping).....	27
4.2.3. Python.....	27
4.2.4. Fastai.....	28
4.3. <i>Métricas</i>	28
4.3.1. Matriz de confusión.....	28
4.3.2. Exactitud.....	29
4.3.3. Precisión.....	30
4.3.4. Sensibilidad.....	30

4.3.5. Especificidad.....	30
4.3.6. <i>F1 score</i>	30
4.3.7. Curva ROC	31
4.3.8. Área bajo la curva ROC	31
4.4. <i>Pre-procesamiento</i>	32
4.4.1. Creación de las imágenes 2D.....	32
4.4.2. <i>Zero padding</i>	33
4.4.3. Extracción de cráneo.....	33
A) Segmentación.....	33
B) Extracción de cráneo	35
4.4.4. Normalización de intensidad	36
4.4.5. Eliminación de cortes axiales con ausencia de tejido cerebral.....	37
4.4.6. Eliminación de cortes axiales con una media de intensidad menor a 0.05.....	38
4.4.7. Eliminación de cortes axiales sin información anatómica de interés para la detección de AD.....	39
4.5. <i>Experimentos previos</i>	41
4.5.1. <i>Zero padding</i>	42
4.5.2. Extracción de cráneo.....	42
4.5.3. Tamaño de imagen	43
4.5.4. Normalización de intensidad	43
4.5.5. Eliminación de los cortes axiales con ausencia de tejido cerebral.....	44
4.5.6. Eliminación de los cortes axiales con una media de intensidad menor a 0.05	44
4.5.7. Eliminación de los cortes axiales sin información anatómica relevante para la detección de AD.....	45
4.5.8. Elección de la arquitectura CNN.....	45
4.5.9. Elección del <i>batch size</i>	46
4.6. <i>Experimentos finales</i>	47
4.6.1. Clasificador binario.....	47
4.6.2. Clasificador multiclase	48
4.6.3. Clasificador multiclase en cascada.....	49
CAPÍTULO 5. RESULTADOS.....	50
5.1. <i>Experimentos previos</i>	50
5.1.1. <i>Zero padding</i>	50
5.1.2. Extracción de cráneo.....	51
5.1.3. Tamaño de imagen	51
5.1.4. Normalización de intensidad	52
5.1.5. Eliminación de los cortes axiales con ausencia de tejido cerebral.....	52
5.1.6. Eliminación de los cortes axiales con una media de intensidad menor a 0.05	53
5.1.7. Eliminación de los cortes axiales sin información anatómica relevante para la detección de AD.....	53
5.1.8. Elección de la arquitectura CNN.....	54
5.1.9. Elección del <i>batch size</i>	55
5.2. <i>Experimentos finales</i>	56
5.2.1. Clasificador binario.....	56
5.2.2. Clasificador multiclase	58
5.2.3. Clasificador multiclase en cascada.....	63

CAPÍTULO 6. DISCUSIÓN.....	70
CAPÍTULO 7. CONCLUSIONES Y LÍNEAS FUTURAS.	73
Bibliografía.....	74

Índice de tablas

Tabla 1. Número de parámetros presentes en cada una de las <i>CNN</i> . Ordenados de menor a mayor según el número de parámetros totales.	13
Tabla 2. Clasificación terciaria. Número de sujetos asociados a cada etiqueta.	25
Tabla 3. Clasificación binaria. Número de sujetos asociados a cada etiqueta.	26
Tabla 4. Resumen para el caso concreto del clasificador binario: falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos.	29
Tabla 5. Clasificación multiclase. Número de imágenes <i>.png</i> asociadas a cada etiqueta.	32
Tabla 6. Clasificación binaria. Número de imágenes <i>.png</i> asociadas a cada etiqueta.	33
Tabla 7. Clasificación multiclase. Número de imágenes <i>.png</i> asociadas a cada etiqueta tras eliminar los cortes axiales con ausencia de tejido cerebral.	37
Tabla 8. Clasificación binaria. Número de imágenes <i>.png</i> asociadas a cada etiqueta tras eliminar los cortes axiales con ausencia de tejido cerebral.	37
Tabla 9. Clasificación multiclase. Número de imágenes <i>.png</i> asociadas a cada etiqueta tras eliminar los cortes axiales con una media de intensidad menor a 0.05.	38
Tabla 10. Clasificación binaria. Número de imágenes <i>.png</i> asociadas a cada etiqueta tras eliminar los cortes axiales con una media de intensidad menor a 0.05.	38
Tabla 11. Clasificación multiclase. Número de imágenes <i>.png</i> asociadas a cada etiqueta tras eliminar los cortes axiales sin información anatómica de interés para la AD.	40
Tabla 12. Clasificación binaria. Número de imágenes <i>.png</i> asociadas a cada etiqueta tras eliminar los cortes axiales sin información anatómica de interés para la AD.	41
Tabla 13. Experimento binario. Número de imágenes que forman cada subconjunto de datos: entrenamiento, validación y <i>test</i>	47
Tabla 14. Experimento multiclase. Número de imágenes que forman cada subconjunto de datos: entrenamiento, validación y <i>test</i>	48
Tabla 15. Experimento multiclase. Subconjunto de <i>test</i> para evaluar CN.	48
Tabla 16. Experimento multiclase. Subconjunto de <i>test</i> para evaluar MCI.	48
Tabla 17. Experimento multiclase. Subconjunto de <i>test</i> para evaluar AD.	48
Tabla 18. Clasificador MCI - AD. Número de imágenes que forman cada subconjunto de datos: entrenamiento, validación y <i>test</i>	49
Tabla 19. Resultados experimento previo 1. Sin <i>zero padding</i>	50
Tabla 20. Resultados experimento previo 1. Con <i>zero padding</i>	50
Tabla 21. Resultados experimento previo 2. Con extracción de cráneo.	51
Tabla 22. Resultados experimento previo 3. 128x128.	51
Tabla 23. Resultados experimento previo 3. 128x128 → 256x256	52
Tabla 24. Resultados experimento previo 4. Con normalización de intensidad.	52
Tabla 25. Resultados experimento previo 5. Con eliminación de los cortes axiales con ausencia de tejido cerebral.	53
Tabla 26. Resultados experimento previo 6. Eliminación de los cortes axiales con una media de intensidad menor a 0.05.	53
Tabla 27. Resultados experimento previo 7. Eliminación de los cortes axiales sin información anatómica relevante para la detección de AD.	54
Tabla 28. Resultados experimento previo 8. <i>ResNet50</i>	54
Tabla 29. Resultados experimento previo 8. <i>Squeezenet1_1</i>	55
Tabla 30. Resultados experimento previo 8. <i>DenseNet121</i>	55
Tabla 31. Resultados experimento previo 9. <i>Batch size</i> mayor, 128.	56
Tabla 32. Resultados experimento previo 9. <i>ResNet50</i> . <i>Batch size</i> menor, 32.	56

Tabla 33. Clasificador binario, <i>DenseNet121</i> . Capas congeladas.	57
Tabla 34. Clasificador binario, <i>DenseNet121</i> . Capas descongeladas.	57
Tabla 35. Clasificador binario. Métricas subconjunto de <i>test</i>	58
Tabla 36. Clasificador multiclase, <i>DenseNet121</i> . Capas congeladas.	59
Tabla 37. Clasificador multiclase, <i>DenseNet121</i> . Capas descongeladas.	59
Tabla 38. Clasificador multiclase. Métricas subconjunto de <i>test</i> CN vs MCI+AD.	60
Tabla 39. Clasificador multiclase. Métricas subconjunto de <i>test</i> MCI vs CN+AD.	61
Tabla 40. Clasificador multiclase. Métricas subconjunto de <i>test</i> AD vs MCI+CN.	62
Tabla 41. Clasificador MCI - AD, <i>DenseNet121</i> . Capas congeladas.	63
Tabla 42. Clasificador MCI - AD, <i>DenseNet121</i> . Capas descongeladas.	64
Tabla 43. Clasificador MCI - AD. Métricas subconjunto de <i>test</i>	65
Tabla 44. Clasificador multiclase en cascada. Métricas subconjunto de <i>test</i> CN vs MCI+AD.	66
Tabla 45. Clasificador multiclase en cascada. Métricas subconjunto de <i>test</i> MCI vs CN+AD.	67
Tabla 46. Clasificador multiclase en cascada. Métricas subconjunto de <i>test</i> AD vs MCI+CN.	68
Tabla 47. Resumen de los valores de exactitud obtenidos para cada uno de los experimentos previos realizados.	70
Tabla 48. Resumen de las métricas obtenidas con los subconjuntos de datos de <i>test</i> para los clasificadores multiclase.	71

Índice de figuras

Figura 1. Diferencias entre un cerebro normales y cerebros con enfermedad de Alzheimer. Sección transversal. [6]	2
Figura 2. Diferencias en la estructura del hipocampo en imágenes de resonancia magnética en sujetos cognitivos normales (CN), con deterioro cognitivo leve (MCI) y con enfermedad de Alzheimer (AD). [9].....	2
Figura 3. Estudio de la Relajación siguiendo la evolución en el tiempo de la componente longitudinal M_z y de la componente transversal $M_{x,y}$. [14]	4 4
Figura 4. Imágenes de RMN potenciadas en: (a) densidad, (b) T1 y (c) T2.	5
Figura 5. Diferencias entre inteligencia artificial, aprendizaje automático y aprendizaje profundo. [18].....	6
Figura 6. Esquema básico de la estructura de una neurona artificial. [20].....	6
Figura 7. Esquema básico de la estructura de una neurona biológica. [21].....	7
Figura 8. Esquema básico de una red neuronal artificial.....	7
Figura 9. Estructura básica de una red neuronal convolucional. [24]	8
Figura 10. Operación de convolución en una imagen 2D. Camino seguido por el <i>kernel</i> sobre los píxeles de la imagen de entrada para realizar la convolución.....	9
Figura 11. <i>Zero padding</i> :	10
Figura 12. Creación del mapa de características. Cada filtro aplicado sobre la imagen nos aportará un nivel de profundidad sobre el mapa de características. [25]	10
Figura 13. Función de activación ReLU.	11
Figura 14. Otras funciones de activación.[26].....	11
Figura 15. Ejemplo de la técnica max-pooling.	12
Figura 16. Arquitectura AlexNet. [28]	14
Figura 17. Arquitectura VGGNet. [29]	14
Figura 18. Arquitectura ResNet18. [31]	15
Figura 19. Arquitectura DenseNet. [33]	16
Figura 20. Arquitectura SqueezeNet. [35]	17
Figura 21. Estimación de la incidencia de demencia en mayores de 60 años. [39].....	21
Figura 22. Envejecimiento de la población española estimada para los próximos años según el Instituto Nacional de Estadística. [41]	21
Figura 23. Alzheimer en los próximos 20 años. De 27 millones a 60 millones de enfermos a nivel mundial. [42].....	21
Figura 24. Clasificación a priori de los 199 sujetos iniciales según la base de datos ADNI. ...	24
Figura 25. Sexo de los 150 sujetos finales según la base de datos ADNI.	25
Figura 26. Edad de los 150 sujetos finales según la base de datos ADNI.	25
Figura 27. Gráfico de barras: número de sujetos asociados a cada etiqueta según el sexo.	26
Figura 28. Estructura de una matriz de confusión.	29
Figura 29. Ejemplo de curva ROC. [61].....	31
Figura 30. Dimensiones de los volúmenes 3D. La tercera dimensión se corresponde con el número de cortes, y la primera y segunda dimensión con el tamaño de la imagen .png.	32
Figura 31. Resultado del relleno con ceros.....	33
Figura 32. Resultado de la segmentación en tejidos: izquierda, sustancia gris; centro, sustancia blanca; derecha: líquido cefalorraquídeo.	35
Figura 33. Máscara binaria utilizada para la extracción de cráneo.	36

Figura 34. Resultado final tras aplicar <i>zero padding</i> y extracción de cráneo.....	36
Figura 35. Resultado final tras aplicar <i>zero padding</i> y normalización de intensidad.	37
Figura 36. Corte axial con una media de intensidad igual a 0.037 (menor a 0.05).	38
Figura 37. Clasificador en cascada.....	49
Figura 38. Clasificador binario. Matriz de confusión subconjunto de validación.	57
Figura 39. Clasificador binario. Curva ROC subconjunto de <i>test</i>	58
Figura 40. Clasificador binario. Matriz de confusión subconjunto de <i>test</i>	58
Figura 41. Clasificador multiclase. Matriz de confusión subconjunto de validación.....	60
Figura 42. Clasificador multiclase. Curva ROC subconjunto de <i>test</i> CN vs MCI+AD.	60
Figura 43. Clasificador multiclase. Matriz de confusión subconjunto de <i>test</i> CN vs MCI+AD.	61
Figura 44. Clasificador multiclase. Curva ROC subconjunto de <i>test</i> MCI vs CN+AD.	61
Figura 45. Clasificador multiclase. Matriz de confusión subconjunto de <i>test</i> MCI vs CN+AD.	62
Figura 46. Clasificador multiclase. Curva ROC subconjunto de <i>test</i> AD vs MCI+CN.	62
Figura 47. Clasificador multiclase. Matriz de confusión subconjunto de <i>test</i> AD vs MCI+CN.	63
Figura 48. Clasificador MCI - AD. Matriz de confusión subconjunto de validación.	64
Figura 49. Clasificador MCI - AD. Curva ROC subconjunto de <i>test</i>	65
Figura 50. Clasificador MCI - AD. Matriz de confusión subconjunto de <i>test</i>	65
Figura 51. Clasificador multiclase en cascada. Curva ROC subconjunto de <i>test</i> CN vs MCI+AD.	66
.....	67
Figura 52. Clasificador multiclase en cascada. Matriz de confusión subconjunto de <i>test</i> CN vs MCI+AD.	67
Figura 53. Clasificador multiclase en cascada. Curva ROC subconjunto de <i>test</i> MCI vs CN+AD.	67
Figura 54. Clasificador multiclase en cascada. Matriz de confusión subconjunto de <i>test</i> MCI vs CN+AD.....	68
Figura 55. Clasificador multiclase en cascada. Curva ROC subconjunto de <i>test</i> AD vs MCI+CN.	69
Figura 56. Clasificador multiclase en cascada. Matriz de confusión subconjunto de <i>test</i> AD vs MCI+CN.	69

CAPÍTULO 1. MARCO TEÓRICO.

1.1. Alzheimer

La enfermedad de Alzheimer (AD) se caracteriza neuropatológicamente por la aparición de depósitos o placas de la proteína β -amiloide extracelulares y ovillos neurofibrilares compuestos por la proteína tau hiperfosforilada (τ) en la zona intracelular. Esto se traduce en la pérdida de neuronas y sinapsis principalmente en la corteza cerebral, aunque también en ciertas regiones subcorticales. [1], [2]

Se trata de una enfermedad crónica de naturaleza degenerativa y evolución progresiva. Es el tipo de demencia más frecuente representando el 70% de los casos, y puede presentarse tanto de forma esporádica como debido a antecedentes familiares. Concretamente en España, entre 500.000 y 800.000 personas sufren esta enfermedad y se prevé que en 2050 la cifra se duplique, según datos proporcionados por CEAFA (Confederación Española de Asociaciones de Familiares de Personas con Alzheimer), como consecuencia del envejecimiento de la población. [3]

Cabe destacar que, en la actualidad, se trata de una enfermedad sin cura cuyo tratamiento no está bien definido. Pese a ello, varios estudios demuestran que algunos factores saludables como el ejercicio físico regular o seguir una dieta saludable pueden prevenir o retrasar su aparición. [4]

En cuanto a tratamientos se refiere, existen dos grandes grupos: tratamientos no farmacológicos/alternativos y tratamientos farmacológicos que afectan sobre los sistemas de neurotransmisión. El primero de los grupos abarca, por una parte, la intervención neuropsicológica que, pese a no retrasar la enfermedad, sí permite desarrollar nuevas redes funcionales cerebrales que puedan compensar la disfunción de otras redes. Por otra parte, engloba también el concepto de terapia ocupacional tratando de dotar al paciente de autonomía en su vida diaria. [4]

Regiones cerebrales mayormente afectadas

Si realizamos una visión general, se puede decir que la enfermedad de Alzheimer ataca en primer lugar, al sistema límbico (amígdala, tálamo... pero sobre todo al hipocampo), afectando tanto a la memoria como a cambios en el estado de ánimo. [5]

Posteriormente, ataca a la corteza cerebral, conllevando esto dependencias en cuanto a tareas diarias se refiere, como puede ser comer o vestirse. Los cambios en la corteza frontal producen, sobre todo, problemas que afectan a las funciones motoras; en la corteza temporal, influye en la comprensión del habla; y, en la corteza parietal, afecta al procesamiento de estímulos y coordinación de movimientos. [5]

Finalmente, en etapas avanzadas consigue atacar el tronco encefálico perjudicando a la actividad de los órganos del cuerpo. [5]

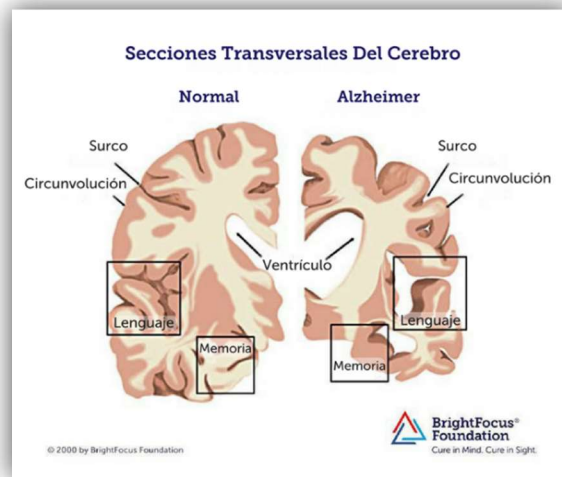


Figura 1. Diferencias entre un cerebro normal y cerebros con enfermedad de Alzheimer. Sección transversal. [6]

Siendo más específicos, se debe conocer los estudios de la progresión de la enfermedad de Alzheimer se focalizan, actualmente, en la atrofia hipocámpal, ya que es el primer lugar afectado como se comentaba anteriormente. Si nos centramos en la corteza, la atrofia empieza a disminuir en el lóbulo temporal medio, siguiendo con una atrofia extendida al resto de la corteza cerebral siguiendo la trayectoria temporal-parietal-frontal. [7], [8]

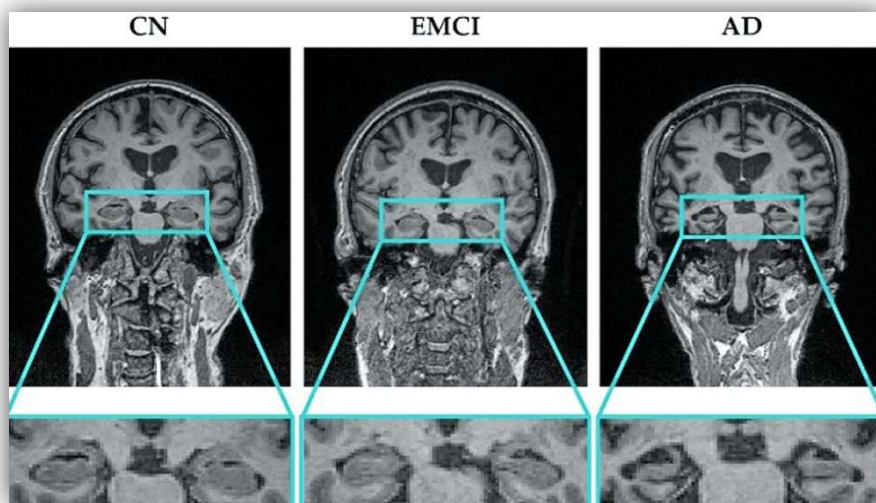


Figura 2. Diferencias en la estructura del hipocampo en imágenes de resonancia magnética en sujetos cognitivos normales (CN), con deterioro cognitivo leve (MCI) y con enfermedad de Alzheimer (AD). [9]

Si nuestro objetivo es la detección precoz de la enfermedad de Alzheimer, desde mediados de los años 80 se sabe que la atrofia en el lóbulo temporal es la más sensible en la detección de alteraciones tempranas. [10]

Para un mayor nivel de especificidad, además de las estructuras anteriormente nombradas también se observan cambios relevantes en la corteza temporal lateral, el cíngulo anterior, la corteza entorrinal y el giro parahipocámpal. [10], [11]

Todos estos cambios referidos a atrofas conllevan una disminución del volumen de las estructuras mencionadas, dejando espacio libre que debe ser compensado para preservar la presión intracraneal. Por este motivo, también se observa como consecuencia un aumento del volumen de líquido cefalorraquídeo en los surcos y ventrículos laterales, concretamente el asta temporal del ventrículo lateral. [10], [11]

1.2. Neuroimagen estructural.

La neuroimagen se define como el conjunto de técnicas que tienen como objetivo obtener imágenes cerebrales que aporten información, acerca de la estructura, función o farmacología del sistema nervioso. En el entorno clínico, los profesionales dedicados a su interpretación son los neuro radiólogos.

Actualmente, estas técnicas se dividen en dos grandes grupos: funcional y estructural. La neuroimagen funcional trata de diferenciar zonas funcionalmente activas, comúnmente a través de relaciones estímulo-respuesta. Las técnicas más comunes en este ámbito son la tomografía por emisión de positrones (PET, por sus siglas en inglés) y la resonancia magnética funcional (RMNf). [12]

Por otro lado, la neuroimagen estructural ofrece una visión estática del cerebro humano y aporta información acerca de su estructura anatómica. Este grupo abarca las técnicas de Tomografía Axial Computarizada (TAC) y Resonancia Magnética Nuclear (RMN). [12]

Para la realización de este trabajo se emplearán neuroimágenes estructurales de RMN. Se trata de una de las técnicas más utilizadas en la actualidad, gracias a la cual se obtienen imágenes tridimensionales con alta resolución espacial y evitando el uso de radiación ionizante. [12], [13]

La técnica de RMN emplea imanes para dar lugar a un campo magnético que obligue a los protones de nuestro cuerpo a alinearse con dicho campo. El valor la intensidad del campo magnético se mide en unidades de inducción magnética denominadas Tesla (T). En la práctica clínica, encontramos habitualmente equipos de RMN entre 0.02T y 3T. [14]

Cabe recordar que el protón se define como el catión hidrógeno (H^+) y que aproximadamente el 65% de nuestro cuerpo es agua. Gracias a ello, esta técnica es capaz de aprovechar la gran cantidad de protones de nuestro cuerpo, evitando así el uso de agentes externos para obtener la información.

Una vez alineados los protones con el campo magnético, se emplean pulsos de radiofrecuencia provocando que los protones estimulados giren fuera del equilibrio contra la fuerza del campo magnético. Al dejar de emitir dicho pulso, los protones vuelven a alinearse con el campo magnético inicial liberando energía que es captada por las antenas receptoras y que es el resultado del exceso de energía absorbido al entrar en resonancia (relajación). [13], [14]

Las diferencias tanto temporales como en la cantidad de energía liberada dependen del medio histoquímico en que se encuentre cada uno de los núcleos. Son justo estas propiedades magnéticas las que permiten diferenciar entre varios tejidos en la imagen de RMN. [13], [14]

Las imágenes de RMN se pueden, además, potenciar según diferentes parámetros (densidad, T1, T2 y T2*) dependiendo de la información que se desee obtener. En las imágenes potenciadas en densidad, la intensidad de cada píxel es directamente proporcional a la densidad de núcleos de hidrógeno que se encuentren en él. [14]

Antes de seguir con el resto de parámetros, se debe conocer que durante la relajación, la magnetización recupera sus dos componentes: la longitudinal (z) y la transversal (x,y), tal y como se puede observar en la figura 3.

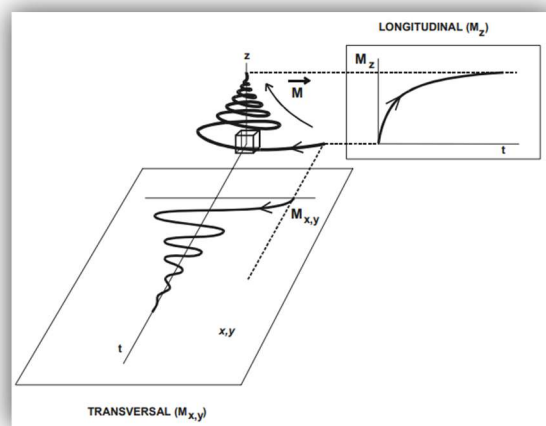


Figura 3. Estudio de la Relajación siguiendo la evolución en el tiempo de la componente longitudinal M_z y de la componente transversal $M_{x,y}$. [14]

Una vez sabido esto, las imágenes potenciadas en T1 permiten estudiar la facilidad de liberación energética dependiendo del entorno, a partir del tiempo de relajación longitudinal (expresado en ms). En estas imágenes, la intensidad de la señal es inversamente proporcional a la facilidad de relajación. [14] Es importante señalar que las imágenes utilizadas para la realización de este proyecto fueron imágenes de RMN estructural potenciadas en T1.

