

Trabajo Fin de Grado

Predicción del diagnóstico de la enfermedad de
Alzheimer mediante deep-learning en imágenes
18F-FDG PET

Diagnosis prediction of Alzheimer's disease with
deep-learning on 18 F-FDG PET images

Autor

David Solanas Sanz

Directoras

Mónica Hernández Giménez
Elvira Mayordomo Cámara

RESUMEN

Predicción del diagnóstico de la enfermedad de Alzheimer mediante deep-learning en imágenes ^{18}F -FDG PET

La enfermedad de Alzheimer es una enfermedad neurodegenerativa que afecta a más de 50 millones de personas en todo el mundo. Es la forma más común de demencia, con un 60-70% de los casos. Actualmente no existe una cura efectiva para ella, aunque sí existen algunos tratamientos que pueden ser eficaces si se aplican en las fases tempranas de la enfermedad, permitiendo retrasar su evolución. Por ello, un diagnóstico preciso y con suficiente antelación es fundamental para poder tomar medidas preventivas. El gran auge del deep-learning en los últimos años ha permitido el desarrollo de diferentes sistemas de predicción que ayuden al diagnóstico de la enfermedad de Alzheimer a partir de imágenes cerebrales.

El principal objetivo de este Trabajo de Fin de Grado es el desarrollo de un sistema de aprendizaje profundo basado en redes neuronales convolucionales que, a partir de imágenes ^{18}F -FDG PET del cerebro sea capaz de predecir el diagnóstico final entre pacientes enfermos (AD), con deterioro cognitivo leve (MCI) o cognitivamente normales (CN). La obtención de las imágenes para el entrenamiento y test de la red se han obtenido del repositorio de la *Alzheimer's Disease Neuroimaging Initiative* (ADNI).

Se han desarrollado dos sistemas con dos arquitecturas diferentes: la original propuesta en [1] y una mejora posterior de la misma propuesta en la literatura en un contexto diferente. Las imágenes utilizadas son 3D mientras que las arquitecturas utilizadas se basan en convoluciones 2D. Por este motivo, las imágenes de ^{18}F -FDG PET han sido preprocesadas antes de ser cargadas en la red. Para el entrenamiento de los sistemas se ha hecho uso de las técnicas de transfer-learning y fine-tuning. La implementación del sistema y el preprocesado de las imágenes se ha realizado en Python 3.6.9, mediante el uso de las librerías de Keras (versión 2.2.4) y TensorFlow (versión 1.12.0). El entrenamiento y test de la red se ha realizado sobre una tarjeta gráfica Titan RTX de 24 GBs de VRAM.

Los experimentos realizados muestran que, ambos sistemas desarrollados pueden llegar a predecir AD hasta 66 meses (5 años y medio) antes del diagnóstico final. El sistema basado en la arquitectura propuesta en [1] es capaz de predecir el diagnóstico final de Alzheimer con una precisión del 77.0% y un AUC de 0.84. Se ha encontrado que el sistema entrenado con los pacientes de AD y CN es capaz de diagnosticar la enfermedad con una precisión del 87.5% y un AUC de 0.97 y se ha analizado cómo afecta en el rendimiento del sistema la introducción de datos de pacientes con MCI. Con la arquitectura más moderna se ha conseguido mejorar los resultados con una precisión de 84.6% y un AUC de 0.89 en la predicción del diagnóstico final de Alzheimer. Finalmente, se han realizado distintos análisis de las redes neuronales convolucionales desarrolladas para comprender los puntos fuertes y débiles de los modelos obtenidos.

GLOSARIO

A continuación se presenta un glosario con las siglas que se utilizan en este trabajo:

- AD: *Alzheimer's Disease.*
- MCI: *Mild Cognitive Impairment.*
- CN: *Cognitive Normal.*
- ADNI: *Alzheimer's Disease Neuroimaging Initiative.*
- MRI: *Magnetic Resonance Imaging.*
- fMRI: *Functional magnetic resonance imaging*
- DTI: *Diffusion Tensor Imaging*
- PET: *Positron Emission Tomography.*
- ¹⁸F: *Fluorine 18*
- FDG: *Fluorodeoxyglucose.*
- ROC: *Receiver Operating Characteristic.*
- AUC: *Area Under the ROC Curve.*
- t-SNE: *t-distributed Stochastic Neighbor Embedding.*
- PCA: *Principal Component Analysis.*
- CNN: *Convolutional Neural Network.*
- RNN: *Recurrent Neural Network.*
- FC: *Fully Connected.*
- SVM: *Support Vector Machine*
- GNB: *Gaussian Naive Bayes*
- MI: *Mutual Information*
- FFT: *Fast Fourier Transform*
- LRP: *Layerwise Relevance Propagation*
- GPU: *Graphics Processing Unit*
- VRAM: *Video Random Access Memory*