Las imágenes potenciadas en T2 y T2*, por su parte, consideran el tiempo de relajación longitudinal informando sobre el sincronismo durante la relajación de las estructuras presentes en un mismo vóxel. La intensidad de la señal, en este caso, es directamente proporcional a la coherencia en la relajación. La diferencia entre estos dos tipos de imágenes reside en el hecho de que las imágenes potenciadas en T2 no tiene en cuenta la influencia de las heterogeneidades del campo magnético ni las variaciones locales magnéticas que actúan sobre los núcleos, mientras que las imágenes potenciadas en T2* sí consideran todos estos factores. [14]

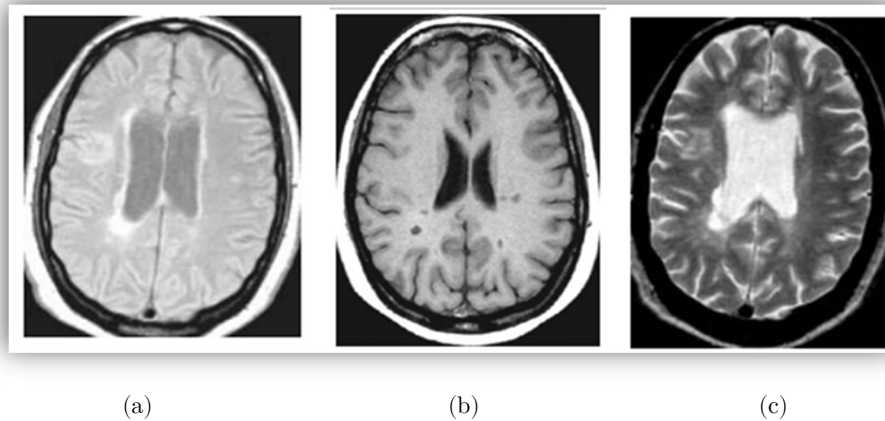


Figura 4. Imágenes de RMN potenciadas en: (a) densidad, (b) T1 y (c) T2.

1.3. Inteligencia artificial.

La inteligencia artificial (IA) es una de las revoluciones más importantes experimentadas por el mundo de la informática hasta el momento. Este término fue acuñado por primera vez en la conferencia de Dartmouth en 1956 y puede definirse como la técnica de software utilizada por los programas informáticos para resolver problemas o tomar decisiones de la misma forma que lo haría el ser humano. [15]

Dentro del campo de la IA diferenciamos varias áreas de investigación una de las cuales es el aprendizaje automático, cuyo objetivo principal es desarrollar técnicas que permitan a las máquinas aprender de forma automática. [15] Existen dos grandes métodos del aprendizaje automático: el aprendizaje supervisado, formado por pares datos-solución donde existe un conocimiento a priori; y el aprendizaje no supervisado, donde solamente se cuenta con datos de entrada con el propósito de obtener información a partir de la estructura de los datos. [16]

El aprendizaje profundo es un subcampo del aprendizaje automático, el cual pretende modelar, mediante el uso de redes neuronales, abstracciones de alto nivel a través de algoritmos más complejos y abstractos que los utilizados tradicionalmente en aprendizaje automático. [17]

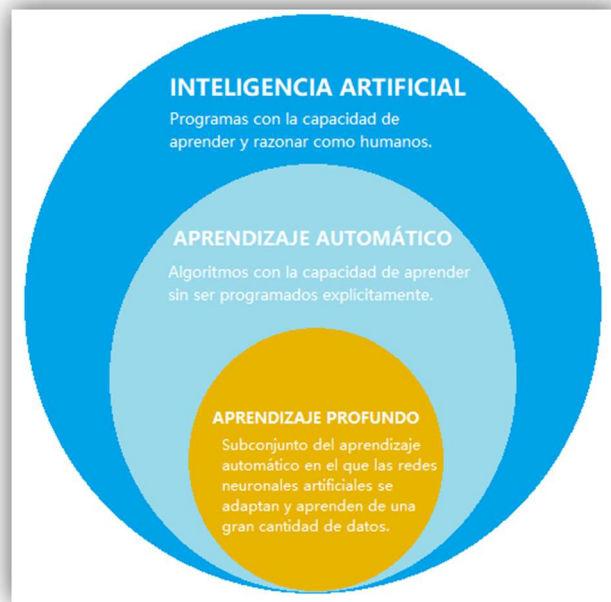


Figura 5. Diferencias entre inteligencia artificial, aprendizaje automático y aprendizaje profundo. [18]

En el ámbito médico es cada vez más común la aplicación de estos modelos para la clasificación de patrones con el propósito de ayudar a los profesionales de la salud en el tratamiento y diagnóstico del paciente. [19] Una de las enfermedades más investigadas en la actualidad y, por tanto, con mayor aplicación de estos algoritmos, es el Alzheimer.

1.3.1. Neurona artificial

Las neuronas artificiales pueden definirse como las unidades básicas de las redes neuronales artificiales (RNA). Su función consiste en, a partir de un vector de entrada, producir una salida única. Los elementos básicos de todo tipo de neuronas se pueden observar en la figura 4 y se explican brevemente a continuación. [20]

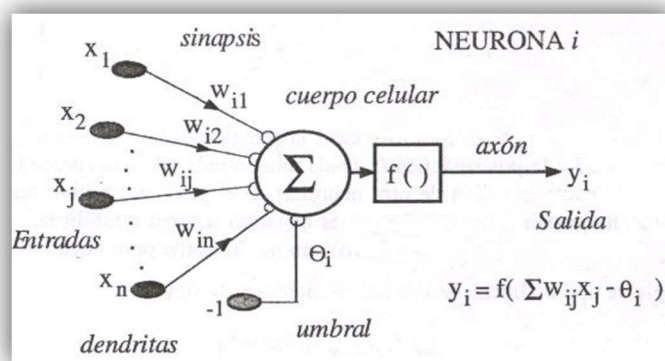


Figura 6. Esquema básico de la estructura de una neurona artificial. [20]

- ❖ Entradas (x). Datos de entrada a la neurona. [20]
- ❖ Pesos sinápticos (w). Se trata de valores numéricos que son modificados a través del entrenamiento de la red neuronal y que muestran un factor de importancia para cada una de las entradas. [20]

- ❖ Regla de la propagación. Se suele aplicar una función que relacione las entradas y los pesos previa realización del procesamiento, siendo ésta normalmente la suma ponderada. [20]
- ❖ Función de activación. El valor obtenido tras aplicar la regla de la propagación pasa a través de una última función llamada función de activación que nos proporcionará la salida de la neurona artificial. Las más comunes son la función lineal, sigmoïdal, limitante, escalón, tangente hiperbólica y gaussiana. [20]

Tratando de realizar una analogía humano-digital, existe una clara relación entre la neurona biológica y la neurona artificial (ver figuras 6 y 7). Los datos de entrada simularían el estímulo recibido por las neuronas biológicas mientras que los datos de salida reflejarían la respuesta a dicho estímulo. La sinapsis estaría relacionada con los datos de entrada y pesos sinápticos procedentes de otras neuronas artificiales. El soma abarcaría las funciones aplicadas a los datos de entrada. Finalmente, el axón sería el último camino hasta llegar a la salida, siendo ésta la entrada para la siguiente neurona artificial.

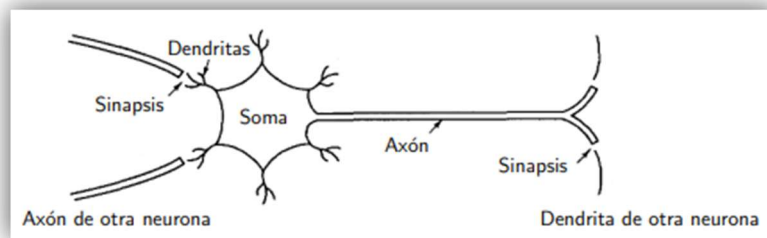


Figura 7. Esquema básico de la estructura de una neurona biológica. [21]

La función principal de ambos tipos de neuronas es la transmisión de información, ya sea en forma de impulsos nerviosos o en forma de modelos matemáticos. [21]

1.3.2. Red neuronal artificial

Las redes neuronales artificiales no son más que modelos matemáticos. En estas estructuras intervienen varias neuronas artificiales de modo que los datos de salida de una neurona serán los datos de entrada para la siguiente. De nuevo, se intenta modelar el comportamiento biológico de las conexiones neuronales que tienen lugar en el cerebro humano. [21]

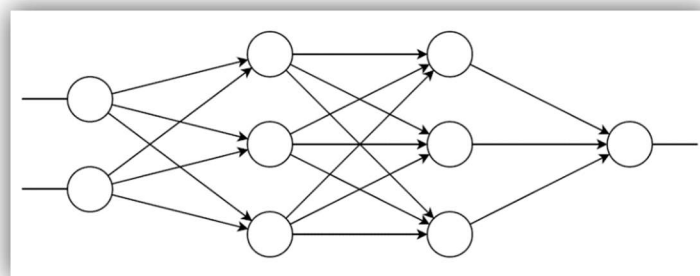


Figura 8. Esquema básico de una red neuronal artificial.

Actualmente, existe una clara necesidad de desarrollar sistemas capaces de ayudar en la detección precoz de la enfermedad para poder actuar y controlar la AD en el menor tiempo posible, como pueden ser las técnicas de aprendizaje profundo mediante redes neuronales tratados en este trabajo.

1.3.3. Red neuronal convolucional

Las redes neuronales convolucionales (CNN, por sus siglas en inglés) forman un determinado tipo de red neuronal artificial dentro del campo del aprendizaje supervisado, comportándose de forma similar a como actúa el córtex visual humano. Son algoritmos en los cuales las transformaciones se realizan utilizando la operación de convolución. [22], [23]

Están formadas por un gran número de capas que siguen una determinada jerarquía, de forma que las primeras capas detectan patrones o formas sencillas, como puede ser la detección de líneas o curvas, mientras que las capas finales o más profundas son capaces de reconocer formas más complejas como rostros. [22]

De forma resumida, su arquitectura básica está formada por una capa de entrada, varias capas ocultas y una capa de salida. La capa de entrada consiste en píxeles tomados de la imagen objeto de estudio. En las capas ocultas se puede encontrar capas convolucionales, capas de reducción o capas completamente conectadas, entre otras. Finalmente, la capa de salida es aquella que nos proporcionará la probabilidad de pertenecer a una u otra clase.

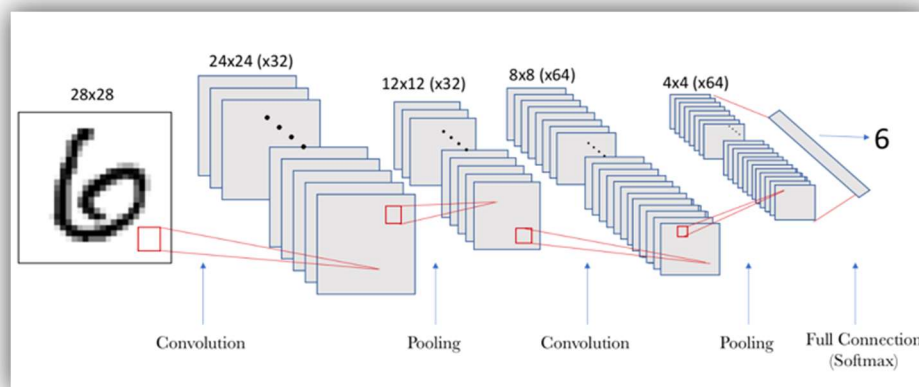


Figura 9. Estructura básica de una red neuronal convolucional. [24]

Las CNN están formadas por diferentes niveles. La arquitectura de cada uno de estos niveles suele ser semejante. Se trata de una primera capa de convolución encargada de la mayor parte de los cálculos realizados, seguida de una capa de normalización y una función de activación responsable de la extracción de características. A continuación, una capa de reducción reduce el número de características extraídas y, finalmente, una capa completamente conectada conduce al vector de salida con las probabilidades de cada píxel de pertenecer a una determinada clase. En los siguientes apartados se explicará más a fondo cada una de las capas anteriormente mencionadas.

A) Capa convolucional

Como se ha comentado en el párrafo anterior, se trata de la capa más importante de toda CNN. Estas capas aplican a los datos de entrada una serie de filtros, denominados *kernels*, mediante la operación de convolución. Estos *kernels* presentan un tamaño $n \times n$ si los datos de entrada son 2D o un tamaño $n \times n \times n$ si los datos de entrada son 3D. [22]

La operación de convolución consiste en tomar una matriz de píxeles de la imagen de entrada y realizar un producto escalar con respecto al *kernel* de forma reiterada hasta recorrer todos los píxeles de la imagen. El tamaño del grupo de píxeles tomados de la imagen de entrada y del *kernel* deben ser, por tanto, iguales. El *kernel* actúa desplazándose de izquierda a derecha y de arriba a abajo realizando así este producto escalar (ver figura 10). [22]

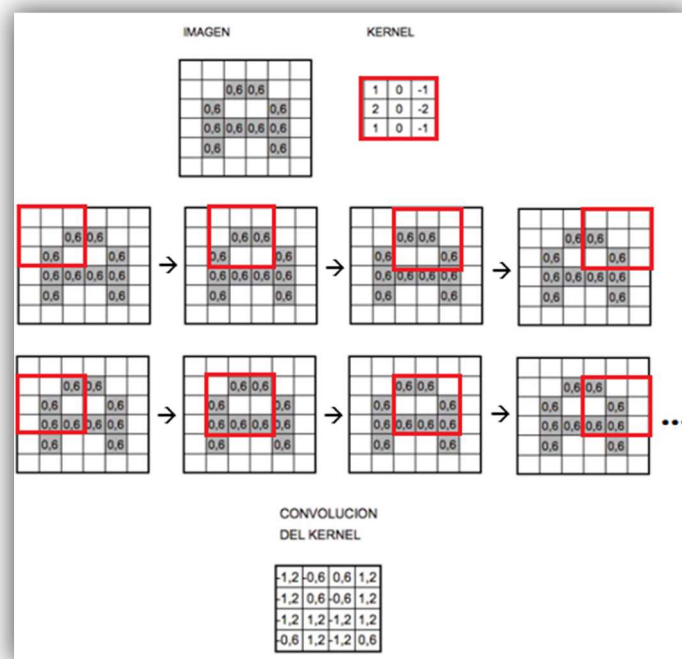


Figura 10. Operación de convolución en una imagen 2D. Camino seguido por el *kernel* sobre los píxeles de la imagen de entrada para realizar la convolución.

Como se puede observar en la figura 10, la matriz resultante de la convolución es de menores dimensiones que la matriz o imagen de entrada. Con el objetivo de preservar las dimensiones de la imagen de entrada se realiza la técnica denominada *zero padding* donde la matriz resultante de la convolución se rellena con ceros hasta conseguir las dimensiones de la imagen de partida (ver figura 11).

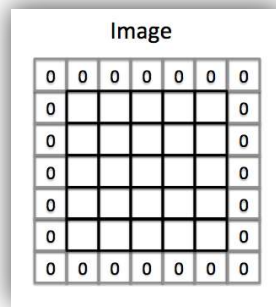


Figura 11. *Zero padding*.

Cada uno de los filtros aplicados nos dará como resultado un mapa de características distinto. Por esta razón, la profundidad del volumen de salida suele verse aumentado, siendo igual al número de filtros aplicados (ver figura 12).

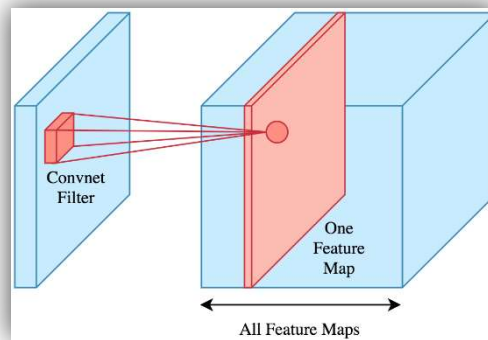


Figura 12. Creación del mapa de características. Cada filtro aplicado sobre la imagen nos aportará un nivel de profundidad sobre el mapa de características. [25]

B) Función de activación

Las funciones de activación pueden ser tanto lineales como no lineales y su objetivo principal es acotar los valores de salida dentro de un rango determinado.

Algunas de las funciones de activación utilizadas se muestran en la figura 14 pero, para este tipo de redes, la mayormente usada por excelencia es la función ReLU (ver figura 13).

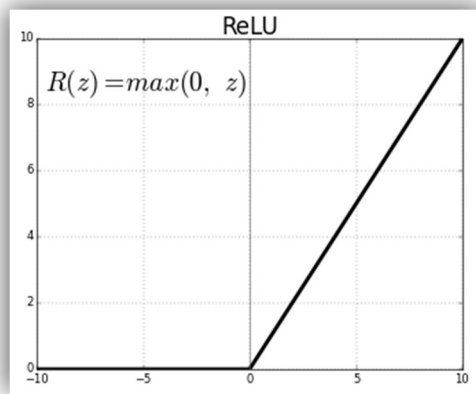


Figura 13. Función de activación ReLU.

	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal a tramos	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq +l \\ +1, & \text{si } x > +l \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0, +1]$	
Sinusoidal	$y = A \text{sen}(\omega x + \varphi)$	$[-1, +1]$	

Fuente: Hilera (1994).

Figura 14. Otras funciones de activación.[26]

C) Capa de reducción

Esta capa pretende reducir las dimensiones del mapa de características previamente creado, reduciendo la representación espacial y el espacio computacional. De forma práctica, permite eliminar características redundantes y mejorar la generalización del modelo. [22], [23]

Una de las formas de hacerlo es mediante la técnica de *max-pooling*, a partir de la cual se recorre mediante un determinado tamaño la matriz de características quedándonos con el valor mayor, intentando preservar así las características más relevantes (ver figura 15). Existen otras formas de reducción como puede ser preservar la media, en lugar del valor máximo. [22], [23]



Figura 15. Ejemplo de la técnica max-pooling.

D) Capa completamente conectada

La función de esta capa es simple, conectar todas las salidas de la capa anterior con todas las entradas de la capa posterior. La capa posterior suele estar formada habitualmente por una función *softmax* capaz de comprimir el vector de k-dimensiones saliente de la capa completamente conectada a un vector de la capa de salida final que tiene tantas neuronas como capas se estén clasificando.

E) Arquitecturas CNN

En este apartado se explicará brevemente la arquitectura de varias CNNs utilizadas en la actualidad. Una forma de medir la complejidad de estas redes es a través del número de parámetros, de forma que a mayor número de parámetros mayor nivel de complejidad. En la tabla 1 se puede observar de forma resumida la complejidad de cada una de las redes a través del número de parámetros.

Modelo	Parámetros totales	Parámetros entrenables	Parámetros no entrenables
SqueezeNet1_0	1,264,322	528,898	735,424
SqueezeNet1_1	1,251,394	528,898	722,496
AlexNet	2,735,426	265,730	2,469,696
DenseNet121	8,009,090	1,138,882	6,870,208
ResNet18	11,705,410	538,498	11,166,912
DenseNet169	14,197,634	1,871,554	12,326,080
VGG16_bn	15,252,034	537,346	14,714,688
DenseNet201	20,069,250	2,205,378	17,863,872
VGG19_bn	20,564,290	539,906	20,024,384
ResNet34	21,813,570	545,922	21,267,648

ResNet50	25,615,938	2,161,026	23,454,912
DenseNet161	28,744,386	2,492,322	26,252,064
ResNet101	44,608,066	2,213,250	42,394,816
ResNet152	60,251,714	2,259,330	57,992,384

Tabla 1. Número de parámetros presentes en cada una de las *CNN*. Ordenados de menor a mayor según el número de parámetros totales.

A la hora de decidir si es conveniente la aplicación de una red neuronal de mayor o menor complejidad se debe estudiar los resultados obtenidos. Una buena forma de comprenderlo es realizando una analogía con una situación común como puede ser el estudio para un examen de un determinado tipo problema matemático. El entrenamiento podría entenderse como los ejercicios realizados como parte del estudio, y la validación como el ejercicio realizado durante el examen. Dependiendo del nivel de profundidad y entendimiento en el estudio los resultados serán unos u otros.

Durante el estudio, puede que las soluciones de los problemas sean perfectas, obteniendo buenos resultados, pero que los resultados del examen sean peores de lo esperado. En este caso, el nivel de profundidad de estudio fue mínimo y solamente se memorizó la metodología de resolución del problema. Necesitaríamos, por tanto, emplear un mayor nivel de profundidad, una red más compleja.

En cambio, también puede ocurrir que el estudio sea demasiado profundo y complejo haciendo que se planteen dudas innecesarias para la resolución de los problemas, provocando malos resultados en la fase de entrenamiento. En cambio, este nivel de profundidad provoca mejores resultados en el examen. En este caso, la solución sería emplear un menor nivel de profundidad en el estudio, una red más sencilla.

En cualquier caso, la situación idílica sería aquella en la que los resultados obtenidos durante el estudio, entrenamiento, sean lo más similares posibles a los del examen, validación.

A continuación, se realizará un breve repaso por la arquitectura de cada uno de los modelos mencionados en la tabla 1.

AlexNet

AlexNet es la primera arquitectura de *CNN* que desempeña bien la clasificación de imágenes. Presenta una arquitectura simple pero potente conceptualizada en la década de 1980, formada por cinco capas convolucionales seguidas de tres capas completamente conectadas. Cabe destacar que la primera, segunda y quinta capa convolucional están seguidas por capas de *max pooling*. La arquitectura termina con un clasificador *softmax* (ver figura 16). [27]

En la figura 14 se puede observar entre las diferentes capas convolucionales una especie de pirámides que representarían los *kernels* utilizados en cada etapa de convolución. Tal y como se ha explicado previamente, el grosor de las capas o mapas de características está relacionado con el número de filtros aplicados.

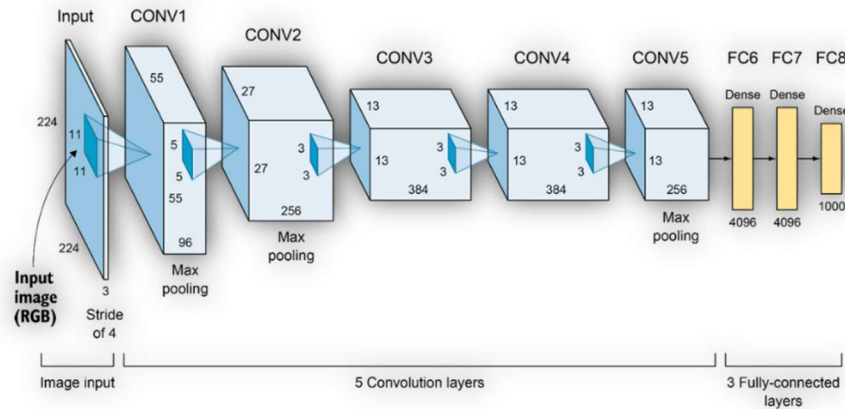


Figura 16. Arquitectura AlexNet. [28]

VGGNet

VGGNet fue desarrollada por el Grupo de geometría visual de la Universidad de Oxford en el año 2014. Los componentes de su arquitectura son exactamente los mismos que *AlexNet*, con la diferencia de que *VGGNet* es una red más profunda con capas más convolucionales, agrupadas y densas. [28]

VGGNet16 contiene dieciséis capas, trece convolucionales y tres completamente conectadas. La red *VGGNet19*, por su parte, está formada por dieciséis capas convolucionales y tres completamente conectadas. Las capas convolucionales están acompañadas por capas de *Max-pooling*. En última posición encontramos el clasificador *softmax*.

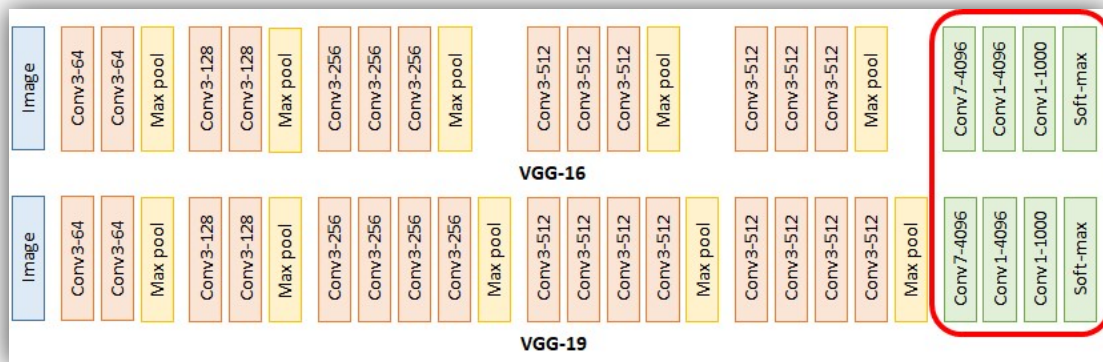


Figura 17. Arquitectura VGGNet. [29]

ResNet

Aparentemente, parece lógico pensar que a mayor profundidad de la red o lo que es lo mismo, mayor número de capas, los modelos presentan mayor flexibilidad para adaptarse y obtener mejores resultados al presentar un mayor número de parámetros. Sin embargo, esto no siempre es así ya que se puede perder la capacidad de generalización de la red. [30].

La red neuronal residual resuelve el ya conocido como problema de desvanecimiento del gradiente. Esta dificultad surge cuando existe un gran número de capas donde es probable

que el valor del gradiente en las primeras capas sea muy próximo a cero, lo cual impide a los pesos cambiar su valor. Esto conlleva a errores en el entrenamiento de las primeras capas que se arrastran a lo largo de toda la red y, en el peor de los casos, a impedir que la red continúe su entrenamiento.

Para solucionar dicho problema, esta red cuenta con una arquitectura que permite realizar atajos, realizando saltos de doble o triple capa, los cuales se muestran en la figura 18 en forma de líneas curvas.

Su arquitectura consiste en una capa de convolución y una capa de agrupación (figura 18, *block A*), seguidas de cuatro bloques convolucionales (figura 18, *block B, C, D y E*), y un clasificador *softmax* para finalizar. Según la cantidad de capas convolucionales, diferenciamos: *ResNet 18, ResNet 34, ResNet 50, ResNet 101 y ResNet 150*. [30]

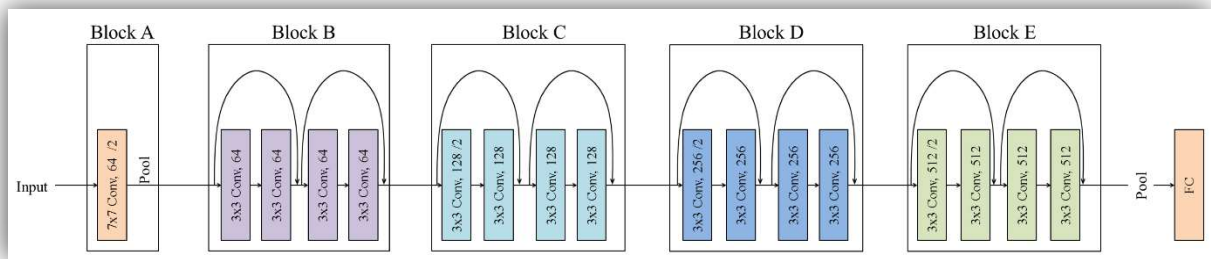


Figura 18. Arquitectura ResNet18. [31]

DenseNet

Esta red consta de dos bloques, un bloque denso y una capa de transición. El bloque denso consiste en una normalización por lotes, una activación ReLu y una convolución 3x3. La capa de transición, por su parte, está formada de una normalización por lotes, una convolución 1x1 y una agrupación promedio. [25]

La ventaja de la arquitectura de esta CNN es que cada capa obtiene la entrada de mapas de características de salida de capas anteriores. Esto conlleva a grandes ventajas como son el fortalecimiento de la propagación y reutilización de características, así como la reducción del número de parámetros. [32]

Es decir, mientras que las CNN tradicionales una sola conexión entre cada capa y su capa posterior (N capas, N conexiones), esta red tiene $N \cdot (N+1) / 2$ conexiones directas. De esta forma, cada capa tiene como entrada todas las características de las capas anteriores y sus propios mapas serán entradas de las capas posteriores (ver figura 19). [32] En la figura 17, se puede observar como a partir de 5 capas se crea un total de $5 \cdot (5+1) / 2 = 15$ conexiones directas.

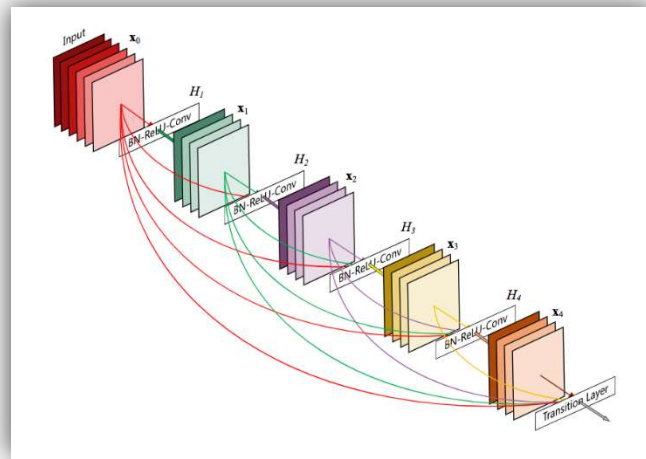


Figura 19. Arquitectura DenseNet. [33]

Dependiendo del número de capas diferenciamos entre *densenet121*, *densenet169*, *densenet201* y *densenet161*.

SqueezeNet

Para un mismo nivel de precisión, el uso de CNN con arquitecturas más pequeñas ofrece ventajas que se deben tener en cuenta, como son la menor comunicación entre servidores, el requerimiento de un menor ancho de banda y necesidad de menos capacidad en cuanto a memoria se refiere. [34]

Con el objetivo de aprovecharse de las ventajas anteriormente mencionadas surgió *SqueezeNet*. Esta nueva arquitectura consigue el mismo nivel de precisión que la *AlexNet*, anteriormente mencionada, con hasta 50 parámetros menos. [34]

La arquitectura de esta red empieza con una capa de convolución independiente, seguida de ocho módulos *Fire* formados por una capa de convolución de compresión alimentando una capa expandida compuesta por una combinación de convolución y filtros. El número de filtros por módulo aumenta de forma gradual a lo largo de la red. Finaliza con una capa de convolución final tal y como se puede observar en la siguiente figura. [34]

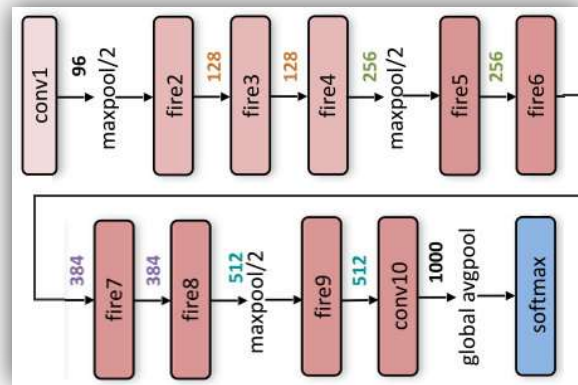


Figura 20. Arquitectura SqueezeNet. [35]

1.3.4. Entrenamiento de las CNN

A) Propagación direct

Como primer paso, los datos deben atravesar toda la red en la dirección de avance para poder generar una salida. Es decir, como situación inicial el valor de los pesos de los diferentes *kernels* serán aleatorios, o se aprovecharán los pesos de un entrenamiento previamente realizado para reducir el coste computacional, y los datos atravesarán toda la CNN hasta generar una salida. Las salidas o predicciones de la red serán entonces comparadas con las clasificaciones reales haciendo uso de la llamada función de pérdidas.

B) Función de pérdidas

Estas funciones tratan de mostrar el error realizado en la predicción a través de la comparación de estos valores con la clasificación verdadera de cada sujeto ya conocida. No existe una función de coste única para todo tipo de CNN, si no que su elección depende de varias características como puede ser la cantidad de valores atípicos o el algoritmo empleado. [36]

Estas funciones se clasifican en dos tipos: pérdidas de regresión, si se trata de la predicción de un conjunto de valores categóricos finitos; o pérdidas de clasificación, si por el contrario se trata de predicciones de valores continuos. [36]

Para el caso de pérdidas de regresión se puede hacer uso, como base, de errores como es el error cuadrático medio (ver ecuación 1) o el error medio absoluto (ver ecuación 2).

$$MSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n} \quad (1)$$

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n} \quad (2)$$

En el caso de pérdidas de clasificación es común el cálculo de pérdida de entropía cruzada (ver ecuación 3) o la pérdida de SVM de clases múltiples (ver ecuación 4).

$$SVMLoss = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1) \quad (3)$$

$$CrossEntropyLoss = -(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \quad (4)$$

La idea principal es que el valor de esta función de pérdidas será mayor durante las primeras iteraciones de entrenamiento e irá disminuyendo a medida que los pesos se vayan ajustando, lo cual implicará la obtención de mejores resultados.

C) Propagación inversa

La propagación inversa o retro-propagación hace uso de la técnica del gradiente descendiente justamente para tratar de reducir al máximo la función de pérdidas anteriormente explicada. Para ello, utiliza de la tasa de error obtenida en la iteración anterior al comparar las predicciones realizadas con las clasificaciones reales previamente conocidas.

El algoritmo de optimización del gradiente descendiente consiste en, para encontrar el mínimo de la función de pérdidas, situarse en un punto de la función y tratar de descender hasta el punto mínimo a través de primeras derivadas. Éste se aplica de forma iterativa.

Las redes neuronales utilizadas en aprendizaje profundo se entrenan utilizando el algoritmo del gradiente descendiente estocástico. En resumen, consiste en estimar el error y ajustar los pesos utilizando solamente un conjunto de ejemplos del conjunto de entrenamiento.

Un hiperparámetro importante y que se debe conocer es la tasa de aprendizaje, cuyo valor suele oscilar entre 0.0 y 1.0. Éste determina el tamaño de los saltos realizados para alcanzar el mínimo local de la función de pérdidas, controlando así la rapidez con la que el modelo conseguirá adaptarse al problema. Es importante tener en cuenta que valores pequeños pueden aumentar el tiempo en gran medida mientras que valores demasiado grandes pueden hacer que el mínimo local no llegue a alcanzarse.

En realidad, el objetivo es conocer la pérdida de cada unidad o nodo de la red neuronal artificial, para de esta forma conocer qué nodo es el responsable de la mayor parte de las pérdidas de una capa y darle así un peso menor. [37]

D) Tamaño de lote

Un lote se define como el conjunto de ejemplos, en este caso imágenes, que se emplean en una iteración del entrenamiento del modelo. El tamaño de lote será, por tanto, el número de imágenes que forman dicho lote.

E) Epochs

Una *epoch* es equivalente a lo que hasta el momento se ha definido como iteración. Es decir, se denomina así al del conjunto de datos completo a través de la red neuronal tanto en la dirección directa como en la inversa.

Congelación de las primeras capas de la CNN

Durante el entrenamiento se suele realizar, en primer lugar, un conjunto de *epochs* manteniendo las primeras capas de las redes neuronales artificiales congeladas. Es decir, de forma que los pesos de estas primeras capas no se modifiquen. De esta forma, se consigue disminuir tanto el coste temporal como computacional y de memoria.

En el caso de la librería empleada en este proyecto, las capas quedan inicialmente divididas en dos grupos, siendo la última capa de convolución la barrera entre dichos grupos. Con la congelación de las primeras capas se refiere a congelar el primero de los grupos y entrenar el segundo.

Posteriormente, y aprovechando la modificación de los pesos tras la congelación, se descongelan todas las capas y se sigue con el entrenamiento.

1.4. Estructura de la memoria.

La memoria de este proyecto se divide en siete capítulos. En el capítulo 2 se comentarán cuáles son los objetivos que se pretenden conseguir, así como cuál es la motivación principal que lleva a la realización de este trabajo. Seguidamente, en el capítulo 3 se realizará una breve revisión por el estado del arte.

A continuación, en el capítulo 4 se comentarán y explicarán las herramientas software empleadas, es decir, cuales han sido los lenguajes de programación y librerías empleadas. Además, también se abarcarán los métodos utilizados para la adquisición y pre-procesado de datos, así como las métricas empleadas para conseguir una buena evaluación de los resultados.

En el capítulo 5 se mostrarán los resultados obtenidos para cada uno de los experimentos realizados y en el capítulo 6 se realizará una breve discusión de los resultados y limitaciones encontradas. Finalmente, en el capítulo 7 se propondrán líneas futuras que permitan mejorar los resultados y se marcarán las conclusiones del proyecto.

CAPÍTULO 2. OBJETIVOS Y MOTIVACIÓN.

2.1. Objetivos

Los objetivos de este trabajo son los siguientes.

- ❖ Estudiar y conocer las técnicas de aprendizaje profundo y pre-procesado de imagen utilizadas en resonancia magnética cerebral estructural.
- ❖ Implementar dichas técnicas para conseguir una clasificación automática entre sujetos sanos y patológicos en cuanto a la enfermedad de Alzheimer se refiere.
- ❖ Comprobar la utilidad del campo de la inteligencia artificial en el diagnóstico de la Enfermedad de Alzheimer.

2.2. Motivación

Tras el análisis de las predicciones de envejecimiento de la población, la epidemiología de la AD y el inmenso potencial del campo de la IA en el ámbito médico se ha detectado la necesidad de desarrollar herramientas eficaces que ayuden a la detección precoz de esta enfermedad.

La motivación principal de este proyecto es, por tanto, la intención de ayudar a los pacientes de AD así como a los profesionales de la sanidad para que se consiga actuar en el menor tiempo posible. Como consecuencia de ello, se logra apoyar al sistema sanitario ya que se desconoce si el sistema de salud tal y como se conoce hoy en día está preparado para afrontar las situaciones futuras.

Se debe tener en cuenta que la incidencia de la enfermedad de Alzheimer ha ido en aumento año tras año (ver figura 21). Esta enfermedad repercute, actualmente y en nuestro país, en la vida diaria de 3,5 millones de personas teniendo en cuenta familiares y cuidadores. [38]

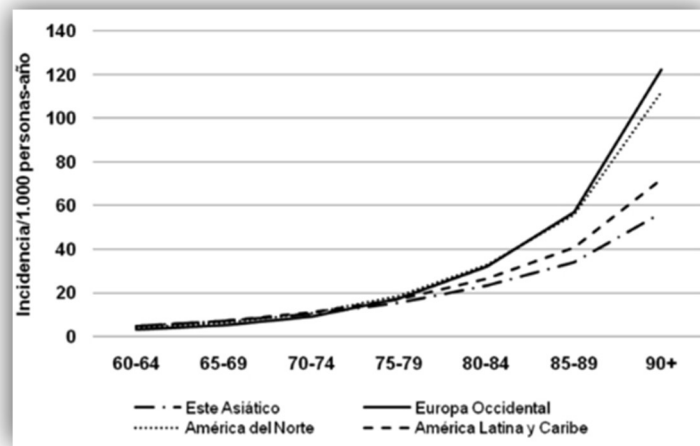


Figura 21. Estimación de la incidencia de demencia en mayores de 60 años. [39]

Además, se espera que en 2050 las personas mayores de 65 años representen el 30% del total de la población española, siendo la edad el principal factor de riesgo en la AD. [40]

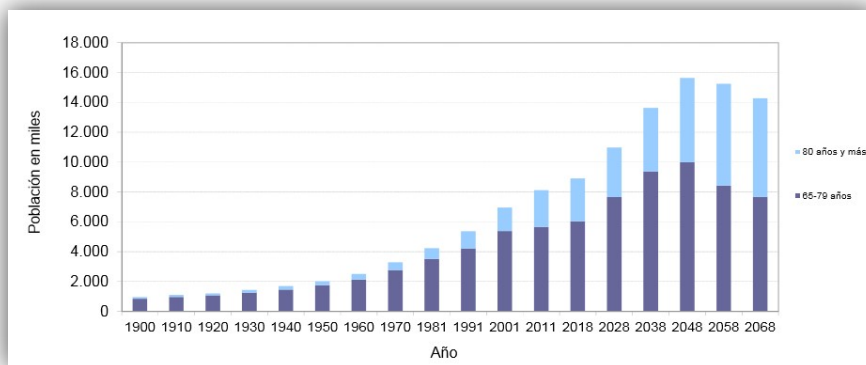


Figura 22. Envejecimiento de la población española estimada para los próximos años según el Instituto Nacional de Estadística. [41]

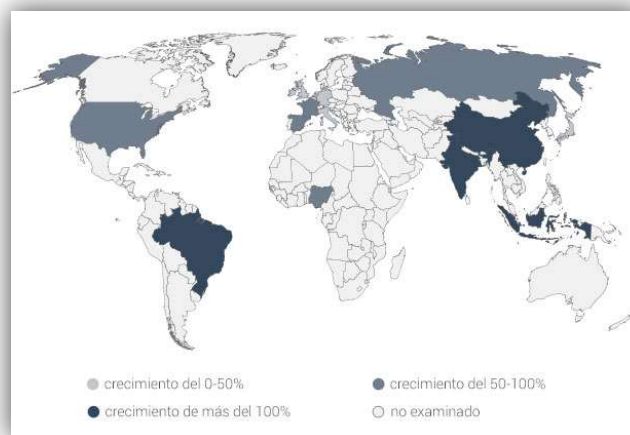


Figura 23. Alzheimer en los próximos 20 años. De 27 millones a 60 millones de enfermos a nivel mundial. [42]

CAPÍTULO 3. ESTADO DEL ARTE.

Es un hecho que la inteligencia artificial, y en concreto el aprendizaje profundo, ha adquirido un gran peso en el campo de la medicina en los últimos años. Su gran repercusión está dirigida sobre todo hacia la medicina predictiva, un campo cuyo objetivo principal es predecir la probabilidad de sufrir una determinada enfermedad. Como veníamos diciendo en el capítulo anterior, también es cierto que la enfermedad de Alzheimer es una de las patologías más investigadas en la actualidad y, consecuentemente, de aplicación de dicha tecnología.

Tras una revisión de los trabajos previamente realizados, se ha podido comprobar que la mayor parte de ellos utilizaron redes neuronales convolucionales tridimensionales (3D), capaces de procesar directamente los volúmenes resultado de la RMN.

En cuanto a bases de datos se refiere la mayormente utilizada es ADNI (*The Alzheimer's Disease Neuroimaging Initiative*). En esta base de datos se pueden encontrar datos e imágenes de RMN estructural de 2.562 sujetos, clasificados como cognitivos normales (CN), con deterioro cognitivo leve (MCI, por sus siglas en inglés) o diagnosticados como enfermos de Alzheimer (AD).

Las técnicas de pre-procesado más comunes son, por su parte, la extracción de cráneo [43]–[45], la normalización de intensidades [43]–[45] y la aplicación de filtros de reducción de ruido [43], [45], [46].

Sin embargo, no necesariamente todos los clasificadores desarrollados hasta el momento están dirigidos única y exclusivamente a imágenes como datos de entrada. Por ejemplo, si nos remontamos al año 2006, se pueden encontrar clasificadores de normalidad/anormalidad basados en la actividad eléctrica cerebral. Concretamente, las señales de entrada utilizadas fueron las transformadas discretas de *wavelet*. Cabe destacar que, pese a que el conjunto de datos utilizados fue de muy pequeño tamaño (imágenes MR de 52 pacientes) los resultados obtenidos fueron muy buenos llegando a conseguir una exactitud del 98%. [46] Cabe destacar que la exactitud, se define como el porcentaje de aciertos con respecto al total de observaciones de las que se dispone. El resultado obtenido en el proyecto anteriormente mencionado es extraordinario, siendo éste ya aplicable en la práctica clínica.

Uno de los estudios a destacar en el uso de imágenes potenciadas en T1 de RMN cerebral como datos de entrada a la CNN es [44]. El objetivo fue demostrar la obtención de buenos resultados sin la necesidad de pasos de extracción de características haciendo uso de las CNN ResNet y VoxVGG. Para ello se realizaron distintas clasificaciones binarias teniendo en cuenta las cuatro etiquetas encontradas en los datos de entrada. Se obtuvieron resultados muy similares con ambas arquitecturas, siendo la exactitud más alta del 80%, para el caso de clasificación entre sujetos cognitivos normales y con la EA, con un valor del área bajo la curva ROC igual a 88. [44]

Otra prueba del gran uso de las 3D CNN en el campo de la EA es el artículo [43]. En este caso la red neuronal estaba construida sobre un *autoencoder* 3D pre entrenado para la

captación de estructuras anatómicas. Este tipo de redes neuronales artificiales están basadas en el aprendizaje de codificaciones de datos de forma no supervisada. Se dispuso de un total de 210 sujetos: 70 CN, 70 MCI y 70 AD. De nuevo el mayor nivel de exactitud se obtuvo para la diferenciación de los grupos CN y AD, siendo del 97.6%. En segundo lugar, el mejor resultado fue para la clasificación de MCI y AD con un nivel de exactitud del 95%. El peor resultado se obtuvo para la clasificación entre los tres grupos o etiquetas.

Como se ha podido comprobar, la mayor parte de los proyectos de investigación están dirigidos a redes neuronales convolucionales tridimensionales, desarrolladas fundamentalmente sobre la estructura de la librería *Keras* debido a la alta documentación que le acompaña.

Únicamente se ha encontrado un artículo dedicado a CNN 2D, [45]. Como técnicas de pre-procesado se realizó una extracción de cráneo, corrección de inhomogeneidades mediante normalización N3 y una normalización espacial. Se hizo uso de un total de 389 imágenes *.png*, 19 de cada sujeto resultado de la selección de un corte cada cinco desde el corte axial 10 hasta el 100. Los mejores resultados se obtuvieron para el corte axial 40 consiguiendo un 73% de exactitud.

Se ha de enfatizar que los conjuntos de datos utilizados en los estudios influyen en gran medida en los resultados obtenidos. Este hecho junto con la variabilidad de formas de entrada de dichos datos a las CNN (técnicas de pre-procesado utilizadas) hace que sea un gran reto realizar comparaciones entre los resultados obtenidos en los distintos estudios.

CAPÍTULO 4. MATERIALES Y MÉTODOS.

4.1. Adquisición de los datos

Los datos utilizados para la realización de este trabajo fueron obtenidos de la base de datos *ADNI*. Esta iniciativa surge a partir del consorcio de varias universidades y centros médicos en los Estados Unidos y Canadá que trabajan conjuntamente reuniendo información que les permita desarrollar un repositorio de datos accesible, con el único fin de desarrollar técnicas estandarizadas de imagen y procedimientos de biomarcadores en sujetos sanos, sujetos con deterioro cognitivo leve y sujetos con enfermedad de Alzheimer. [47]

Concretamente se trabajó con un subconjunto de 199 imágenes en formato NIfTI de resonancia magnética cerebral potenciadas en T1 adquiridas con un equipo de 3 Teslas. En cuanto al diagnóstico de la enfermedad, de estos 199 sujetos 60 eran CN, 97 presentaban MCI y 42 sujetos padecían AD (ver figura 24).

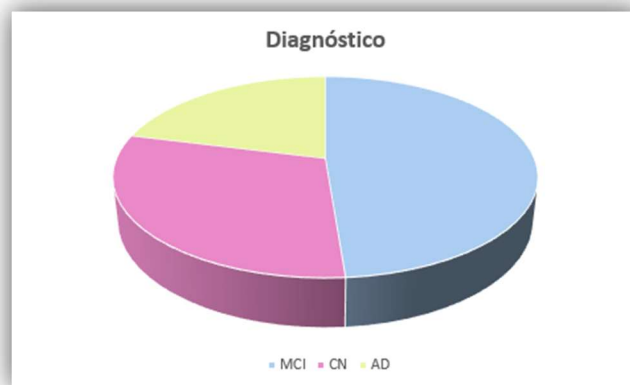


Figura 24. Clasificación a priori de los 199 sujetos iniciales según la base de datos ADNI.

Además, de estos 199 sujetos, 105 resultaron ser mujeres (F) mientras que 94 eran hombres (M).

Tras revisar los datos se observó que 49 sujetos presentaban dos imágenes asociadas, correspondientes a dos instantes de tiempo o puntos de control distintos. Estas imágenes no presentaban diferencias significativas y, por tanto, se decidió eliminar las imágenes asociadas al segundo punto de control. Esta decisión evita entrenar la red neuronal con información que pueda interpretarse como repetida y que no aporte ninguna información extra que pueda serle útil para generalizar el problema a cualquier sujeto.

Con ello, el resultado total fue de 78 mujeres y 72 hombres, siendo un total de 150 sujetos (ver figura 25). Las edades se encontraron en el rango entre 55 y 89 años, siendo la mayoría de ellos sujetos de entre 70 y 80 años (ver figura 26). Se ha de recordar que la enfermedad de Alzheimer se da sobre todo en sujetos de edad avanzada, mayores de 65 años.

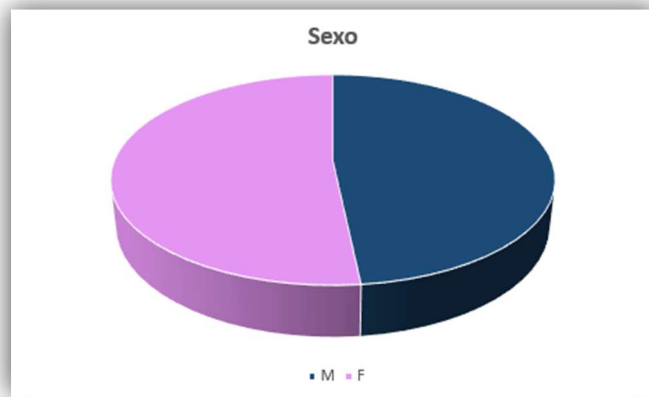


Figura 25. Sexo de los 150 sujetos finales según la base de datos ADNI.