ÍNDICE DE CONTENIDO COMPLETO

INTRODUCCIÓN	4
Motivación y contexto	4
Estado del arte	7
Objetivos	9
Organización de la memoria	9
MÉTODO DESARROLLADO	10
Redes neuronales convolucionales	10
InceptionV3	10
InceptionResNetV2	13
Transfer-learning y fine-tuning	14
Sistemas desarrollados	14
Datos utilizados	16
Preprocesado de las imágenes	17
Entrenamiento de la red	18
Detalles de la implementación	19
RESULTADOS	21
Arquitectura InceptionV3	21
Arquitectura InceptionResNetV2	25
Sistema de predicción AD vs CN	27
CONCLUSIONES	31
ANEXO A: Fundamentos de Deep-learning	34
ANEXO B: Métricas de análisis	42
ANEXO C: Proyecto ADNI	43
ANEXO D: PET en ADNI2	44
BIBLIOGRAFÍA	49

1. INTRODUCCIÓN

1.1. Motivación y contexto

La enfermedad de Alzheimer (Alzheimer's Disease, AD) es una enfermedad neurodegenerativa que afecta a más de 50 millones de personas en todo el mundo. Es la forma más común de demencia, con un 60-70% de los casos, y cada año se registran cerca de 10 millones de nuevos casos [2]. Esta enfermedad se caracteriza por un deterioro cognitivo progresivo. Los principales síntomas son la pérdida de memoria, dificultad en el lenguaje, trastornos conductuales, y por último la dificultad e incluso imposibilidad de realizar tareas del día a día debido a la pérdida de funciones biológicas, produciendo finalmente la muerte. Las causas de la enfermedad de Alzheimer son desconocidas, y no existe una cura efectiva a esta enfermedad [3]. Sí existen ciertos tratamientos que pueden ralentizar la evolución de la enfermedad, siendo más efectivos en las fases tempranas de la enfermedad.

El diagnóstico temprano de la enfermedad de Alzheimer está muchas veces relacionado con la detección de un deterioro cognitivo leve en el paciente (Mild Cognitive Impairment, MCI). Ésta es en muchas ocasiones una fase prodrómica del Alzheimer [4]. Hay un gran riesgo de que el deterioro cognitivo leve evolucione a Alzheimer con el paso del tiempo, por ello, es de vital importancia un diagnóstico precoz y preciso para poder tomar medidas preventivas. Por estas razones, existe un gran interés en el desarrollo de sistemas predictivos y búsqueda de biomarcadores que sean de utilidad en el diagnóstico temprano de la enfermedad. Los biomarcadores pueden ser de distintas modalidades: existen biomarcadores clínicos, genéticos, de imagen médica y otros.

Actualmente, los sistemas de inteligencia artificial basados en redes convolucionales profundas (CNNs) se están convirtiendo en los métodos más efectivos para la ayuda al diagnóstico y predicción de la enfermedad [1,33,34,35,36]. Los distintos sistemas de aprendizaje utilizan muy diferentes biomarcadores de imagen médica. Entre las modalidades de imagen más relevantes se encuentran las siguientes:

- Imágenes por resonancia magnética (MRI): Se trata de una técnica no invasiva para generar imágenes de los órganos del cuerpo a través de fuertes campos magnéticos. Estas imágenes permiten visualizar la concentración de líquido cefalorraquídeo, materia gris y materia blanca del cerebro proporcionando información anatómica de las diferentes estructuras cerebrales [5]. En la figura 1.1 se muestra un ejemplo de estas imágenes.