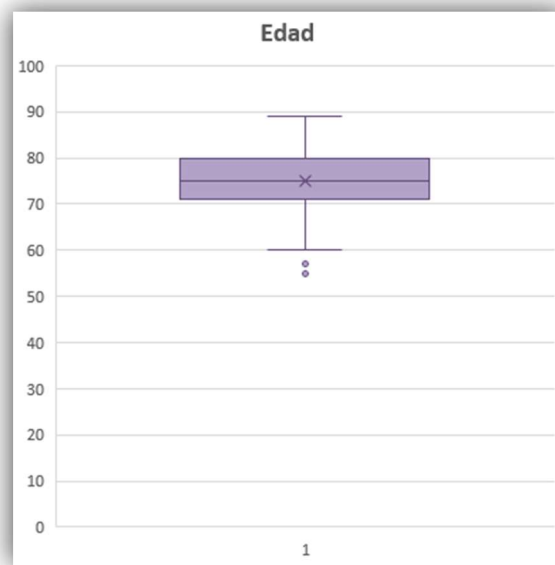


Figura 26. Edad de los 150 sujetos finales según la base de datos ADNI.

En dos de los experimentos se consideraron los 3 grupos de diagnóstico (CN, MCI, AD) teniendo en cuenta los 150 sujetos anteriormente mencionados.

Etiquetas	Sujetos
CN	46
MCI	71
AD	33
Total	150

Tabla 2. Clasificación terciaria. Número de sujetos asociados a cada etiqueta.

Pese a ello, también se realizó una clasificación binaria *Normal/Anormal*. Por esta razón, se establecieron paralelamente las nuevas etiquetas con las que se entrenarían las CNN. Estas dos etiquetas se denominaron *Normal* y *Anormal*. La primera de ellas recogería aquellos

pacientes sanos (CN) mientras que la segunda abarcaría tanto sujetos con MCI como enfermos de AD.

En la siguiente tabla y gráfico se muestran los sujetos finales que forman cada una de las etiquetas.

Etiquetas	Sujetos
Normal	46
Anormal	104
Total	150

Tabla 3. Clasificación binaria. Número de sujetos asociados a cada etiqueta.

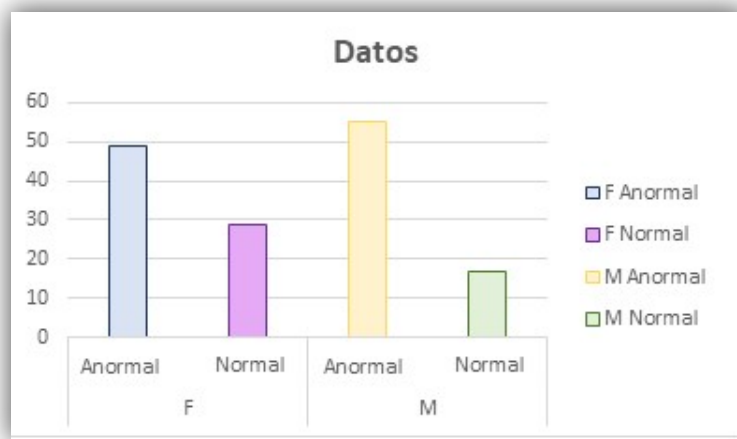


Figura 27. Gráfico de barras: número de sujetos asociados a cada etiqueta según el sexo.

Una vez seleccionadas las imágenes 3D y los datos con los que se trabajarían, se procedió a realizar los distintos experimentos para conseguir el objetivo establecido.

4.2. Herramientas software

4.2.1. MATLAB

MATLAB es un lenguaje de programación de alto nivel soportado por los tres sistemas operativos principales (*Mac*, *Windows* y *Linux*) que destaca por la realización de tareas de cálculo complejas de una forma rápida y eficaz. Está basado en operaciones vectoriales y matriciales, las cuales son fundamentales en el mundo de la ingeniería y ciencia. Sin embargo, su principal uso es en el entorno académico debido, en primer lugar, a que se trata de un software privado que exige una licencia de pago para su uso y, en segundo lugar, a su interfaz sencilla e intuitiva. [48]

La importancia de este lenguaje en la realización de este trabajo recae en el pre-procesado de imágenes. Concretamente, el uso de funciones relacionadas con el procesamiento de imágenes. Éstas permiten visualizar imágenes tridimensionales así como el desarrollo de

algoritmos cuyo objetivo sea mejorar la calidad, segmentar, registrar, realizar transformaciones geométricas o incluso procesar imágenes 3D. [49]

4.2.2. SPM (Statistical Parametric Mapping)

En un primer momento se pensó en utilizar la biblioteca de software *FMRIB v6.0* (conocida comúnmente como *FSL*) escrita en *Python* para el tratamiento de las imágenes de resonancia magnética cerebrales. Sin embargo, dicha librería solo es ejecutable en sistemas basados en UNIX y, desafortunadamente, solamente se contaba con un equipo *Windows* para la realización de este trabajo. Esto dificultaba bastante el uso de la librería al ser necesario ejecutarla a través de una máquina o contenedor virtual.

Por todo ello, se decidió utilizar el software gratuito y de código abierto *SPM* (*Statistical Parametric Mapping*) pese a que ello conllevara utilizar *MATLAB* como lenguaje de programación durante el pre-procesado de las imágenes. Concretamente, se utilizó la última versión *SPM12*.

Se trata de uno de los softwares más utilizados en el mundo para el análisis de imágenes médicas, concretamente neuroimágenes, tanto para su preprocesamiento como para su visualización. SPM como técnica estadística fue desarrollada en el ámbito de la neuroimagen a mediados de la década de los 90 por el investigador Karl Friston. Sin embargo, actualmente también utilizamos dichas siglas para referirnos al software creado por el Departamento de Neurociencia de Imágenes de Wellcome (*University College of London*), cuyo objetivo es analizar volúmenes de imagen cerebral. [50]

4.2.3. Python

Si bien es cierto que existe una gran variedad de herramientas software con las que trabajar, para este proyecto se decidió utilizar *Python 3*, con ayuda del entorno de desarrollo integrado *PyCharm*, tanto para el tratamiento de los datos como para realizar los experimentos hasta conseguir el modelo de clasificación final.

La razón principal por la que se optó por *Python* como lenguaje de programación principal es que es considerado la mejor opción tanto por científicos como por ingenieros que buscan crear aplicaciones científicas a partir de un lenguaje de alto nivel. [51]

Es uno de los lenguajes de programación más utilizados en la actualidad, ocupando la primera posición en diferentes rankings de lenguajes de desarrollo de inteligencia artificial. Esto es debido principalmente a su sintaxis sencilla y fácil de entender, así como a su corto tiempo de desarrollo en comparación con otros lenguajes. [52], [53]

Además, presenta otras ventajas como pueden ser su licencia de código abierto (venta, uso y distribución gratuita) y el hecho de ser multiplataforma y multiparadigma (permite programación orientada a objetos, estructurada y funcional). También cuenta con un gran número de librerías, es dinámico (no es necesario definir el tipo de variable antes de su uso) e interpretado, es capaz de manejar la memoria de forma automática (*Python* reserva la memoria

antes de utilizarla) y existen gran cantidad de foros dedicados a resolver dudas y problemas referentes a dicho lenguaje. [51]

4.2.4. Fastai

Concretamente, se trabajó con la librería de código abierto *Fastai* de *Python* para aprendizaje profundo. Actualmente, ya se utiliza tanto en investigación e industria como en enseñanza y es capaz de simplificar en gran medida el entrenamiento de redes neuronales reduciéndolo a muy pocas líneas de código en comparación con otras librerías como puede ser *Keras*. [54], [55], [56]

Está construida sobre *PyTorch* (biblioteca de aprendizaje automático de código abierto) y contiene varios modelos pre entrenados de *torchvision*, un paquete que incluye datos populares, arquitecturas de modelos y transformaciones comunes de imágenes. Entre estos modelos encontramos las siguientes arquitecturas neuronales: *ResNet*, *SqueezeNet*, *DenseNet*, *VGGNet* y *AlexNet*.

Éstas han sido previamente entrenadas con un conjunto de datos y contienen los pesos y sesgos que representan las características de dichos datos. La ventaja de utilizar este tipo de modelos es que las características son comúnmente transferibles a datos distintos, ahorrándonos tanto tiempo como recursos computacionales. [45]

4.3. Métricas

Para evaluar la calidad del clasificador se decidió emplear ocho métricas explicadas a continuación: matriz de confusión, exactitud, precisión, sensibilidad, especificidad, *F1 score*, curva ROC y área bajo la curva ROC.

4.3.1. Matriz de confusión

Como su propio nombre indica, el resultado es una matriz cuadrada con tantas filas y columnas como clases. Las filas indican los valores reales mientras que las columnas indican los valores predichos. Cada entrada se incrementará en una unidad cuando la combinación valor real X y valor predicho Y se satisfaga. De esta forma, permite conocer de una forma visual si el sistema presenta confusiones entre clases (ver figura 28).

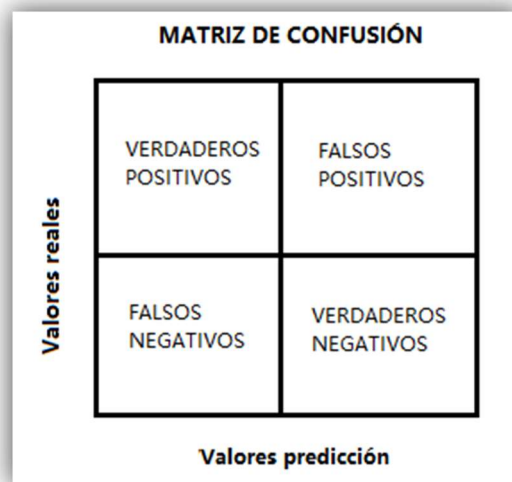


Figura 28. Estructura de una matriz de confusión.

A partir de los datos de dicha matriz se pueden extraer determinados términos relevantes para el análisis de los resultados: verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. Por ejemplo, en el caso concreto del clasificador binario, los verdaderos positivos serán aquellas imágenes clasificadas como patológicas cuando realmente lo son. Verdaderos negativos, se referirá a aquellas muestras clasificadas como no patológicas cuando realmente lo son. En tercer lugar, los falsos positivos se referirán a las observaciones clasificadas como patológicas cuando realmente son no patológicas. Por último, los falsos negativos serán aquellas imágenes clasificadas como no patológicas cuando realmente son patológicas.

Término	Valor predicho	Valor real
TP (verdadero positivo)	Patológica	Patológica
TN (verdadero negativo)	No patológica	No patológica
FP (falso positivo)	Patológica	No patológica
FN (falso negativo)	No patológica	Patológica

Tabla 4. Resumen para el caso concreto del clasificador binario: falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos.

4.3.2. Exactitud

La exactitud se define como la relación entre el número de predicciones correctas y el número total de muestras de entrada (ver ecuación 5).

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

Esta métrica presenta algunas limitaciones, como puede ser la dependencia del número de observaciones pertenecientes a cada grupo. Es decir, imaginemos que la clase A forma el 98% de las observaciones. Si el clasificador es capaz de acertar las muestras correspondientes

a dicha clase conseguiría una precisión del 0.98 pese a no conseguir ningún acierto en el caso de la clase B. El problema se agrava cuando el coste de una clasificación errónea en la clase minoritaria es muy alto, como puede ser el caso de una enfermedad mortal. [60]

4.3.3. Precisión

La precisión se define como la razón entre los verdaderos positivos y el total de valores reales positivos (ver ecuación 6).

$$\text{Precisión} = \frac{TP}{TP + FP} \quad (6)$$

4.3.4. Sensibilidad

Esta métrica también es conocida como la tasa de verdaderos positivos. Está definida como el total de verdaderos positivos entre la suma de verdaderos positivos y falsos negativos (ver ecuación 7).

De forma práctica, nos informa acerca de la capacidad de detectar correctamente muestras patológicas entre el total de muestras patológicas.

$$\text{Sensibilidad} = \frac{TP}{TP + FN} \quad (7)$$

4.3.5. Especificidad

La especificidad muestra la capacidad del clasificador de detectar muestras no patológicas entre el total de muestras no patológicas. Matemáticamente, se define como el cociente entre verdaderos positivos y la suma de verdaderos positivos y falsos negativos.

$$\text{Especificidad} = \frac{TN}{TN + FP} \quad (8)$$

4.3.6. F1 score

Esta métrica es capaz de mostrar la información de la precisión y sensibilidad a la vez. Es útil cuando el objetivo es conseguir un equilibrio entre la precisión y la sensibilidad. Su cálculo se puede observar en la ecuación 9.

$$F1 \text{ score} = 2 \cdot \frac{\text{precisión} * \text{sensibilidad}}{\text{precisión} + \text{sensibilidad}} \quad (9)$$

4.3.7. Curva ROC

La curva de característica operativa del receptor (curva ROC) es un gráfico que es capaz de representar dos parámetros: la tasa de verdaderos positivos (ver ecuación 7) y la tasa de falsos positivos (ver ecuación 10).

$$TFP = \frac{FP}{FP + VN} \quad (10)$$

Siendo más específicos, representa tasa de verdaderos positivos frente a tasa de falsos positivos en diferentes umbrales (ver figura 29).

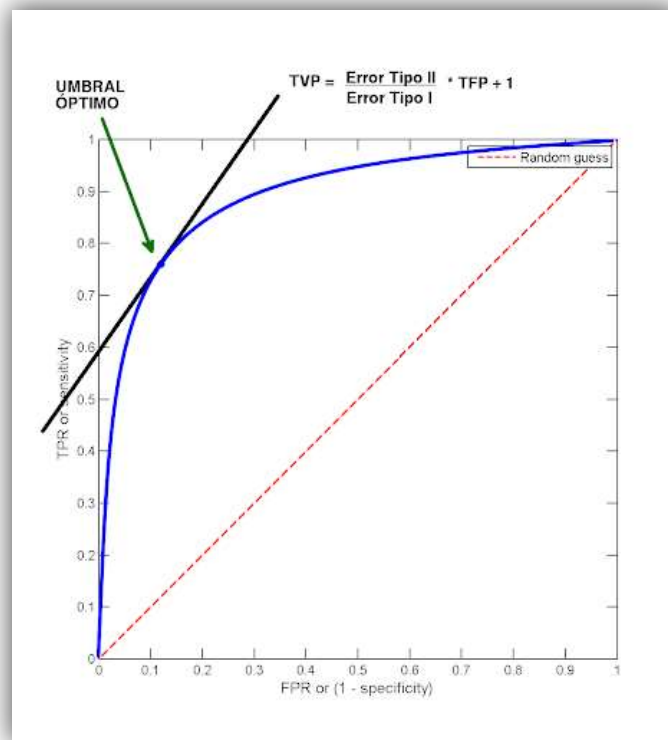


Figura 29. Ejemplo de curva ROC. [61]

4.3.8. Área bajo la curva ROC

El área bajo la curva ROC (AUC) es una de las métricas más utilizadas sobre todo en casos de clasificación binaria. Se podría definir como la probabilidad de que el clasificador sea capaz de clasificar un ejemplo positivo elegido al azar con una probabilidad mayor que un ejemplo negativo escogido al azar. Este valor oscila entre 0 y 1 siendo mejor el rendimiento de nuestro modelo cuanto más cercano a 1 sea el resultado obtenido.

Para el cálculo de esta métrica se mide toda el área bidimensional bajo la curva ROC anteriormente explicada.

4.4. Pre-procesamiento

4.4.1. Creación de las imágenes 2D

El trabajo se realizó sobre redes neuronales artificiales 2D. Por este motivo, y también como parte del pre-procesado, se transformó a través del lenguaje de programación *Python* cada uno de los volúmenes 3D NIfTI en imágenes *.png* (2D) correspondientes a cortes axiales de dichos volúmenes.

Los volúmenes 3D presentaron como primera dimensión un valor fijo de 256. En cambio, la segunda y tercera dimensión fueron dependientes de la adquisición (ver figura 30). Las segundas dimensiones presentaron uno de los siguientes valores: 160, 166 o 170; mientras que la tercera dimensión fue 240 o 256.

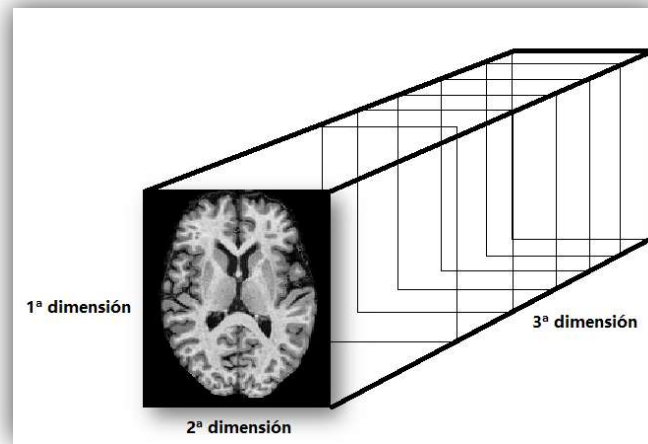


Figura 30. Dimensiones de los volúmenes 3D. La tercera dimensión se corresponde con el número de cortes, y la primera y segunda dimensión con el tamaño de la imagen *.png*.

Todas las imágenes de un mismo sujeto contienen la etiqueta asociada al volumen tridimensional de éste. Es decir, si un volumen 3D (256x170x256) correspondiente al sujeto 1 es diagnosticado como *Normal*, el resultado será 256 imágenes *.png* etiquetadas como *Normales*. Con ello, se obtuvo un total de 37.168 imágenes *.png* (ver tablas 5 y 6).

Etiqueta	Número de imágenes
CN	11.488
MCI	17.776
AD	7.904
TOTAL	37.168

Tabla 5. Clasificación multiclase. Número de imágenes *.png* asociadas a cada etiqueta.

Etiqueta	Número de imágenes
Normal	11.488
Anormal	25.680
TOTAL	37.168

Tabla 6. Clasificación binaria. Número de imágenes .png asociadas a cada etiqueta.

4.4.2. Zero padding

Teniendo en cuenta la diversidad de tamaño de las imágenes, se realizó un redimensionado a 256x256 píxeles para conseguir así un tamaño de imagen cuadrada con el que empezar a trabajar. Con el objetivo de no alterar las estructuras anatómicas, ya que puede ser de interés para el diagnóstico de la enfermedad, se optó por realizar un relleno con ceros (*zero padding*) en todas las imágenes .png. El resultado se muestra en la siguiente figura.

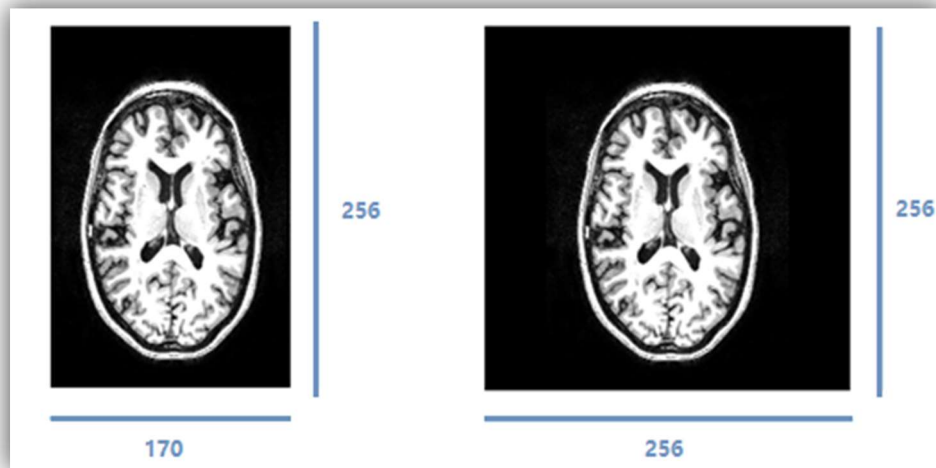


Figura 31. Resultado del relleno con ceros.

4.4.3. Extracción de cráneo

A) Segmentación

Se realizó, en primer lugar, una segmentación para extraer aquellas estructuras de la imagen de resonancia magnética cerebral potenciada en T1 que son de interés para el estudio de la enfermedad de Alzheimer. Para ello, se utilizó la herramienta SPM comentada en apartados anteriores. Concretamente, se hizo uso de la opción *Segmentation*.

Con ella, se realizó la segmentación de los tejidos de interés. Para cada una de las imágenes cerebrales se extrajeron los tres tejidos principales: sustancia gris (tejido 1), sustancia blanca (tejido 2) y líquido cefalorraquídeo (tejido 3), preservando las opciones predefinidas por la herramienta. Los tejidos se extrajeron conservando su posición en el espacio nativo original para cada uno de los sujetos.

De forma resumida, las opciones predefinidas más importantes se explican a continuación.

- ❖ *Bias regularisation* → *light regularisation*. Las imágenes de RMN suelen estar corrompidas por un artefacto que afecta uniformemente a toda la imagen produciendo un sesgo en el valor de intensidad. Este hecho no es observable visualmente pero sí da lugar a problemas durante el procesamiento de las imágenes. [50]
- ❖ *Bias FWHM*. → *60 mm* Anchura a media altura del *kernel* de suavizado gaussiano. [50]
- ❖ *MRF Parameter* → *1*. Controla la fuerza del campo aleatorio de Markov. [50]
- ❖ *Clean up* → *light clean*. Intenta perfeccionar la extracción del cerebro de imágenes segmentadas. Para ello, erosiona la sustancia blanca para deshacerse así de vóxeles extraños. A continuación, realiza dilataciones condicionales basándose en la presencia de sustancia gris y blanca. Así consigue eliminar tejidos como el humor acuoso y vítreo u otras regiones fluidas. [50]
- ❖ *Warping Regularisation* → *1x5 double*. A mayor regularización conseguimos deformaciones de mayor suavidad. [50]
- ❖ *Smoothness* → *0*. El suavizado espacial pretende compensar la variabilidad anatómica funcional que no se ha conseguido con la normalización espacial. Generalmente se utiliza el valor 0 en el caso de imágenes de RMN. [50]
- ❖ *Sampling distance* → *3*. Determina la distancia entre puntos muestreados, es decir, el tamaño de cada uno de los vóxeles. [50]

A continuación, se muestra el resultado obtenido para uno de los sujetos observando el plano axial del volumen 3D.

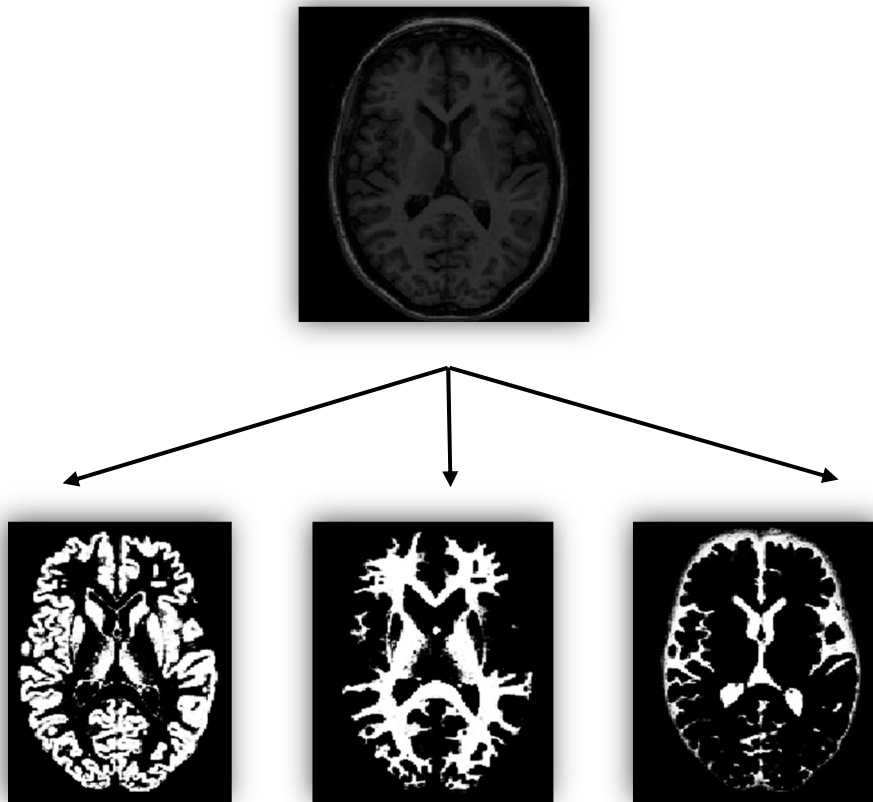


Figura 32. Resultado de la segmentación en tejidos: izquierda, sustancia gris; centro, sustancia blanca; derecha: líquido cefalorraquídeo.

Como se puede observar, el resultado fue una máscara probabilística para cada uno de los tejidos a segmentar.

B) Extracción de cráneo

Para poder extraer el cráneo se hizo uso de una segunda opción de SPM, *imCalc*. Ésta permite realizar manipulaciones algebraicas especificadas por el usuario sobre las imágenes de entrada.

En este caso específico, se realizó una suma de las tres máscaras mostradas en la figura 26, resultando en una nueva máscara donde en valores altos de intensidad (cercaos a 1) se mostrarán los tejidos que se pretenden segmentar mientras que los valores más bajos (cercaos a 0) de intensidad abarcarán el resto de la imagen (ver figura 33).



Figura 33. Máscara binaria utilizada para la extracción de cráneo.

Seguidamente, se realizó una multiplicación de la imagen original con la nueva máscara, de forma que las zonas de interés conservaron mientras que el resto de las zonas se eliminaron de la imagen original, tal y como se muestra en la figura 34.

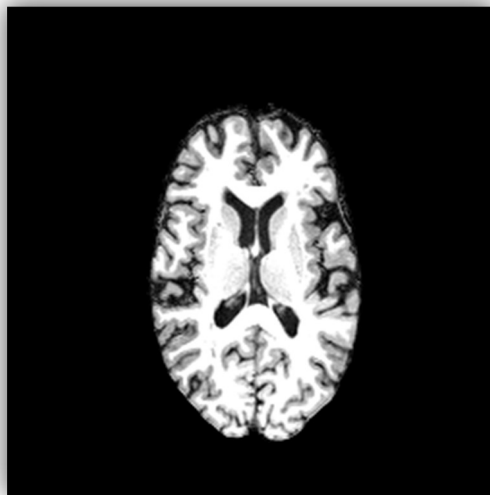


Figura 34. Resultado final tras aplicar *zero padding* y extracción de cráneo.

4.4.4. Normalización de intensidad

En vista del resultado obtenido, se siguió con una normalización de intensidad en el rango [0, 255] haciendo uso de la librería *OpenCV* de *Python* (ver figura 35).

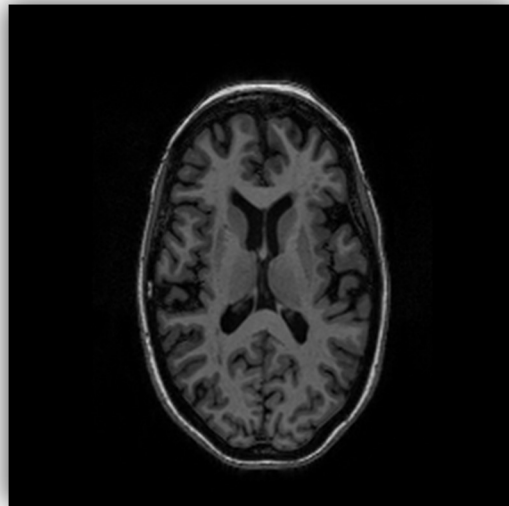


Figura 35. Resultado final tras aplicar *zero padding* y normalización de intensidad.

4.4.5. Eliminación de cortes axiales con ausencia de tejido cerebral

Entre las imágenes se observó gran cantidad de ellas con ausencia de tejido cerebral, resultado del proceso de adquisición. Por tanto, se decidió eliminarlas por no contener información. El resultado fue un total de 35.462 imágenes (ver tablas 7 y 8).

Etiqueta	Número de imágenes
CN	10.953
MCI	17.019
AD	7.489
TOTAL	35.461

Tabla 7. Clasificación multiclase. Número de imágenes *.png* asociadas a cada etiqueta tras eliminar los cortes axiales con ausencia de tejido cerebral.

Etiqueta	Número de imágenes
Normal	10.953
Anormal	24.508
TOTAL	35.461

Tabla 8. Clasificación binaria. Número de imágenes *.png* asociadas a cada etiqueta tras eliminar los cortes axiales con ausencia de tejido cerebral.

4.4.6. Eliminación de cortes axiales con una media de intensidad menor a 0.05

Tras la eliminación de las imágenes con ausencia de tejido cerebral, se siguieron observando cortes axiales irrelevantes. Es decir, se identificaron cortes que contenían muy poca información y que, por tanto, no eran clasificables (ver figura 36).

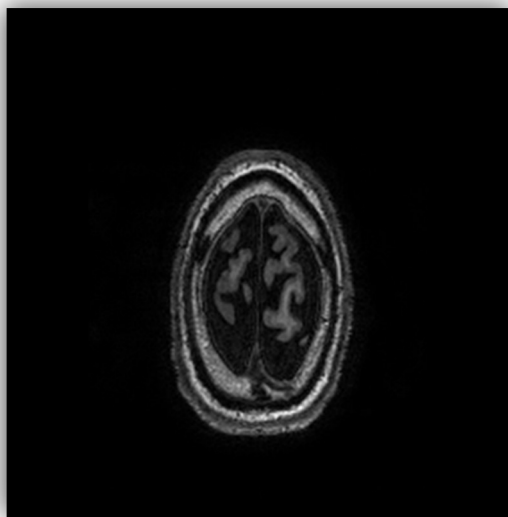


Figura 36. Corte axial con una media de intensidad igual a 0.037 (menor a 0.05).

Tras un pequeño estudio, se estableció que los cortes axiales deberían tener una media de intensidad superior o igual al valor 0.05, eliminando del conjunto de datos las imágenes que no cumplieran dicho criterio (ver tabla 9 y 10), quedando como resultado un total de 32.692 imágenes *.png*:

Etiqueta	Número de imágenes
CN	10.098
MCI	15.774
AD	6.820
TOTAL	32.692

Tabla 9. Clasificación multiclase. Número de imágenes *.png* asociadas a cada etiqueta tras eliminar los cortes axiales con una media de intensidad menor a 0.05.

Etiqueta	Número de imágenes
Normal	10.098
Anormal	22.594
TOTAL	32.692

Tabla 10. Clasificación binaria. Número de imágenes *.png* asociadas a cada etiqueta tras eliminar los cortes axiales con una media de intensidad menor a 0.05.

4.4.7. Eliminación de cortes axiales sin información anatómica de interés para la detección de AD

Con el objetivo de mejorar los resultados, se propuso considerar solo aquellos cortes axiales que mostraran las estructuras anatómicas cerebrales de interés para la enfermedad de Alzheimer, eliminando aquellos cortes donde se observaran estructuras sin información en cuanto a la enfermedad se refiere.

Esta decisión fue debida a una razón similar a la eliminación de cortes con ausencia de tejido cerebral. Es decir, de esta forma se evita introducir información contradictoria a la red, ya que cortes que no muestren estructuras afectadas por AD serían similares tanto en sujetos de la clase CN como en sujetos de la clase AD.

Con este objetivo, se revisaron diferentes artículos científicos que aportasen información acerca de las estructuras cerebrales mayormente afectadas por la AD y que, además, se tratase de cambios reconocibles en imágenes cerebrales de resonancia magnética. Los resultados obtenidos se pueden consultar en el apartado 1.1. sección *Regiones cerebrales mayormente afectadas*.

Con todo ello, se decidió preservar los cortes que abarcaban las estructuras que forman el lóbulo frontal, temporal y parietal, así como el sistema límbico. Esto se puede traducir en una eliminación de los cortes axiales donde solo se observan estructuras del lóbulo occipital o cerebelo ya que no se aprecian cambios relevantes en sus estructuras en la enfermedad de Alzheimer.

Para ello, se hizo uso del atlas Harvard-Oxford. Este atlas fue desarrollado en el centro de análisis morfométrico de la Universidad de Harvard y distribuido por *FSL*. Los datos son descargables de forma gratuita desde la fuente de datos original e incluyen el atlas de la superficie cerebral completo, así como información acerca de las etiquetas. [57]

El procedimiento fue el explicado a continuación. Previa realización del pre-procesado explicado anteriormente, se normalizó espacialmente el atlas Harvard-Oxford del espacio MNI al espacio nativo para cada uno de los sujetos, de forma que el resultado será la obtención de un atlas para cada sujeto que se encontrará en el mismo espacio nativo que el cerebro en cuestión.

La normalización espacial se entiende como la realización de transformaciones espaciales lineales (traslación, rotación y escalado) y no lineales (por ejemplo, mediante *B-splines*) para registrar un elemento en una plantilla o cambiar el espacio en el que se encuentra (en este caso, del espacio MNI al nativo).

El Instituto Neurológico de Montreal consiguieron, a través de gran cantidad de imágenes de RMN cerebral de sujetos normales, definir un cerebro que fuese representativo para toda la población. Por espacio MNI nos referimos, por tanto, al espacio en que esta plantilla tan utilizada en la actualidad se encuentra. [58]

Para ello, en primer lugar, se extrajo el campo de deformación inverso para cada una de las imágenes tridimensionales cerebrales haciendo uso, nuevamente, de la opción *Segmentation* de SPM.

El campo de deformación se puede definir como una gran matriz tridimensional que contiene las transformaciones necesarias para proyectar cada uno de los vóxeles de la imagen en un espacio distinto del que se encuentran. [59] En este caso concreto, desde el espacio MNI donde se encuentra el atlas al espacio nativo de cada una de las imágenes de resonancia magnética tridimensionales. De esta forma, el resultado fueron 150 campos de deformación inversos que contienen las transformaciones necesarias para realizar el camino contrario: del espacio MNI al espacio nativo de cada sujeto.

El siguiente paso fue aplicar dicho campo de deformación inverso al atlas Harvard-Oxford. Se hizo uso de nuevo del programa SPM, concretamente de la herramienta *Deformations* a la cual se puede acceder a través del *Batch Editor*. Llegados a este punto, simplemente se deben introducir como entradas el propio atlas y el campo de deformación de cada una de las imágenes. El resultado, por tanto, serán 150 atlas Harvard-Oxford en el espacio nativo de cada uno de los sujetos.

En tercer lugar, se eliminaron para los 150 atlas las estructuras anatómicas que resultaron no ser de interés para el estudio de la EA: cerebelo y lóbulo occipital. Se identificaron las etiquetas correspondientes a dichas estructuras en el atlas y, a continuación, se les dio un valor de 0 a éstas y un valor de 1 a todas las estructuras restantes, consiguiendo así 150 máscaras binarias.

Para finalizar, se identificaron las posiciones del primer y último 1 a lo largo del eje z para cada una de las máscaras binarias. Plasmando estas posiciones sobre los volúmenes tras la aplicación del pre-procesado explicado anteriormente, se consiguieron conocer los cortes axiales de interés. Es decir, el atlas nos sirvió para identificar las posiciones de las estructuras más relevantes para nuestro caso de estudio en el espacio nativo de la imagen.

Los cortes anteriormente mencionados fueron los utilizados tanto para el entrenamiento como validación de la red mientras que el resto no se consideraron (ver tablas 11 y 12).

Etiqueta	Número de imágenes
CN	6.378
MCI	9.907
AD	4.418
TOTAL	20.703

Tabla 11. Clasificación multiclase. Número de imágenes *.png* asociadas a cada etiqueta tras eliminar los cortes axiales sin información anatómica de interés para la AD.

Etiqueta	Número de imágenes
Normal	6.378
Anormal	14.325
TOTAL	20.703

Tabla 12. Clasificación binaria. Número de imágenes *.png* asociadas a cada etiqueta tras eliminar los cortes axiales sin información anatómica de interés para la AD.

4.5. Experimentos previos

Para la realización de cada uno de los experimentos las imágenes *.png* fueron divididas en tres conjuntos: entrenamiento, validación y *test*. El conjunto de entrenamiento es utilizado para entrenar el modelo dado y abarca el 70% de los datos; el conjunto de validación se utiliza para ser capaz de elegir entre modelos dependiendo de los resultados obtenidos, formado por un 20% de los datos; finalmente, el conjunto de *test* permite evaluar nuestro modelo, corroborando que sea generalizable y no se haya ajustado al conjunto de validación, abarcando otro 10% de los datos.

Se debe tener en cuenta que es fundamental que los datos de entrada a la red neuronal estén bien balanceados, tanto en el subconjunto de datos de entrenamiento como en el de prueba. Las redes neuronales tratan de buscar el camino que obtenga un mayor número de aciertos de forma que, si se cuenta con un mayor número de observaciones de la clase 1 que de la clase 2, la red opta por clasificar todos los datos como clase 1, consiguiendo así un mayor número de aciertos que de fallos.

Si revisamos, por ejemplo, nuestro conjunto de imágenes *.png* en la tabla 9 se puede comprobar cómo se tiene más del doble de imágenes de la clase *Anormal* que de la clase *Normal*. Teniendo en cuenta todo lo anterior, el número de observaciones presentes en cada clase quedará determinado por el número de imágenes del grupo de menor tamaño.

Previa realización de los experimentos de clasificación, se llevaron a cabo unos pequeños experimentos utilizando la arquitectura Resnet34 como referencia manteniendo las primeras capas congeladas, con el objetivo de extraer conclusiones extrapolables a cualquier experimento posterior. En cada uno de ellos se evaluó una de las técnicas de pre-procesado anteriormente mencionadas, siempre teniendo en cuenta los resultados obtenidos en experimentos previos.

4.5.1. Zero padding

❖ Sin *zero padding*

- Conjunto de datos empleado: 37.168 imágenes *.png* (ver tabla 5).
- Pre-procesado: ninguno.
- Etiquetas: Normal, Anormal
- Duración: 5 *epochs*
- Tamaño de imagen: 256x256
- *Batch size*: 64
- Arquitectura: *ResNet34*

❖ Con *zero padding*

- Conjunto de datos empleado: 37.168 imágenes *.png* (ver tabla 5).
- Pre-procesado: *zero padding*.
- Etiquetas: Normal, Anormal
- Duración: 5 *epochs*
- Tamaño de imagen: 256x256
- *Batch size*: 64
- Arquitectura: *ResNet34*

4.5.2. Extracción de cráneo

❖ Sin extracción de cráneo

- Conjunto de datos empleado: 37.168 imágenes *.png* (ver tabla 5).
- Pre-procesado: *zero padding*.
- Etiquetas: Normal, Anormal
- Duración: 5 *epochs*
- Tamaño de imagen: 256x256
- *Batch size*: 64
- Arquitectura: *ResNet34*

❖ Con extracción de cráneo

- Conjunto de datos empleado: 37.168 imágenes *.png* (ver tabla 5).
- Pre-procesado: *zero padding* y extracción de cráneo.
- Etiquetas: Normal, Anormal
- Duración: 5 *epochs*
- Tamaño de imagen: 256x256
- *Batch size*: 64
- Arquitectura: *ResNet34*

4.5.3. Tamaño de imagen

❖ Tamaño 128x128

- Conjunto de datos empleado: 37.168 imágenes *.png* (ver tabla 5).
- Pre-procesado: *zero padding*.
- Etiquetas: Normal, Anormal
- Duración: 5 *epochs*
- Tamaño de imagen: 128x128
- *Batch size*: 64
- Arquitectura: *ResNet34*

❖ 128x128 → 256x256

Este experimento se realizó aprovechando los pesos obtenidos del experimento anterior (128x128). Tras ello, se realizaron 5 *epochs* más con un tamaño de imagen diferente: 256x256.

- Conjunto de datos empleado: 37.168 imágenes *.png* (ver tabla 5).
- Pre-procesado: *zero padding* y extracción de cráneo.
- Etiquetas: Normal, Anormal
- Duración: 5 *epochs* (128x128) → 5 *epochs* (256x256)
- Tamaño de imagen: 128x128 → 256x256
- *Batch size*: 64
- Arquitectura: *ResNet34*

4.5.4. Normalización de intensidad

❖ Sin normalización de intensidad

- Conjunto de datos empleado: 37.168 imágenes *.png* (ver tabla 5).
- Pre-procesado: *zero padding*.
- Etiquetas: Normal, Anormal
- Duración: 5 *epochs*
- Tamaño de imagen: 128x128
- *Batch size*: 64
- Arquitectura: *ResNet34*

❖ Con normalización de intensidad

- Conjunto de datos empleado: 37.168 imágenes *.png* (ver tabla 5).
- Pre-procesado: *zero padding* y normalización de intensidad.
- Etiquetas: Normal, Anormal
- Duración: 5 *epochs*
- Tamaño de imagen: 128x128
- *Batch size*: 64
- Arquitectura: *ResNet34*

4.5.5. Eliminación de los cortes axiales con ausencia de tejido cerebral

- ❖ Sin eliminación de los cortes axiales con ausencia de tejido cerebral
 - Conjunto de datos empleado: 37.168 imágenes *.png* (ver tabla 5).
 - Pre-procesado: *zero padding* y normalización de intensidad.
 - Etiquetas: Normal, Anormal
 - Duración: 5 *epochs*
 - Tamaño de imagen: 128x128
 - *Batch size*: 64
 - Arquitectura: *ResNet34*

- ❖ Con eliminación de los cortes axiales con ausencia de tejido cerebral
 - Conjunto de datos empleado: 35.461 imágenes *.png* (ver tabla 7).
 - Pre-procesado: *zero padding* y normalización de intensidad.
 - Etiquetas: Normal, Anormal
 - Duración: 5 *epochs*
 - Tamaño de imagen: 128x128
 - *Batch size*: 64
 - Arquitectura: *ResNet34*

4.5.6. Eliminación de los cortes axiales con una media de intensidad menor a 0.05

- ❖ Sin eliminación de los cortes axiales con una media de intensidad menor a 0.05
 - Conjunto de datos empleado: 35.461 imágenes *.png* (ver tabla 7).
 - Pre-procesado: *zero padding* y normalización de intensidad.
 - Etiquetas: Normal, Anormal
 - Duración: 5 *epochs*
 - Tamaño de imagen: 128x128
 - *Batch size*: 64
 - Arquitectura: *ResNet34*

- ❖ Con eliminación de los cortes axiales con una media de intensidad menor a 0.05
 - Conjunto de datos empleado: 32.692 imágenes *.png* (ver tabla 9).
 - Pre-procesado: *zero padding* y normalización de intensidad.
 - Etiquetas: Normal, Anormal
 - Duración: 5 *epochs*
 - Tamaño de imagen: 128x128
 - *Batch size*: 64
 - Arquitectura: *ResNet34*

4.5.7. Eliminación de los cortes axiales sin información anatómica relevante para la detección de AD

- ❖ Sin eliminación de los cortes axiales sin información anatómica relevante para la detección de AD
 - Conjunto de datos empleado: 32.692 imágenes *.png* (ver tabla 9).
 - Pre-procesado: *zero padding* y normalización de intensidad.
 - Etiquetas: Normal, Anormal
 - Duración: 5 *epochs*
 - Tamaño de imagen: 128x128
 - *Batch size*: 64
 - Arquitectura: *ResNet34*

- ❖ Con eliminación de los cortes axiales sin información anatómica relevante para la detección de AD
 - Conjunto de datos empleado: 20.703 imágenes *.png* (ver tabla 11).
 - Pre-procesado: *zero padding* y normalización de intensidad.
 - Etiquetas: Normal, Anormal
 - Duración: 5 *epochs*
 - Tamaño de imagen: 128x128
 - *Batch size*: 64
 - Arquitectura: *ResNet34*

4.5.8. Elección de la arquitectura CNN

- ❖ *ResNet34*
 - Conjunto de datos empleado: 20.703 imágenes *.png* (ver tabla 11).
 - Pre-procesado: *zero padding*, normalización de intensidad y eliminación de los cortes axiales sin información anatómica relevante para la detección de AD.
 - Etiquetas: Normal, Anormal
 - Duración: 5 *epochs*
 - Tamaño de imagen: 128x128
 - *Batch size*: 64
 - Arquitectura: *ResNet34*

- ❖ *ResNet50*
 - Conjunto de datos empleado: 20.703 imágenes *.png* (ver tabla 11).
 - Pre-procesado: *zero padding*, normalización de intensidad y eliminación de los cortes axiales sin información anatómica relevante para la detección de AD.
 - Etiquetas: Normal, Anormal
 - Duración: 5 *epochs*
 - Tamaño de imagen: 128x128
 - *Batch size*: 64

- Arquitectura: *ResNet50*

❖ *SqueezeNet1_1*

- Conjunto de datos empleado: 20.703 imágenes *.png* (ver tabla 11).
- Pre-procesado: *zero padding*, normalización de intensidad y eliminación de los cortes axiales sin información anatómica relevante para la detección de AD.
- Etiquetas: Normal, Anormal
- Duración: 5 *epochs*
- Tamaño de imagen: 128x128
- *Batch size*: 64
- Arquitectura: *SqueezeNet1_1*

❖ *DenseNet121*

- Conjunto de datos empleado: 20.703 imágenes *.png* (ver tabla 11).
- Pre-procesado: *zero padding*, normalización de intensidad y eliminación de los cortes axiales sin información anatómica relevante para la detección de AD.
- Etiquetas: Normal, Anormal
- Duración: 5 *epochs*
- Tamaño de imagen: 128x128
- *Batch size*: 64
- Arquitectura: *DenseNet121*

4.5.9. Elección del *batch size*

❖ *Batch size* base

- Conjunto de datos empleado: 20.703 imágenes *.png* (ver tabla 11).
- Pre-procesado: *zero padding*, normalización de intensidad y eliminación de los cortes axiales sin información anatómica relevante para la detección de AD.
- Etiquetas: Normal, Anormal
- Duración: 5 *epochs*
- Tamaño de imagen: 128x128
- *Batch size*: 64
- Arquitectura: *DenseNet121*

❖ *Batch size* mayor

- Conjunto de datos empleado: 20.703 imágenes *.png* (ver tabla 11).
- Pre-procesado: *zero padding*, normalización de intensidad y eliminación de los cortes axiales sin información anatómica relevante para la detección de AD.
- Etiquetas: Normal, Anormal
- Duración: 5 *epochs*
- Tamaño de imagen: 128x128
- *Batch size*: 128
- Arquitectura: *DenseNet121*

❖ *Batch size* menor

- Conjunto de datos empleado: 20.703 imágenes *.png* (ver tabla 11).
- Pre-procesado: *zero padding*, normalización de intensidad y eliminación de los cortes axiales sin información anatómica relevante para la detección de AD.
- Etiquetas: Normal, Anormal
- Duración: 5 *epochs*
- Tamaño de imagen: 128x128
- *Batch size*: 32
- Arquitectura: *DenseNet121*

4.6. Experimentos finales

Teniendo en cuenta las conclusiones extraídas de los experimentos previos expuestos en el apartado 3.4. se procedió al entrenamiento y extracción de métricas de los clasificadores finales.

Todas las imágenes utilizadas en estos experimentos cuentan con un preprocesado que engloba *zero padding*, normalización de intensidad y eliminación de cortes axiales sin información anatómica relevante para la detección de AD. La arquitectura CNN empleada fue *DenseNet121* con un *batch size* igual a 64 y un tamaño de imagen de 128x128.

4.6.1. Clasificador binario

Este clasificador tuvo como objetivo realizar una clasificación Normal (CN) – Anormal (MCI, AD). Tras la aplicación de las técnicas de pre-procesado correspondientes el total de imágenes de las que se dispuso fue de 20.703 (ver tabla 12).

Teniendo en cuenta el número de imágenes del grupo minoritario, en este caso *Normal*, el número de imágenes total con el que trabajó la red neuronal convolucional fue de 12.756, tal y como se muestra en la tabla 13.

Conjunto	Número de imágenes
Entrenamiento	9.184
Validación	2.296
<i>Test</i>	1.276
Total	12.756

Tabla 13. Experimento binario. Número de imágenes que forman cada subconjunto de datos: entrenamiento, validación y *test*.

4.6.2. Clasificador multiclase

Se siguió con un clasificador multiclase que fuese capaz de clasificar entre muestras CN, MCI y AD. Para ello, se trabajó con un total de 20.703 imágenes (ver tabla 11). El grupo minoritario fue el formado por las muestras clasificadas como AD. Los subconjuntos de muestras con los que se trabajó finalmente se muestran en la tabla 14.

Conjunto	Número de imágenes
Entrenamiento	9.543
Validación	2.385
<i>Test</i>	1.326
Total	13.254

Tabla 14. Experimento multiclase. Número de imágenes que forman cada subconjunto de datos: entrenamiento, validación y *test*.

Para poder evaluar los resultados de *test* para cada uno de los grupos por separado se crearon tres subconjuntos de *test* a partir del mostrado en la tabla 14 (ver tablas 15, 16 y 17).

Conjunto	Número de imágenes
CN	442
MCI+AD	442
Total <i>test</i>	884

Tabla 15. Experimento multiclase. Subconjunto de *test* para evaluar CN.

Conjunto	Número de imágenes
MCI	442
CN+AD	442
Total <i>test</i>	884

Tabla 16. Experimento multiclase. Subconjunto de *test* para evaluar MCI.

Conjunto	Número de imágenes
AD	442
CN+MCI	442
Total <i>test</i>	884

Tabla 17. Experimento multiclase. Subconjunto de *test* para evaluar AD.

4.6.3. Clasificador multiclase en cascada

Con el objetivo de mejorar el clasificador anterior, se siguió con un clasificador multiclase donde se hizo uso de dos clasificadores binarios en cascada, tal y como se muestra en la figura 37.

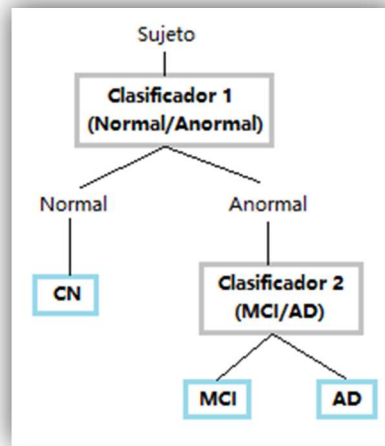


Figura 37. Clasificador en cascada.