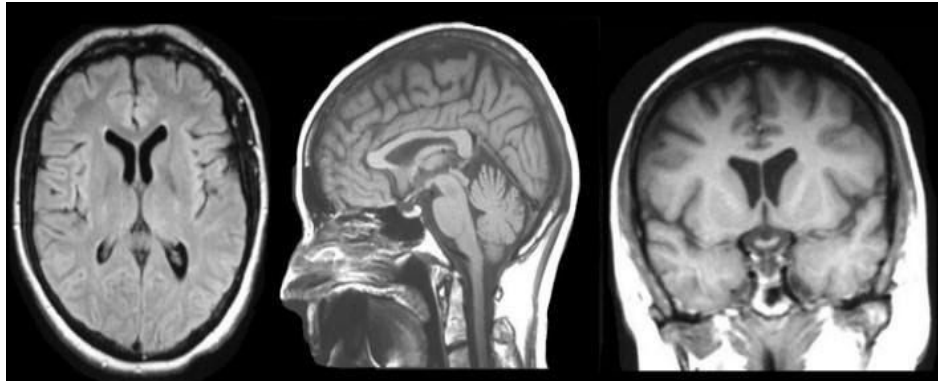


Figura 1.1: Imagen por resonancia magnética. De izquierda a derecha: Vistas axial, sagital y coronal de una imagen 3D. Imagen obtenida de: <https://case.edu/med/neurology/NR/MRI%20Basics.htm>

- Imagen por resonancia magnética funcional (fMRI): Técnica no invasiva que mide la actividad cerebral al detectar cambios asociados con el flujo sanguíneo. Se utiliza para mapear la actividad neuronal en el cerebro o en la médula espinal [17]. En la figura 1.2 se muestra un ejemplo de estas imágenes.

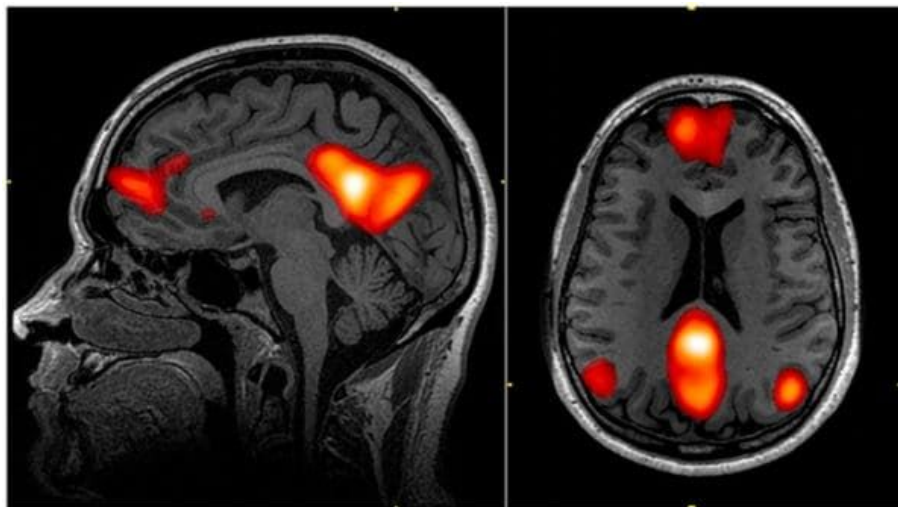


Figura 1.2: Imagen por resonancia magnética funcional. De izquierda a derecha: Vista sagital y axial de una imagen 3D. Imagen obtenida de: <http://www.cantechletter.com/2016/07/study-finds-software-glitch-40000-fmri-results/>

- Imágenes de tensor de difusión (DTI): Esta técnica es una versión mejorada de la resonancia magnética convencional en la que las señales se generan a partir del movimiento libre de las moléculas de agua en los tejidos. Con esta técnica se puede visualizar claramente la característica anisotrópica de las fibras de materia blanca, que permiten estudiar las conexiones macroscópicas de las regiones del cerebro. DTI es actualmente el único método basado en imágenes médicas que puede mostrar de forma no invasiva estructuras de fibras internas en el cerebro [16]. En la figura 1.3 se muestra un ejemplo de estas imágenes.