Es decir, como primer paso, los sujetos de entrada se clasificaron como *Normales/Anormales* aprovechando los pesos del clasificador binario ya realizado (apartado 3.5.1). Seguidamente, los sujetos identificados como *Normales* se asignaron como *CN* mientras que los sujetos identificados como *Anormales* pasaron por un segundo clasificador capaz de diferenciar entre sujetos MCI y AD.

En primer lugar, fue necesario entrenar un nuevo clasificador MCI – AD. Se disponía de un total de 8.836 imágenes teniendo en cuenta el grupo minoritario, AD (ver tabla 11). Los subconjuntos de datos con los que trabajó la CNN se muestran en la tabla 18.

Conjunto	Número de imágenes
Entrenamiento	6.362
Validación	1.590
Test	884
Total	8.836

Tabla 18. Clasificador MCI - AD. Número de imágenes que forman cada subconjunto de datos: entrenamiento, validación y *test*.

En segundo lugar, se evaluó el clasificador MCI – AD con el subconjunto de datos de *test* mostrado en la tabla 18.

Finalmente, se aplicó el clasificador multiclase en cascada para los datos de *test* vistos para el clasificador multiclase (ver tablas 15, 16 y 17) y se obtuvieron las métricas considerando datos de las tres etiquetas.

CAPÍTULO 5. RESULTADOS.

5.1. Experimentos previos

Todos experimentos previos se realizaron utilizando la arquitectura CNN *Resnet34*, durante 5 *epochs* y manteniendo las primeras capas congeladas.

5.1.1. Zero padding

En ambos experimentos, se utilizó un tamaño de imagen de 256x256 y un *batch size* de 64. En el primer experimento, las imágenes no contaron con ningún tipo de pre-procesado mientras que, en el segundo, se ha hizo uso de la técnica *zero padding* como pre-procesado.

❖ Sin *zero padding*

<i>epochs</i>	pérdidas entrenamiento	pérdidas validación	exactitud	tiempo(min)
0	0.944185	0.633476	0.667025	02:24
1	0.706393	0.559778	0.728643	02:33
2	0.635587	0.537494	0.738733	02:41
3	0.603312	0.528929	0.737387	02:25
4	0.591317	0.527457	0.739002	01:55

Tabla 19. Resultados experimento previo 1. Sin *zero padding*.

❖ Con *zero padding*

<i>epochs</i>	pérdidas entrenamiento	pérdidas validación	exactitud	tiempo(min)
0	0.710476	0.572781	0.725279	00:45
1	0.588656	0.546001	0.747881	00:43
2	0.520277	0.474713	0.781918	01:02
3	0.458759	0.422006	0.814207	01:17
4	0.427432	0.381721	0.827391	00:59

Tabla 20. Resultados experimento previo 1. Con *zero padding*.

5.1.2. Extracción de cráneo

El tamaño de imagen empleado fue de 256x256 y un *batch size* de valor 64. Se aplicó *zero padding* como parte del pre-procesado y, solo en el segundo de los experimentos se realizó además extracción de cráneo.

- ❖ Sin extracción de cráneo (ver tabla 20)
- ❖ Con extracción de cráneo

<i>epochs</i>	pérdidas entrenamiento	pérdidas validación	exactitud	tiempo(min)
0	0.714868	0.625294	0.694740	00:44
1	0.640051	0.643057	0.698103	00:49
2	0.627305	0.612074	0.693125	00:48
3	0.605378	0.592653	0.701601	00:48
4	0.593364	0.590852	0.702274	00:42

Tabla 21. Resultados experimento previo 2. Con extracción de cráneo.

5.1.3. Tamaño de imagen

El *batch size* utilizado fue de 64 en ambos experimentos. En vista de los resultados obtenidos, el pre-procesado aplicado a las imágenes utilizadas en estos experimentos fue únicamente *zero padding*. El primero de los experimentos tuvo una duración de 5 *epochs* haciendo uso de un tamaño de imagen de 128x128. Seguidamente, y para el segundo experimento, se utilizaron los pesos actualizados durante estas primeras 5 *epochs* y, a continuación, se entrenó durante 5 *epochs* más utilizando un tamaño de imagen nuevo de 256x256.

- ❖ 128x128

<i>epochs</i>	pérdidas entrenamiento	pérdidas validación	exactitud	tiempo(min)
0	0.642722	0.598841	0.680344	00:57
1	0.604045	0.548368	0.736580	00:56
2	0.580099	0.513942	0.761200	00:57
3	0.524982	0.475015	0.782053	01:09
4	0.517924	0.462776	0.785013	01:07

Tabla 22. Resultados experimento previo 3. 128x128.

❖ 128x128 → 256x256

<i>epochs</i>	pérdidas	pérdidas	exactitud	tiempo(min)
	entrenamiento	validación		
0	0.612580	0.592043	0.702543	01:29
1	0.577165	0.573671	0.717745	01:52
2	0.579492	0.563077	0.722185	02:39
3	0.571827	0.558991	0.726355	02:06
4	0.553903	0.560051	0.723261	02:37

Tabla 23. Resultados experimento previo 3. 128x128 → 256x256

5.1.4. Normalización de intensidad

Se aplicó un tamaño de imagen de 256x256 y un *batch size* igual a 64. En el primero de los experimentos el único pre-procesado aplicado fue *zero padding* mientras que, en el segundo, también se realizó una normalización de intensidad.

❖ Sin normalización de intensidad (ver tabla 22)

❖ Con normalización de intensidad

<i>epochs</i>	pérdidas	pérdidas	exactitud	tiempo(min)
	entrenamiento	validación		
0	0.703143	0.660786	0.657339	00:27
1	0.605629	0.592307	0.715324	00:28
2	0.554732	0.512244	0.753262	00:29
3	0.503760	0.460476	0.791336	00:31
4	0.468697	0.424360	0.808422	00:31

Tabla 24. Resultados experimento previo 4. Con normalización de intensidad.

5.1.5. Eliminación de los cortes axiales con ausencia de tejido cerebral

Las imágenes de entrada para estos experimentos contaban con las técnicas de *zero-padding* y normalización de intensidad como parte del pre-procesado. El *batch size* utilizado fue de 64 y el tamaño de imagen de 128x128. La diferencia entre ambos experimentos es que, en el segundo, se eliminaron aquellos cortes axiales cuya media de intensidad fuese igual a 0.

❖ Sin eliminación de los cortes axiales con ausencia de tejido cerebral. (ver tabla 24)

- ❖ Con eliminación de los cortes axiales con ausencia de tejido cerebral

<i>epochs</i>	pérdidas entrenamiento	pérdidas validación	exactitud	tiempo(min)
0	0.738637	0.644271	0.684715	00:29
1	0.573254	0.532555	0.738720	00:27
2	0.511835	0.472834	0.781585	00:29
3	0.459428	0.412084	0.813452	00:31
4	0.426012	0.388012	0.825578	00:32

Tabla 25. Resultados experimento previo 5. Con eliminación de los cortes axiales con ausencia de tejido cerebral.

5.1.6. Eliminación de los cortes axiales con una media de intensidad menor a 0.05

De nuevo, el único pre-procesado aplicado a las imágenes de entrada fue *zero-padding* y normalización de intensidad. El *batch size* utilizado fue de 64 y el tamaño de imagen de 128x128. En este caso, en el primero de los experimentos se eliminaron aquellas imágenes con una media igual a 0 mientras que en el segundo se eliminaron los cortes axiales con una media de intensidad menor a 0.05.

- ❖ Sin eliminación de los cortes axiales con una media de intensidad menor a 0.05 (ver tabla 25)
- ❖ Con eliminación de los cortes axiales con una media de intensidad menor a 0.05

<i>epochs</i>	pérdidas entrenamiento	pérdidas validación	exactitud	tiempo(min)
0	0.746502	0.668948	0.699296	00:26
1	0.581043	0.542660	0.715050	00:28
2	0.510483	0.445118	0.799021	00:30
3	0.462756	0.389608	0.829917	00:31
4	0.412911	0.369950	0.836953	00:31

Tabla 26. Resultados experimento previo 6. Eliminación de los cortes axiales con una media de intensidad menor a 0.05.

5.1.7. Eliminación de los cortes axiales sin información anatómica relevante para la detección de AD

El pre-procesado aplicado a las imágenes de entrada fue *zero-padding* y normalización de intensidad. El *batch size* utilizado fue de 64 y el tamaño de imagen de 128x128. En el

primero de los experimentos se eliminaron aquellas imágenes con una media de intensidad menor a 0.05 mientras que en el segundo se eliminaron los cortes axiales sin información anatómica relevante para la detección de AD según las conclusiones extraídas del atlas *Harvard-Oxford* (ver apartado 4.4.7).

- ❖ Con eliminación de los cortes axiales sin información anatómica relevante para la detección de AD. (ver tabla 26)
- ❖ Sin eliminación de los cortes axiales sin información anatómica relevante para la detección de AD

<i>epochs</i>	pérdidas entrenamiento	pérdidas validación	exactitud	tiempo(min)
0	0.803087	0.584088	0.700966	00:17
1	0.560692	0.508239	0.764010	00:18
2	0.499558	0.417379	0.817874	00:19
3	0.423826	0.332016	0.849758	00:19
4	0.380530	0.309337	0.862319	00:19

Tabla 27. Resultados experimento previo 7. Eliminación de los cortes axiales sin información anatómica relevante para la detección de AD.

5.1.8. Elección de la arquitectura CNN

El *batch size* empleado fue de 64 y el tamaño de imagen de 128x128. Se aplicó a las imágenes un pre-procesado de *zero padding*, normalización de intensidad y eliminación de cortes axiales sin información anatómica relevante para la detección de AD. En el primero de los experimentos, la arquitectura CNN usada fue *ResNet32*, *ResNet50* en el segundo, *Squeezenet1_1* en el tercero y *DenseNet121* en el cuarto.

- ❖ *ResNet34* (ver tabla 27)

- ❖ *ResNet50*

<i>epochs</i>	pérdidas entrenamiento	pérdidas validación	exactitud	tiempo(min)
0	0.744327	0.972041	0.627536	00:27
1	0.571916	0.536936	0.743237	00:27
2	0.442671	0.346099	0.842754	00:27
3	0.362129	0.255881	0.891546	00:27
4	0.318958	0.232500	0.904106	00:28

Tabla 28. Resultados experimento previo 8. *ResNet50*.

❖ *Squeezenet1_1*

<i>epochs</i>	pérdidas entrenamiento	pérdidas validación	exactitud	tiempo(min)
0	0.801659	0.718163	0.682850	00:10
1	0.618368	0.573814	0.703865	00:11
2	0.580175	0.557235	0.730435	00:11
3	0.552406	0.529091	0.751449	00:13
4	0.534140	0.495503	0.757488	00:13

Tabla 29. Resultados experimento previo 8. *Squeezenet1_1*.

❖ *Densenet121*

<i>epochs</i>	pérdidas entrenamiento	pérdidas validación	exactitud	tiempo(min)
0	0.712824	0.739552	0.735749	00:29
1	0.540203	0.514329	0.755556	00:29
2	0.390048	0.281240	0.875362	00:29
3	0.282958	0.184552	0.926812	00:30
4	0.210866	0.146299	0.945169	00:30

Tabla 30. Resultados experimento previo 8. *DenseNet121*.

5.1.9. Elección del *batch size*

En vista de los resultados obtenidos en el apartado 5.1.8, la arquitectura CNN aplicada en estos experimentos fue *DenseNet121*. El *batch size* utilizado fue, nuevamente, 64 y el tamaño de imagen 128x128. Como pre-procesado se aplicó *zero padding*, normalización de intensidad y eliminación de cortes axiales sin información anatómica relevante para la detección de AD.

❖ *Batch size* base (ver tabla 30)

❖ *Batch size* mayor

<i>epochs</i>	pérdidas entrenamiento	pérdidas validación	exactitud	tiempo(min)
0	0.759356	0.632387	0.716184	00:37
1	0.474480	0.411446	0.824638	00:37
2	0.370355	0.301550	0.870048	00:38
3	0.270071	0.191613	0.921256	00:39
4	0.187011	0.128939	0.933140	00:41

Tabla 31. Resultados experimento previo 9. *Batch size* mayor, 128.

❖ *Batch size* menor

<i>epochs</i>	pérdidas entrenamiento	pérdidas validación	exactitud	tiempo(min)
0	0.785314	0.568118	0.740338	00:57
1	0.564810	0.478700	0.769082	00:58
2	0.491931	0.428309	0.808454	01:20
3	0.442506	0.378756	0.836715	01:54
4	0.412471	0.365436	0.846618	01:47

Tabla 32. Resultados experimento previo 9. *ResNet50*. *Batch size* menor, 32.

5.2. Experimentos finales

Los clasificadores finales se llevaron a cabo haciendo uso de la arquitectura *DenseNet121*, con un *batch size* de 64 y un tamaño de imagen de 128x128. Las técnicas de pre-procesado empleadas fueron *zero padding*, normalización de intensidad y eliminación de cortes axiales sin información anatómica relevante para la detección de AD.

5.2.1. Clasificador binario

Se obtuvieron mejoras durante cinco *epochs* manteniendo las primeras capas congeladas y durante 5 *epochs* tras descongelarlas. Se consiguió una exactitud de 0.919.

Capas congeladas

<i>epochs</i>	pérdidas entrenamiento	pérdidas validación	exactitud	tiempo
0	0.922037	1.269915	0.592335	00:21
1	0.670500	0.515491	0.760889	00:23
2	0.490285	0.417299	0.809669	00:23
3	0.388272	0.305923	0.863676	00:23
4	0.315944	0.269037	0.886324	00:23

Tabla 33. Clasificador binario, *DenseNet121*. Capas congeladas.

Capas descongeladas

<i>epochs</i>	pérdidas entrenamiento	pérdidas validación	exactitud	tiempo
0	0.291861	0.254728	0.896341	00:26
1	0.279090	0.240112	0.904617	00:28
2	0.268119	0.223858	0.914634	00:29
3	0.261952	0.213739	0.918118	00:30
4	0.247189	0.212316	0.919861	00:30

Tabla 34. Clasificador binario, *DenseNet121*. Capas descongeladas.

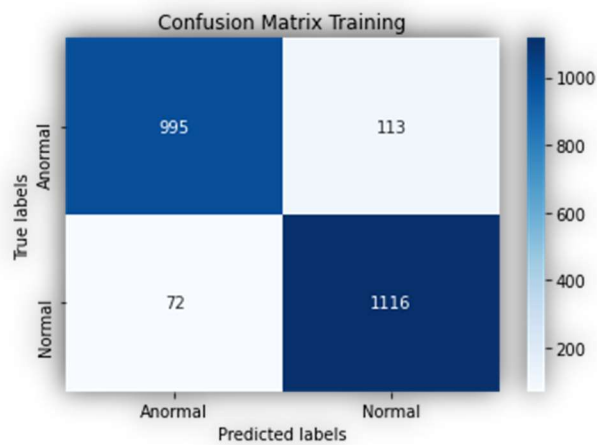


Figura 38. Clasificador binario. Matriz de confusión subconjunto de validación.

Seguidamente, se evaluó el modelo con los datos de *test*.

Modelo de <i>test</i>	exactitud	precisión	sensibilidad	especificidad	f1 <i>score</i>	AUC
normal vs anormal	0.905172	0.877372	0.942006	0.868339	0.908541	0.905172

Tabla 35. Clasificador binario. Métricas subconjunto de *test*.

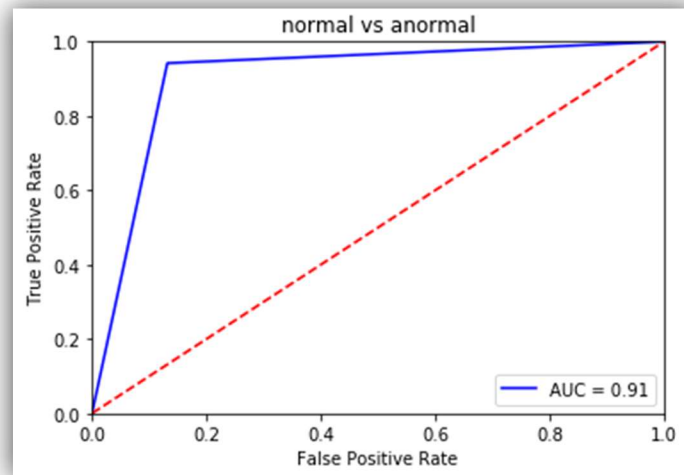


Figura 39. Clasificador binario. Curva ROC subconjunto de *test*.

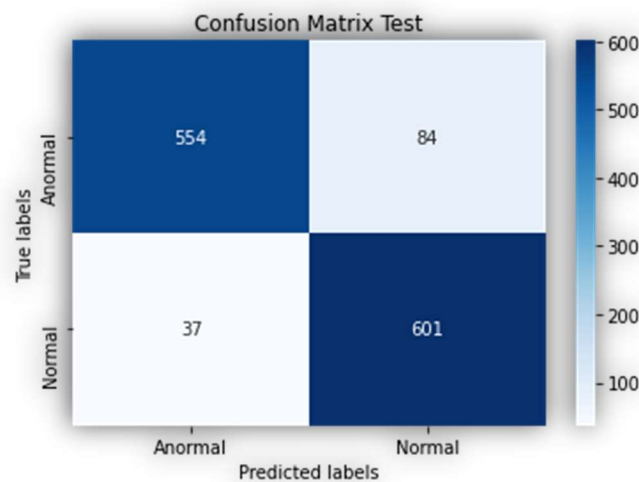


Figura 40. Clasificador binario. Matriz de confusión subconjunto de *test*.

5.2.2. Clasificador multiclase

Para este primer clasificador multiclase se obtuvo, para el subconjunto de datos de validación, una exactitud del 0.90. En cuanto al entrenamiento, se consiguieron mejoras durante 5 *epochs* en el caso de las primeras capas descongeladas y durante 15 *epochs* una vez descongeladas éstas.

Capas congeladas

<i>epochs</i>	pérdidas entrenamiento	pérdidas validación	exactitud	tiempo
0	1.350536	1.216620	0.491824	00:25
1	1.039707	0.937241	0.590356	00:25
2	0.825516	0.712332	0.692243	00:26
3	0.685478	0.549825	0.795388	00:25
4	0.549607	0.469454	0.821803	00:27

Tabla 36. Clasificador multiclase, *DenseNet121*. Capas congeladas.

Capas descongeladas

<i>epochs</i>	pérdidas entrenamiento	pérdidas validación	exactitud	tiempo
0	0.528695	0.459056	0.823899	00:29
1	0.510404	0.428740	0.836478	00:30
2	0.496991	0.407776	0.844444	00:32
3	0.474782	0.395227	0.850314	00:32
4	0.480616	0.393771	0.851992	00:32
5	0.475812	0.391555	0.854927	00:32
6	0.449526	0.358190	0.863312	00:32
7	0.423685	0.320704	0.890566	00:32
8	0.409819	0.314601	0.889308	00:34
9	0.383124	0.305903	0.892243	00:33
10	0.380972	0.301733	0.891405	00:33
11	0.376570	0.287363	0.898533	00:32
12	0.358184	0.274026	0.903564	00:33
13	0.355603	0.270791	0.899790	00:32
14	0.344859	0.271633	0.900629	00:35

Tabla 37. Clasificador multiclase, *DenseNet121*. Capas descongeladas.

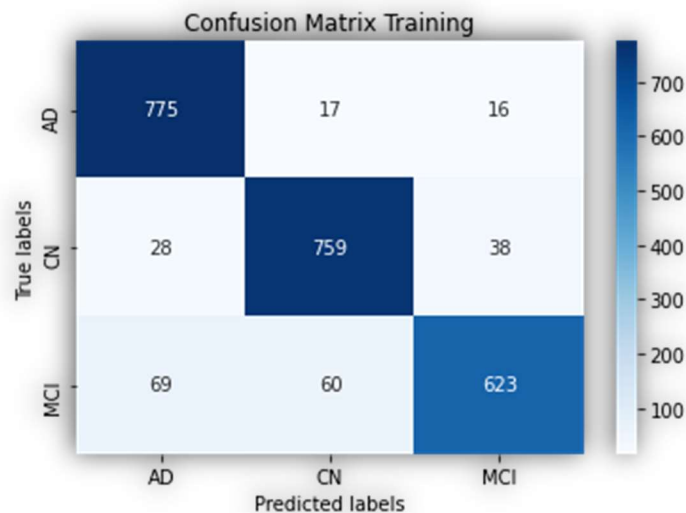


Figura 41. Clasificador multiclase. Matriz de confusión subconjunto de validación.

A continuación, y para evaluar el modelo, se extrajeron las diferentes métricas para los subconjuntos de datos de *test*.

❖ CN vs MCI+AD

Modelo de <i>test</i>	exactitud	precisión	sensibilidad	especificidad	f1 score	AUC
CN vs MCI+AD	0.935520	0.952941	0.916290	0.954751	0.934256	0.935520

Tabla 38. Clasificador multiclase. Métricas subconjunto de *test* CN vs MCI+AD.

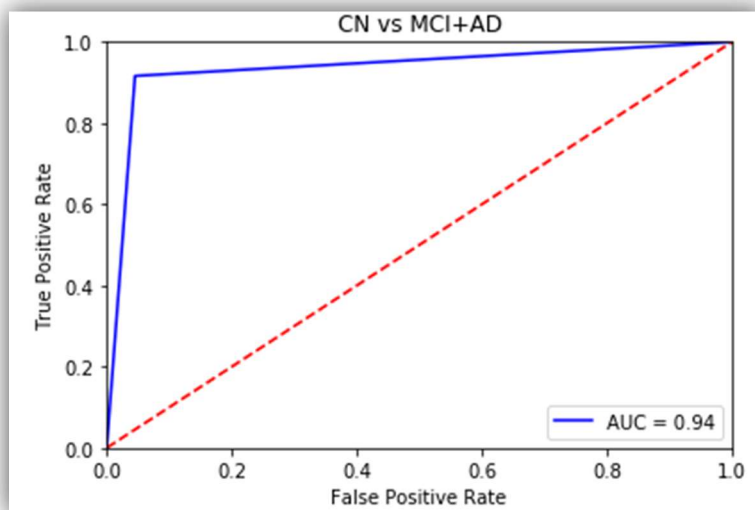


Figura 42. Clasificador multiclase. Curva ROC subconjunto de *test* CN vs MCI+AD.

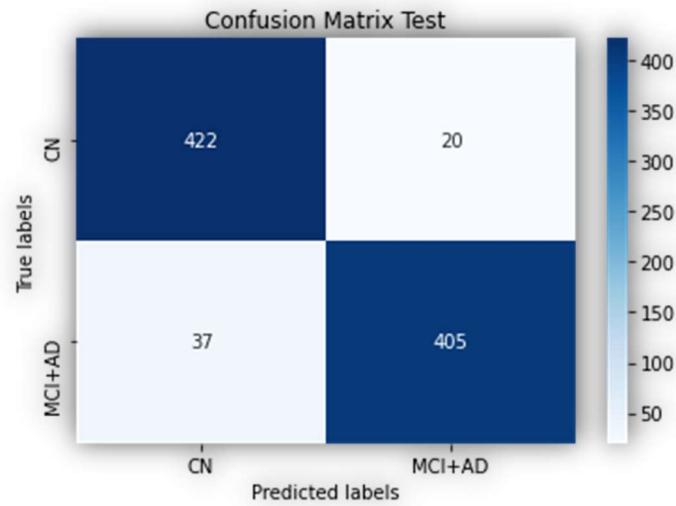


Figura 43. Clasificador multiclase. Matriz de confusión subconjunto de *test* CN vs MCI+AD.

❖ MCI vs CN+AD

Modelo de <i>test</i>	exactitud	precisión	sensibilidad	especificidad	f1 score	AUC
MCI vs CN+AD	0.906109	0.976127	0.832579	0.979638	0.898657	0.906109

Tabla 39. Clasificador multiclase. Métricas subconjunto de *test* MCI vs CN+AD.

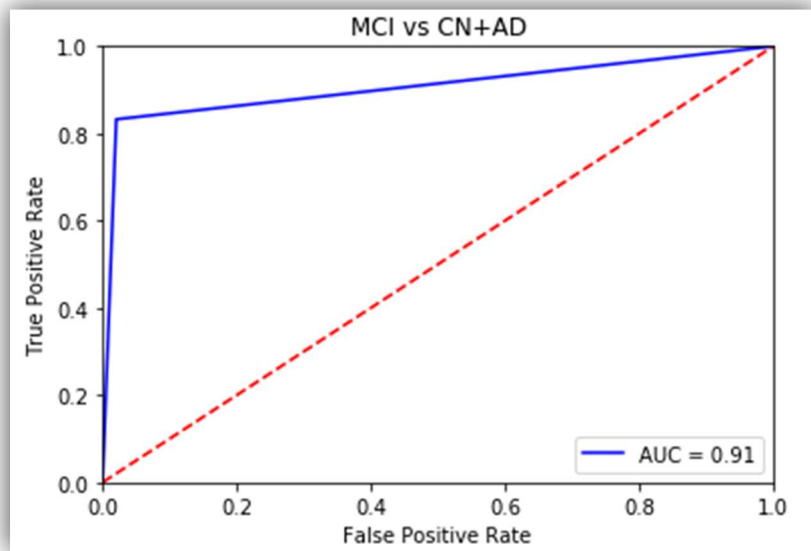


Figura 44. Clasificador multiclase. Curva ROC subconjunto de *test* MCI vs CN+AD.

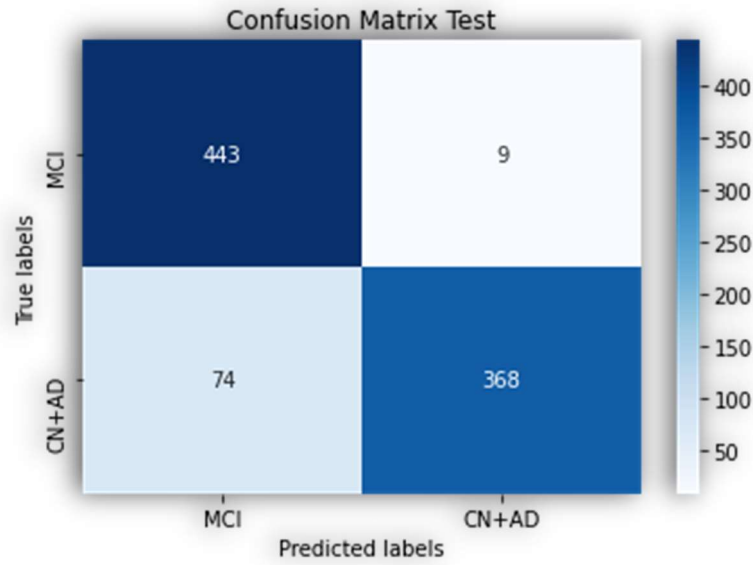


Figura 45. Clasificador multiclase. Matriz de confusión subconjunto de *test* MCI vs CN+AD.

❖ AD vs CN+MCI

Modelo de <i>test</i>	exactitud	precisión	sensibilidad	especificidad	f1 score	AUC
AD vs MCI+CN	0.951357	0.927195	0.979638	0.923077	0.952695	0.951357

Tabla 40. Clasificador multiclase. Métricas subconjunto de *test* AD vs MCI+CN.

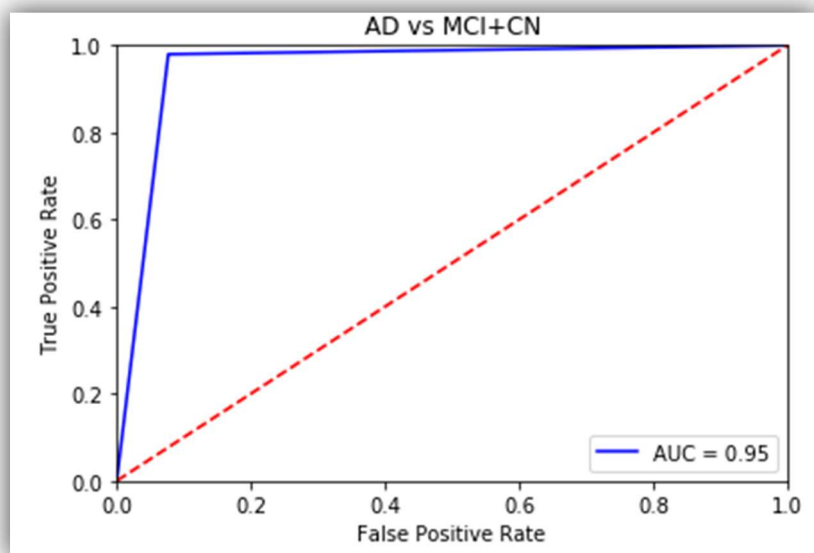


Figura 46. Clasificador multiclase. Curva ROC subconjunto de *test* AD vs MCI+CN.

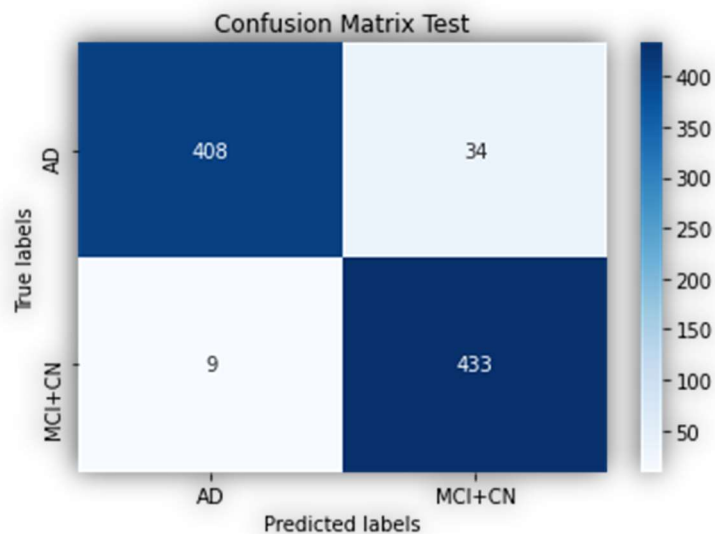


Figura 47. Clasificador multiclase. Matriz de confusión subconjunto de *test* AD vs MCI+CN.

5.2.3. Clasificador multiclase en cascada

❖ Clasificador MCI – AD

En este nuevo clasificador se observaron mejoras durante las primeras 5 *epochs* manteniendo las primeras capas congeladas y durante 15 *epochs* una vez descongeladas dichas capas. Los resultados obtenidos se pueden ver en las tablas 41 y 42, así como en la figura 28.

Capas congeladas

<i>epochs</i>	pérdidas entrenamiento	pérdidas validación	exactitud	tiempo
0	0.920409	0.747410	0.605031	00:29
1	0.775375	0.633586	0.724528	00:29
2	0.537046	0.383942	0.825157	00:22
3	0.404401	0.280127	0.882390	00:27
4	0.310367	0.252539	0.894969	00:24

Tabla 41. Clasificador MCI - AD, *DenseNet121*. Capas congeladas.

Capas descongeladas

<i>epochs</i>	pérdidas entrenamiento	pérdidas validación	exactitud	tiempo
0	0.296512	0.214912	0.915723	01:12
1	0.285105	0.203300	0.923899	00:40
2	0.273990	0.213890	0.918239	00:56
3	0.253415	0.197026	0.922642	00:36
4	0.249192	0.190932	0.923899	00:43
5	0.262206	0.178523	0.935220	01:42
6	0.258684	0.174004	0.938994	00:39
7	0.254112	0.174336	0.938365	00:42
8	0.244623	0.170192	0.940880	00:44
9	0.240748	0.170150	0.938994	00:55
10	0.239603	0.167649	0.944025	00:35
11	0.229893	0.163225	0.944654	00:34
12	0.240425	0.163531	0.942138	00:32
13	0.237192	0.158217	0.944025	00:29
14	0.224108	0.157153	0.949686	00:29

Tabla 42. Clasificador MCI - AD, *DenseNet121*. Capas descongeladas.

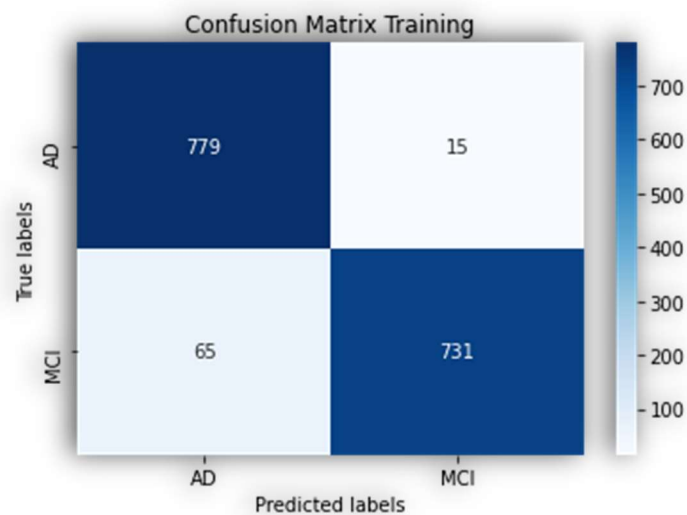


Figura 48. Clasificador MCI - AD. Matriz de confusión subconjunto de validación.

Con el objetivo de evaluar este clasificador, se obtuvieron las métricas para el subconjunto de datos de *test*.

Modelo de <i>test</i>	exactitud	precisión	sensibilidad	especificidad	f1 <i>score</i>	AUC
MCI vs AD	0.940045	0.977887	0.900452	0.979638	0.937574	0.940045

Tabla 43. Clasificador MCI - AD. Métricas subconjunto de *test*.

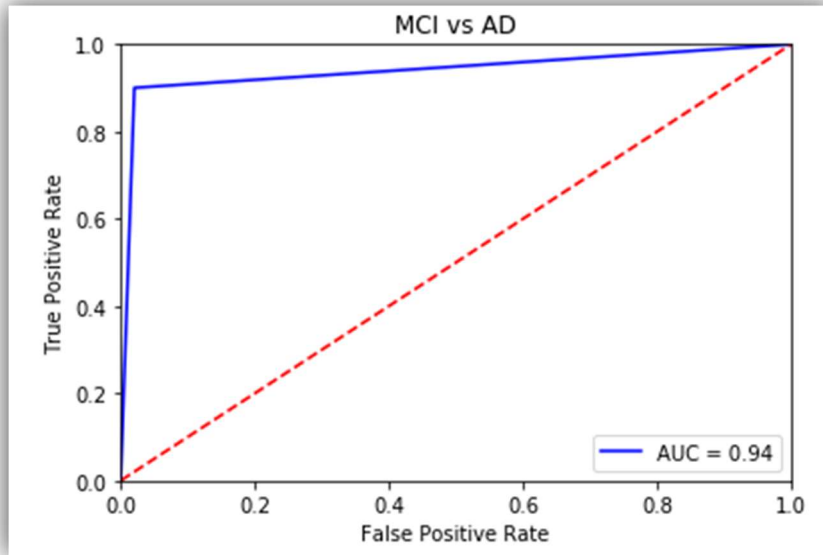


Figura 49. Clasificador MCI - AD. Curva ROC subconjunto de *test*.

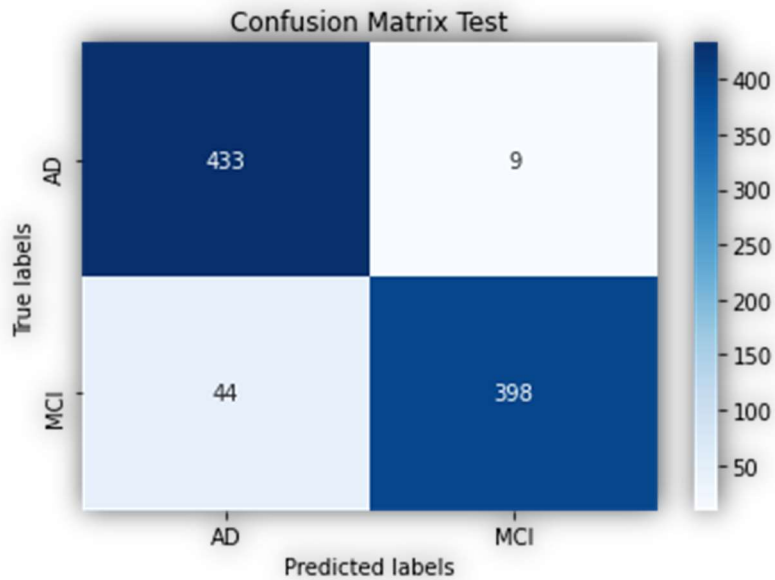


Figura 50. Clasificador MCI - AD. Matriz de confusión subconjunto de *test*.

❖ Clasificador en cascada

Como se ha explicado en el apartado 3.5.3 este clasificador se formó a partir de dos clasificadores binarios en cascada. Los resultados para los datos de validación del primer clasificador pueden consultarse en el apartado 5.2.1 y, en el caso del segundo clasificador, los resultados son los expuestos anteriormente (Clasificador MCI - AD). Por esta razón, simplemente se estudiarán en esta sección los resultados para el subconjunto de datos de *test*.

- CN vs MCI+AD

Modelo de <i>test</i>	exactitud	precisión	sensibilidad	especificidad	f1 score	AUC
CN vs MCI+AD	0.91629	0.889831	0.950226	0.882353	0.919037	0.91629

Tabla 44. Clasificador multiclase en cascada. Métricas subconjunto de *test* CN vs MCI+AD.

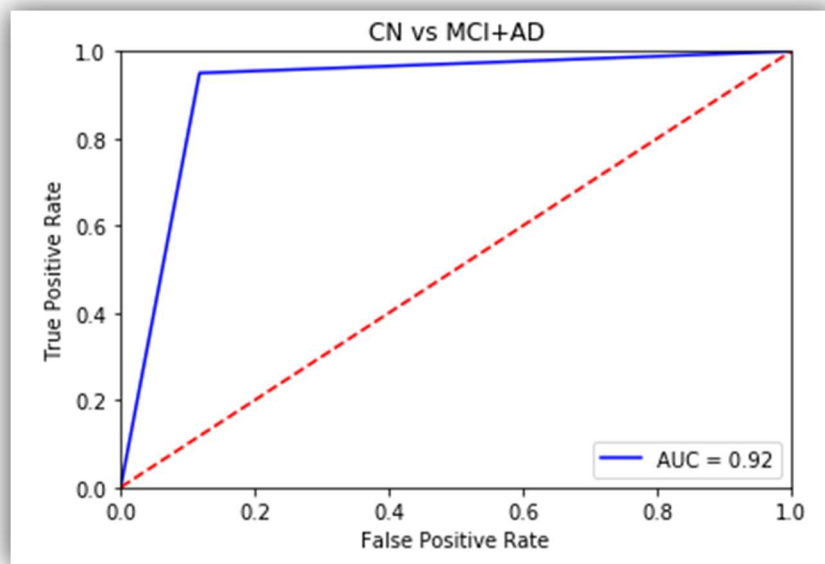


Figura 51. Clasificador multiclase en cascada. Curva ROC subconjunto de *test* CN vs MCI+AD.

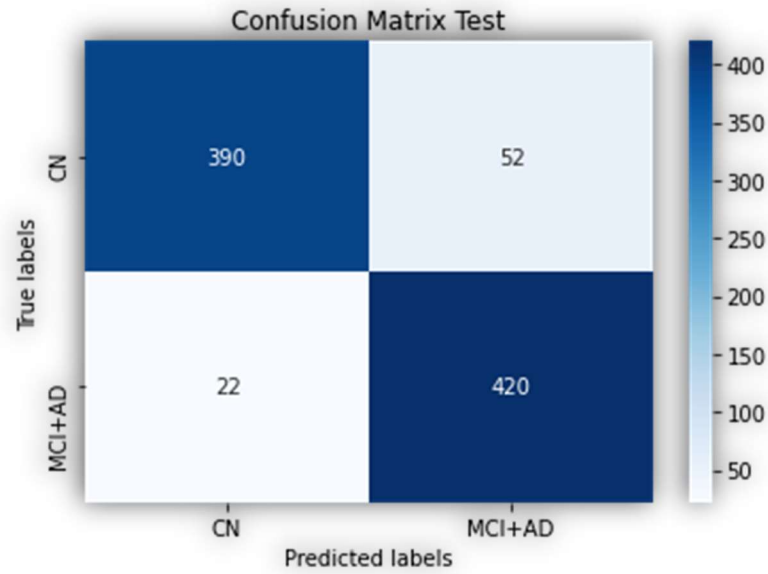


Figura 52. Clasificador multiclase en cascada. Matriz de confusión subconjunto de *test* CN vs MCI+AD.

- MCI vs CN+AD

Modelo de <i>test</i>	exactitud	precisión	sensibilidad	especificidad	f1 score	AUC
MCI vs CN+AD	0.899321	0.725979	0.944444	0.884731	0.820926	0.914587

Tabla 45. Clasificador multiclase en cascada. Métricas subconjunto de *test* MCI vs CN+AD.

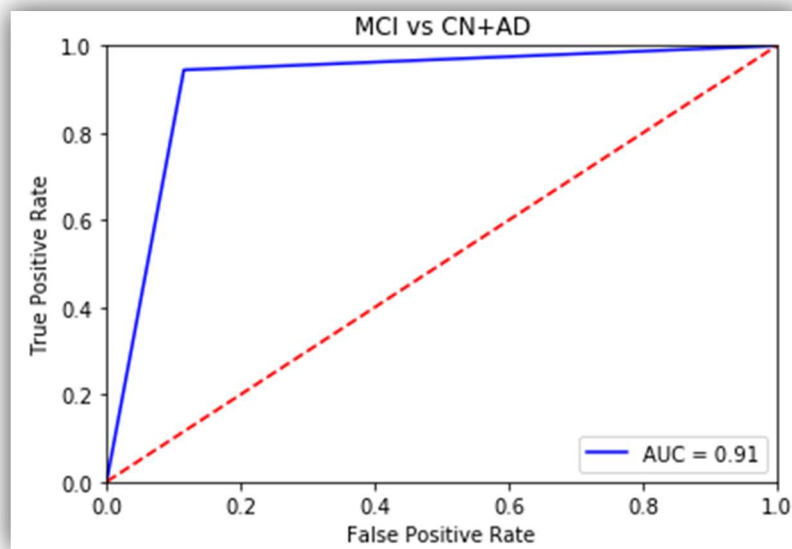


Figura 53. Clasificador multiclase en cascada. Curva ROC subconjunto de *test* MCI vs CN+AD.

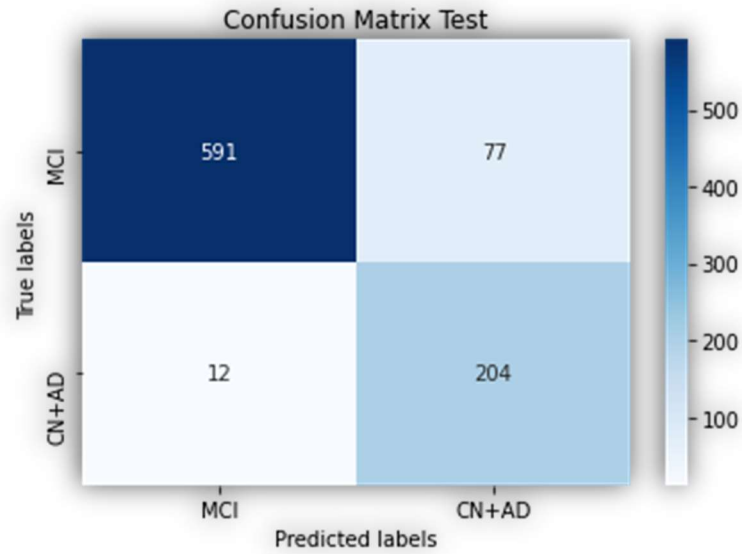


Figura 54. Clasificador multiclase en cascada. Matriz de confusión subconjunto de *test* MCI vs CN+AD.

- AD vs CN+MCI

Modelo de <i>test</i>	exactitud	precisión	sensibilidad	especificidad	f1 score	AUC
AD vs MCI+CN	0.911765	0.745583	0.97235	0.892054	0.844	0.932202

Tabla 46. Clasificador multiclase en cascada. Métricas subconjunto de *test* AD vs MCI+CN.

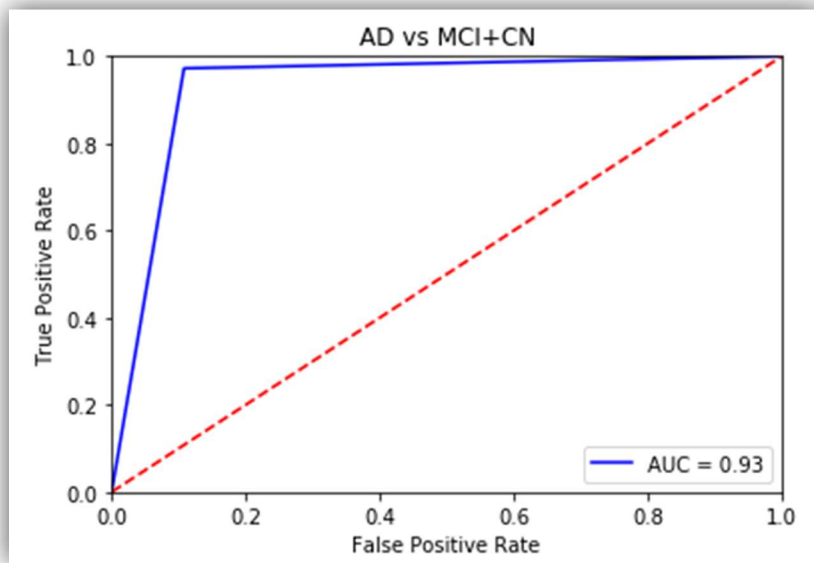


Figura 55. Clasificador multiclase en cascada. Curva ROC subconjunto de *test* AD vs MCI+CN.

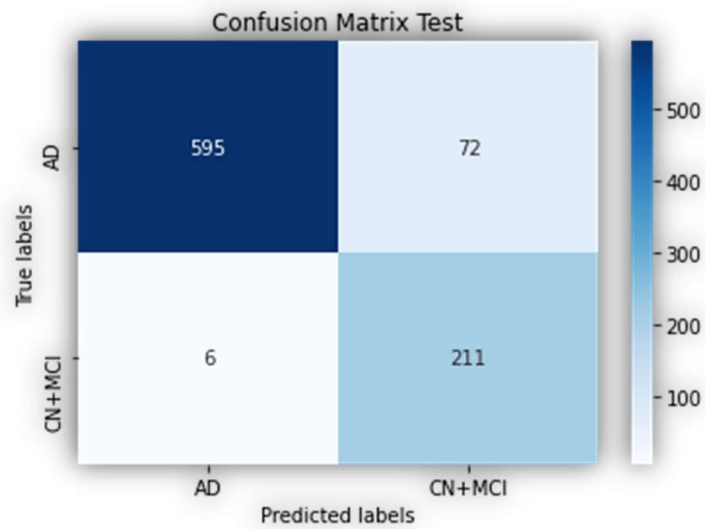


Figura 56. Clasificador multiclase en cascada. Matriz de confusión subconjunto de *test* AD vs MCI+CN.

CAPÍTULO 6. DISCUSIÓN.

Durante la realización de este proyecto se ha querido abordar el problema de la detección de Alzheimer mediante técnicas de IA desde un punto de vista bidimensional, tratando imágenes *.png* 2D. Para ello, se han realizado tanto clasificaciones binarias (patológico – no patológico) como multiclase (CN, MCI, AD).

En primer lugar, se realizaron varios experimentos previos (ver tabla 47) en los cuales se pudo comprobar que las técnicas de pre-procesado de *zero padding*, normalización de intensidad y eliminación de los cortes axiales sin información anatómica relevante, permitían obtener mejores resultados de clasificación. Sin embargo, la extracción de cráneo no obtenía mejoras. En cuanto al entrenamiento, la arquitectura CNN con la que se obtuvieron mejores resultados fue *DenseNet121*, con un tamaño de imagen de 128x128 y un *batch size* igual a 64.

Experimentos previos		Exactitud final
<i>Zero padding</i>	Con	0.827391
	Sin	0.739002
Extracción de cráneo	Con	0.702274
	Sin	0.739002
Tamaño de imagen	128x128	0.785013
	128x128 → 256x256	0.723261
Normalización de intensidad	Con	0.808422
	Sin	0.785013
Eliminación de cortes axiales con ausencia de tejido cerebral	Con	0.825578
	Sin	0.808422
Eliminación de cortes axiales con una media de intensidad inferior a 0.05	Con	0.836953
	Sin	0.825578
Eliminación de cortes axiales sin información anatómica relevante para la detección de AD	Con	0.862319
	Sin	0.836953
Elección de la arquitectura CNN	ResNet34	0.862319
	ResNet50	0.904106
	SqueezeNet1_1	0.757488
	DenseNet121	0.945169
Elección del <i>batch size</i>	64	0.945169
	32	0.933140
	128	0.846618

Tabla 47. Resumen de los valores de exactitud obtenidos para cada uno de los experimentos previos realizados.

En segundo lugar, y en cuanto a la clasificación binaria, se obtuvieron buenos resultados, con una exactitud del 90% y una AUC de 0.9. Además, se puede observar que el

valor de especificidad conseguido es mayor, aproximadamente en un 10%, al valor de sensibilidad (ver tabla 35). Este hecho es importante, ya que prueba que el clasificador tiene una mayor capacidad para identificar correctamente muestras patológicas que no patológicas.

Considerando la enfermedad de Alzheimer, es conveniente que las muestras patológicas sean detectadas en mayor medida que las no patológicas. Ahora bien, como se comentó en la introducción, el principal problema asociado a la AD es la detección precoz y, por tanto, sería preferible que el clasificador fuese capaz de identificar el deterioro cognitivo leve para poder así actuar lo más rápido posible.

Por consiguiente, se analizarán ahora dos clasificadores multiclase realizados, cuyos resultados se muestran de forma resumida en la tabla 48.

Modelo de test	clasificador	exactitud	precisión	sensibilidad	especificidad	f1 score	AUC
CN vs MCI+AD	Multiclase	0.935520	0.952941	0.916290	0.954751	0.934256	0.935520
	Multiclase con cascada	0.91629	0.889831	0.950226	0.882353	0.919037	0.91629
MCI vs CN+AD	Multiclase	0.906109	0.976127	0.832579	0.979638	0.898657	0.906109
	Multiclase con cascada	0.899321	0.725979	0.944444	0.884731	0.820926	0.914587
AD vs CN+MCI	Multiclase	0.951357	0.927195	0.979638	0.923077	0.952695	0.951357
	Multiclase con cascada	0.911765	0.745583	0.97235	0.892054	0.844	0.932202

Tabla 48. Resumen de las métricas obtenidas con los subconjuntos de datos de *test* para los clasificadores multiclase.

Como se puede observar, los valores de exactitud obtenidos son mayores para el caso del clasificador sin cascada para todos los modelos de *test*, lo cual indica que este clasificador es capaz de identificar un mayor número de muestras correctamente. Esto mismo ocurre para las métricas de precisión, especificidad y *f1 score*, pero no es así para el caso de la métrica de sensibilidad. En el caso del AUC, cabe destacar que solamente el valor obtenido para el modelo MCI vs CN+AD es ligeramente mayor en el caso del clasificador con cascada.

En un primer momento, puede parecernos grave que los valores de sensibilidad sean menores ya que estaría indicando que el clasificador es capaz de identificar correctamente en mayor medida muestras no patológicas que muestras patológicas. Sin embargo, el modelo ideal sería aquel que consiguiese un balance entre sensibilidad y especificidad, y para ello se cuenta con la métrica de *f1 score*, siendo ésta en todo caso superior para el clasificador multiclase sin cascada.

Además, otra muestra del buen funcionamiento del clasificador sin cascada son los valores del AUC, donde se puede observar que en todo caso son valores superiores a 0.9, mostrando que existe una probabilidad mayor de clasificar correctamente una muestra positiva elegida al azar que una negativa. Este valor es fundamental sobre todo a la hora de la detección de la clase *CN*, ya que es de especial importancia evitar que un paciente patológico sea identificado como no patológico. En la tabla 48 puede observarse que dicho valor es superior para el clasificador multiclase.

Finalmente, si valoramos las matrices de confusión obtenidas en el caso del clasificador multiclase (ver figuras 43, 45 y 47) y las comparamos con el clasificador multiclase en cascada (ver figuras 52, 54 y 56) se puede observar que el mayor número de errores se dan para la detección de MCI. En el clasificador multiclase en cascada (figura 54) ocurre debido al alto número de falsos negativos mientras que en el clasificador multiclase (figura 45) se debe al alto número de falsos positivos, siendo este caso preferible debido a la gravedad de la enfermedad y a la necesidad de actuar con la mayor brevedad posible.

CAPÍTULO 7. CONCLUSIONES Y LÍNEAS FUTURAS.

Durante la realización de este proyecto se ha podido comprobar, por una parte, la gran cantidad de técnicas de pre-procesado de imagen médica que son aplicables a las imágenes de resonancia magnética cerebral y, por otra parte, también la gran labor que el campo de la inteligencia artificial puede aportar a la detección precoz de la enfermedad de Alzheimer. Sin embargo, se trata de técnicas complejas y que requieren de un gran entendimiento del problema a abordar.

Los resultados obtenidos, muestran una nueva línea de investigación que implicaría el uso de imágenes bidimensionales en los clasificadores. Esto reduce en gran medida el coste computacional necesario obteniendo resultados alrededor del 90% de aciertos y que podrían mejorarse con experimentos futuros, siendo este valor de exactitud superior al 73% del único experimento con CNN bidimensionales encontrado en el estudio del estado del arte (ver capítulo 3). El principal problema de confusión en la clasificación de imágenes se da en aquellos sujetos que presentan un deterioro cognitivo leve.

Cabe destacar que un mayor número de observaciones, o en este caso imágenes, podrían mejorar los resultados ya que permiten conseguir un mejor ajuste de los pesos de las CNN. Del mismo modo, también podrían aplicarse técnicas de pre-procesado más complejas.

Aun así, sí es cierto que las redes CNN 3D consiguen valores de exactitud superiores en aproximadamente un 5-7%(ver capítulo 3). Esto muestra la necesidad de abordar el problema desde un punto de vista tridimensional, ya que el objetivo de estos clasificadores no debería ser clasificar imágenes bidimensionales si no clasificar a los propios sujetos.

Por tanto, una posible línea de estudio futura y que sí permitiría abordar el problema tridimensionalmente podría ser estudiar el número de cortes consecutivos y por sujeto que son clasificados como MCI o AD y, a partir de ello, establecer umbrales que permitan ajustar las clasificaciones. De esta forma, se conseguiría obtener una sola clasificación por sujeto en lugar de una clasificación por imagen.

Para finalizar este proyecto, se considera importante recordar que en ningún momento estas técnicas permiten reemplazar el trabajo que nuestros profesionales de la sanidad realizan día tras día, sino simplemente servirles de apoyo en la toma de decisiones. Tal y como muestra un titular del MIT: *“La inteligencia artificial no puede reemplazar a los doctores, pero puede hacerlos mejores profesionales”*. [3]

Bibliografía

- [1] A. Donoso, «La enfermedad de Alzheimer», *Rev. Chil. Neuro-Psiquiatr.*, vol. 41, pp. 13-22, nov. 2003, doi: 10.4067/S0717-92272003041200003.
- [2] E. N. Merino, M. A. C. Sendin, y J. A. V. Osorio, «Enfermedad de Alzheimer», *Med. - Programa Form. Médica Contin. Acreditado*, vol. 11, n.º 72, pp. 4306-4315, feb. 2015, doi: 10.1016/j.med.2015.01.002.
- [3] «AI can't replace doctors. But it can make them better.», *MIT Technology Review*. <https://www.technologyreview.com/2018/10/23/139414/ai-cant-replace-doctors-but-it-can-make-them-better/> (accedido may 09, 2020).
- [4] Elsevier, «La cura del Alzheimer, ¿por qué cada vez estamos más cerca?», *Elsevier Connect*. <https://www.elsevier.com/es-es/connect/ciencia/la-cura-del-alzheimer,-por-que-cada-vez-estamos-mas-cerca> (accedido may 09, 2020).
- [5] «Cambios del cerebro en la enfermedad del Alzheimer». <https://www.dementiacarecentral.com/caregiverinfo/cambios-del-cerebro-en-la-enfermedad-del-alzheimer/> (accedido may 13, 2020).
- [6] «Alzhéimer», *AFAGA*. <https://afaga.com/es/las-demencias/alzheimer/> (accedido may 14, 2020).
- [7] A. del P. Rueda-O., L. F. Enríquez-S., A. del P. Rueda-O., y L. F. Enríquez-S., «A REVIEW OF BASIC NEUROIMAGING TECHNIQUES IN DIAGNOSIS OF NEURODEGENERATIVE DISEASES», *Biosalud*, vol. 17, n.º 2, pp. 59-90, jul. 2018, doi: 10.17151/biosa.2018.17.2.5.
- [8] E. Arana Fernández de Moya, «Demencias e imagen: lo básico», *Radiología*, vol. 52, n.º 1, pp. 4-17, ene. 2010, doi: 10.1016/j.rx.2009.09.005.
- [9] Unknown, «EL BLOG DE CEES: Alzheimer: análisis 2D y 3D del hipocampo», *EL BLOG DE CEES*, feb. 19, 2019. <http://ceessblog.blogspot.com/2019/02/alzheimer-analisis-2d-y-3d-del-hipocampo.html> (accedido may 14, 2020).
- [10] J. C. Masdeu Puche, «La neuroimagen en la enfermedad de Alzheimer: perspectiva actual», *Rev. Neurol.*, vol. 38, n.º 12, p. 1156, 2004, doi: 10.33588/rn.3812.2004291.
- [11] D. I. Orellana *et al.*, «TRABAJOS EN CARTEL. XXXVI CURSO ANUAL DE RADIOLOGÍA SMRI», p. 5.
- [12] C. Acadia, «Las técnicas de neuroimagen», *Psicología y logopedia infantil, juvenil*, jul. 04, 2016. <https://www.centroacadia.es/tecnicas-neuroimagen/> (accedido may 15, 2020).

- [13] «Imagen por Resonancia Magnética (IRM)». <https://www.nibib.nih.gov/espanol/temas-cientificos/imagen-por-resonancia-magn%C3%A9tica-irm> (accedido may 15, 2020).
- [14] D. J. Gili, «RESONANCIA MAGNÉTICA EN NEUROIMAGEN», p. 229.
- [15] G. Fernández y L. Alberto, «Usos y aplicaciones de la inteligencia artificial», 2004. <https://cdigital.uv.mx/> (accedido may 09, 2020).
- [16] «Machine learning: conoce qué es y las diferencias entre sus tipos», *Think Big*, nov. 16, 2017. <https://empresas.blogthinkbig.com/que-algoritmo-elegir-en-ml-aprendizaje/> (accedido ago. 05, 2020).
- [17] «¿Qué es Aprendizaje profundo (deep learning)? - Definición en WhatIs.com», *SearchDataCenter en Español*. <https://searchdatacenter.techtarget.com/es/definicion/Aprendizaje-profundo-deep-learning> (accedido may 09, 2020).
- [18] «Artificial Intelligence | Machine learning | Deep learning», *Argility*. <https://www.argility.com/argility-ecosystem-solutions/iot/machine-learning-deep-learning/> (accedido ago. 05, 2020).
- [19] A. Delgado, «Aplicación de las Redes Neuronales en Medicina», *Rev. Fac. Med.*, vol. 47, n.º 4, pp. 221-223, oct. 1999.
- [20] «La neurona artificial». https://www.ibiblio.org/pub/linux/docs/LuCaS/Presentaciones/200304curso-galisa/redes_neuronales/curso-galisa-redes_neuronales-html/x38.html (accedido ago. 05, 2020).
- [21] C. J. Tablada y G. A. Torres, «Redes Neuronales Artificiales», *Rev. Educ. Matemática*, vol. 24, n.º 3, 2009, Accedido: ago. 05, 2020. [En línea]. Disponible en: <https://revistas.psi.unc.edu.ar/index.php/REM/article/view/10280>.
- [22] Na8, «¿Cómo funcionan las Convolutional Neural Networks? Visión por Ordenador», *Aprende Machine Learning*, nov. 29, 2018. <https://www.aprendemachinelarning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/> (accedido may 26, 2020).
- [23] D. Prakash, N. Madusanka, S. Bhattacharjee, H.-G. Park, C.-H. Kim, y H.-K. Choi, «A Comparative Study of Alzheimer’s Disease Classification using Multiple Transfer Learning Models», *Journal of Multiempodia Information System*, 2019. http://www.jmis.org/archive/view_article?pid=jmis-6-4-209 (accedido may 11, 2020).
- [24] «Deep Learning – Introducción práctica con Keras - Jordi TORRES.AI». <https://torres.ai/deep-learning-inteligencia-artificial-keras/> (accedido ago. 06, 2020).

- [25] «Convolutional Neural Networks: A Python Tutorial Using TensorFlow and Keras», *KDnuggets*. <https://www.kdnuggets.com/convolutional-neural-networks-a-python-tutorial-using-tensorflow-and-keras.html/> (accedido ago. 06, 2020).
- [26] G. Hilera, *Redes neuronales artificiales. Fundamentos, modelos y aplicaciones*. 1994.
- [27] «10 Advanced Deep Learning Architectures Data Scientists Should Know!», *Analytics Vidhya*, ago. 09, 2017. <https://www.analyticsvidhya.com/blog/2017/08/10-advanced-deep-learning-architectures-data-scientists/> (accedido may 11, 2020).
- [28] «5 Advanced CNN Architectures · Deep Learning for Vision Systems MEAP V08 livebook». <https://livebook.manning.com/book/grokking-deep-learning-for-computer-vision/chapter-5/v-3/> (accedido may 12, 2020).
- [29] S.-H. Tsang, «Review: VGGNet — 1st Runner-Up (Image Classification), Winner (Localization) in ILSVRC 2014», *Medium*, sep. 24, 2019. <https://medium.com/coinmonks/paper-review-of-vggnet-1st-runner-up-of-ilsvrc-2014-image-classification-d02355543a11> (accedido may 12, 2020).
- [30] P. Ruiz, «Understanding and visualizing ResNets», *Medium*, abr. 23, 2019. <https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8> (accedido may 26, 2020).
- [31] S. Ghassemi y E. Magli, «Convolutional Neural Networks for On-Board Cloud Screening», *Remote Sens.*, vol. 11, n.º 12, p. 1417, ene. 2019, doi: 10.3390/rs11121417.
- [32] M. Chablani, «DenseNet», *Medium*, ago. 25, 2017. <https://towardsdatascience.com/densenet-2810936aeebb> (accedido may 12, 2020).
- [33] K. Patel, «Architecture comparison of AlexNet, VGGNet, ResNet, Inception, DenseNet», *Medium*, mar. 08, 2020. <https://towardsdatascience.com/architecture-comparison-of-alexnet-vggnet-resnet-inception-densenet-beb8b116866d> (accedido ago. 06, 2020).
- [34] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, y K. Keutzer, «SqueezeNet: AlexNet-level exactitud with 50x fewer parameters and <0.5MB model size», *ArXiv160207360 Cs*, nov. 2016, Accedido: may 12, 2020. [En línea]. Disponible en: <http://arxiv.org/abs/1602.07360>.
- [35] «(a) Fire module in SqueezeNet; (b) SqueezeNet architecture (Figure from... | Download Scientific Diagram». https://www.researchgate.net/figure/a-Fire-module-in-SqueezeNet-b-SqueezeNet-architecture-Figure-from-7_fig2_326809561 (accedido ago. 06, 2020).
- [36] R. Parmar, «Common Loss functions in machine learning», *Medium*, sep. 02, 2018. <https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23> (accedido ago. 06, 2020).

- [37] A. Al-Masri, «How Does Back-Propagation in Artificial Neural Networks Work?», *Medium*, ene. 29, 2019. <https://towardsdatascience.com/how-does-back-propagation-in-artificial-neural-networks-work-c7cad873ea7> (accedido ago. 06, 2020).
- [38] B. Frades, A. Frank, L. Gangoiti, M. I. Genua, M. I. González, y A. Hueros, «Estado del arte de la enfermedad de Alzheimer en España», *Madr. Editor. Pwc*, 2013.
- [39] «World Alzheimer Report 2015, The Global Impact of Dementia: An analysis of prevalence, incidence, cost and trends», p. 87.
- [40] J. L. Fernández, C. Parapar, y M. Ruiz, «El envejecimiento de la población», *LYCHNOS Cuad. Fund. Gen. CSIC*, vol. 2, 2010.
- [41] «Cinco grandes necesidades que tienen los mayores de 50 años», *60 y mucho más*. <https://www.60ymuchomas.com/cinco-grandes-necesidades-que-tienen-los-mayores-de-50-anos/> (accedido ago. 05, 2020).
- [42] «El alzheimer en números Fundación CITA Alzheimer». <http://www.cita-alzheimer.org/la-enfermedad/el-alzheimer-en-numeros> (accedido ago. 05, 2020).
- [43] E. Hosseini-Asl, R. Keynton, y A. El-Baz, «Alzheimer's disease diagnostics by adaptation of 3D convolutional network», en *2016 IEEE International Conference on Image Processing (ICIP)*, sep. 2016, pp. 126-130, doi: 10.1109/ICIP.2016.7532332.
- [44] S. Korolev, A. Safiullin, M. Belyaev, y Y. Dodonova, «Residual and Plain Convolutional Neural Networks for 3D Brain MRI Classification», *ArXiv170106643 Cs*, ene. 2017, Accedido: jun. 08, 2020. [En línea]. Disponible en: <http://arxiv.org/abs/1701.06643>.
- [45] «(PDF) Convolutional Neural Networks for alzheimer disease classification from 2D brain images», *ResearchGate*. https://www.researchgate.net/publication/324747235_Convolutional_Neural_Networks_for_alzheimer_disease_classification_from_2D_brain_images (accedido jun. 08, 2020).
- [46] S. Chaplot, L. M. Patnaik, y N. R. Jagannathan, «Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network», *Biomed. Signal Process. Control*, vol. 1, n.º 1, pp. 86-92, ene. 2006, doi: 10.1016/j.bspc.2006.05.002.
- [47] R. C. Petersen *et al.*, «Alzheimer's Disease Neuroimaging Initiative (ADNI)», *Neurology*, vol. 74, n.º 3, pp. 201-209, ene. 2010, doi: 10.1212/WNL.0b013e3181cb3e25.
- [48] «Matlab - csic.es». <http://sitios.csic.es/web/calculo-cientifico/matlab> (accedido may 25, 2020).
- [49] «Image Processing Toolbox». <https://es.mathworks.com/products/image.html> (accedido may 25, 2020).

- [50] «SPM - Statistical Parametric Mapping». <https://www.fil.ion.ucl.ac.uk/spm/> (accedido may 09, 2020).
- [51] T. E. Oliphant, «Python for Scientific Computing», *Comput. Sci. Eng.*, vol. 9, n.º 3, pp. 10-20, may 2007, doi: 10.1109/MCSE.2007.58.
- [52] «▷ Top 10: Lenguajes de Programación más Usados en Todo el Mundo». <https://www.yeeply.com/blog/lenguajes-de-programacion-mas-usados/> (accedido may 09, 2020).
- [53] «Top Lenguajes de Programación para Inteligencia Artificial - Adictec», *Adictec - Adicción por la tecnología*, nov. 25, 2017. <https://adictec.com/lenguajes-programacion-para-inteligencia-artificial/> (accedido may 09, 2020).
- [54] J. Howard y S. Gugger, «Fastai: A Layered API for Deep Learning», *Information*, vol. 11, n.º 2, p. 108, feb. 2020, doi: 10.3390/info11020108.
- [55] J. Howard, *fastai: fastai makes deep learning with PyTorch faster, more accurate, and easier*.
- [56] «training | fastai». <https://docs.fast.ai/training.html> (accedido may 09, 2020).
- [57] «Atlases - FslWiki». <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/Atlases> (accedido may 13, 2020).
- [58] «MNI Space». <https://brainmap.org/training/BrettTransform.html> (accedido ago. 09, 2020).
- [59] «Deformation Field - an overview | ScienceDirect Topics». <https://www.sciencedirect.com/topics/computer-science/deformation-field> (accedido may 14, 2020).
- [60] A. Mishra, «Metrics to Evaluate your Machine Learning Algorithm», *Medium*, nov. 01, 2018. <https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234> (accedido may 13, 2020).
- [61] «Evaluación de modelos de clasificación: Matriz de Confusión y Curva ROC | Welcome to the Jungle». <http://www.ericmelillanca.cl/content/evaluaci-n-modelos-clasificaci-n-matriz-confusi-n-y-curva-roc> (accedido ago. 13, 2020).

PRESUPUESTO

Diseño e implementación de un clasificador para la detección de la enfermedad de Alzheimer mediante el uso de técnicas de inteligencia artificial en imágenes de resonancia magnética

Índice del presupuesto

1. Introducción.....	79
2. Contenido del presupuesto.....	80
2.1. Coste de la mano de obra.....	80
2.2. Coste de la maquinaria y software.....	80
2.3. Coste de diseño y desarrollo de la metodología.....	81
2.4. Coste de material fungible.....	82
3. Resumen del presupuesto.....	82

Índice de tablas

Tabla 1. Coste de la mano de obra.....	80
Tabla 2. Coste de la maquinaria y software.....	81
Tabla 3. Coste de diseño y desarrollo de la metodología.....	81
Tabla 4. Coste de material fungible.....	82
Tabla 5. Resumen del presupuesto.....	82

1. Introducción.

A continuación, se realizará una valoración económica de los costes asociados a la realización de este proyecto. Los precios indicados en el siguiente apartado pretenden ser orientativos según la información de la que se dispone considerando salarios base y los precios de venta actuales.

Cabe destacar que se ha definido como duración total del proyecto un periodo de 9 meses, teniendo en cuenta las reuniones iniciales de presentación del proyecto y la formación en inteligencia artificial y pre-procesado de imagen médica.

2. Contenido del presupuesto.

Para un estudio más detallado, se han dividido los costes en 4 grandes grupos. Por una parte, el coste de la mano de obra englobará los sueldos de todas aquellas personas que han participado en el proyecto; el coste de equipo y software, por su parte, implica toda la maquinaria y programas utilizados para el desarrollo del proyecto; el coste de material fungible es aquel asociado a materiales como folios que se eliminan una vez utilizados; por último, el coste de desarrollo de la metodología tiene en cuenta todos los gastos restantes como puede ser la búsqueda de información o la adquisición de datos.

2.1. Coste de la mano de obra.

Se han considerado 4 trabajadores implicados en el proyecto. Concretamente, los puestos asociados son: *profesor titular de universidad*, siendo este el propio tutor del proyecto; *ingeniero de I+D*, como primer cotutor asociado y especialista en pre-procesado de imagen médica; *científico de datos en inteligencia artificial*, como segundo cotutor y especializado en técnicas de inteligencia artificial; y, para finalizar, un *ingeniero biomédico junior* siendo éste el autor del trabajo final de grado.

Personal	Coste (€/h)	Tiempo (h)	Total (€)
Profesor titular de universidad (tutor)	35	10	350
Ingeniero de I+D (cotutor 1)	30	50	1.500
Científico de datos en inteligencia artificial (cotutor 2)	30	60	1.800
Ingeniero biomédico junior (autor)	12	360	4.320
Total			7.970

Tabla 1. Coste de la mano de obra.

2.2. Coste de la maquinaria y software.

El equipo principal para la realización de este proyecto ha sido un ordenador portátil *Lenovo U41-70 80JV* con ayuda de un *cluster* de GPUs. En cuanto a software utilizado se debe considerar por parte de pre-procesado MATLAB y SPM. En cuanto a inteligencia artificial y programación, se debe tener en cuenta Python y el entorno de desarrollo integrado PyCharm. Los precios detallados se muestran en la siguiente tabla.

Para el coste del ordenador portátil y del *cluster* se ha indicado el precio de venta actual. Para la licencia de MATLAB, por su parte, se ha tenido en cuenta el coste de licencia anual, mientras que el resto de los programas son de uso gratuito. Además, se ha añadido un factor de amortización calculado según el cociente entre la duración de uso y la vida útil estimada de cada producto en meses.

Concepto	Coste (€/ud)	Unidades (ud)	Factor de amortización	Total (€)
Lenovo U41-70 80JV	1.029	1	8/48	171,5
<i>Cluster</i> de GPUs	6.799	1	5/48	708,22
MATLAB	800	1	5/12	333,33
SPM12	0	1	5/12	0
Python	0	1	8/12	0
PyCharm	0	1	8/12	0
Total				1.213,05

Tabla 2. Coste de la maquinaria y software.

2.3. Coste de diseño y desarrollo de la metodología.

Como se ha comentado en apartados anteriores las imágenes de resonancia magnética cerebral empleadas fueron obtenidas a partir de la base de datos ADNI. Por tanto, los gastos derivados de la adquisición de datos serían nulos teniendo en cuenta que se trata de una base de datos gratuita una vez concedido el acceso a ella.

Por otro lado, sí se han consultado gran cantidad de artículos para la realización del proyecto, aproximadamente 40. Todos ellos se pudieron adquirir de forma gratuita gracias a las licencias concedidas por parte de la UPV a sus alumnos. Sin embargo, la mayoría eran de pago y, por tanto, se considerará un coste por artículo estándar de 35€.

Concepto	Coste (€/ud)	Unidades (ud)	Total (€)
Obtención de imágenes de RMN estructural cerebral	0	633	0
Consulta de artículos científicos	35	40	1.400
Total			1.400

Tabla 3. Coste de diseño y desarrollo de la metodología.

2.4. Coste de material fungible.

Como material fungible simplemente se ha utilizado material de papelería, entendiéndose por ello los costes derivados de folios, bolígrafos, subrayadores, impresiones y encuadernaciones.

Concepto	Coste (€/ud)	Unidades (ud)	Total (€)
Bolígrafo	0,3	2	0,6
Folio A4 80g/m2	0,006	500	3
Subrayador	0,7	3	2,1
Impresión + encuadernación del proyecto	23	2	46
Total			51,7

Tabla 4. Coste de material fungible.

3. Resumen del presupuesto.

Concepto	Total (€)
Coste de la mano de obra	7.970
Coste de la maquinaria y software	1.213,05
Coste de diseño y desarrollo de la metodología	1.400
Coste de material fungible	51,7
Total TFG	10.634,75

Tabla 5. Resumen del presupuesto.

Según este presupuesto el precio total derivado del desarrollo del proyecto ascendería a **diez mil seiscientos treinta y cuatro euros con setenta y cinco céntimos** .