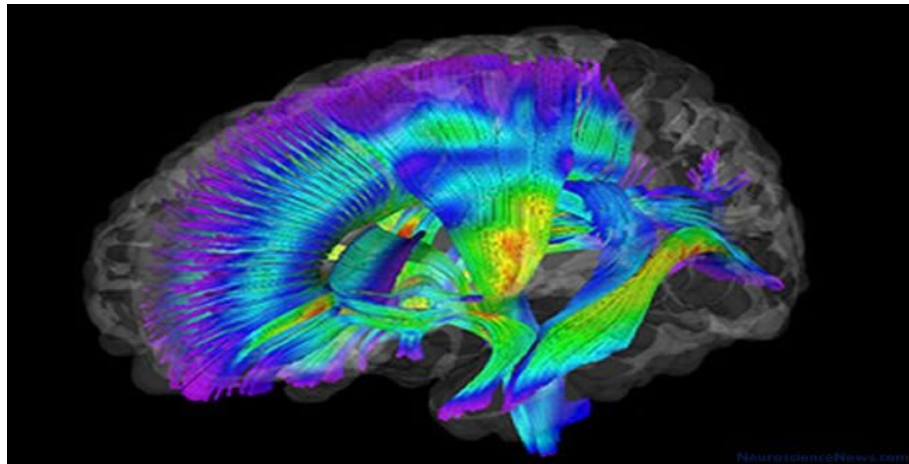


Figura 1.3: Imagen de tensor de difusión. Vista sagital de una imagen 3D. Superpuestas sobre la imagen pueden apreciarse una visualización de las fibras de la materia blanca. Imagen obtenida de: <https://nciceneews.com/brain-scans-6-months-infants-autism-diffusion-tensor-imaging-dti/>

- Tomografía por emisión de positrones (PET): Técnica que utiliza un radiofármaco para visualizar y medir la actividad metabólica del cuerpo humano. El radiofármaco es introducido por inyección intravenosa y, según que se desee estudiar, se usan unos radiofármacos u otros [6]. Los más relevantes para la enfermedad de Alzheimer son FDG-PET, AV-45 PET y PIB-PET (el principio activo que buscan estos PET es la glucosa en el caso de FDG-PET y el péptido β -amiloide en AV-45 PET y PIB-PET). En la figura 1.4 se muestra un ejemplo de diferentes imágenes de PET-Tomografía Computarizada (PET-CT) obtenidas de diferentes individuos sanos y enfermos.

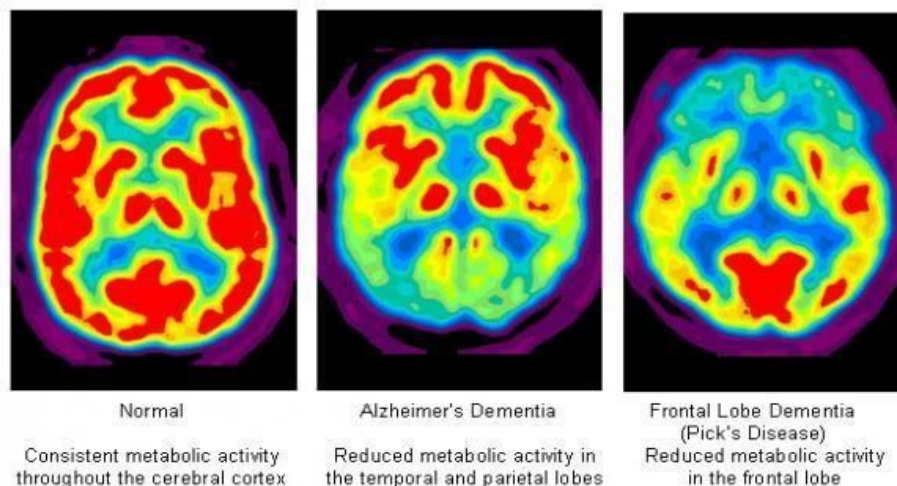


Figura 1.4: Tomografía por emisión de positrones-tomografía computarizada (PET-CT). De izquierda a derecha: Escáner de un paciente normal (CN), un paciente con Alzheimer (AD) y otro con una demencia del lóbulo frontal. Imagen obtenida de: <https://norcalscans.org/index.php/for-physicians/clinical-info-2/brain-disorders/>

Este Trabajo de Final de Grado (TFG) se centra en la utilización de imágenes PET como biomarcadores, más concretamente imágenes ^{18}F -FDG PET. Éstas se obtienen al utilizar como radiofármaco el Flúor-18, que es capaz de unirse a la 2-O-trifluorometilsulfonilmanosa para obtener el trazador Fluorodesoxiglucosa (FDG). De este modo, se puede identificar, localizar y cuantificar el consumo de glucosa en el cerebro [6]. Este tipo de imágenes han revelado una reducción en el transporte y la utilización de la glucosa cerebral en regiones concretas de pacientes con Alzheimer [7]. Estos cambios se producen en etapas tempranas de la enfermedad y también están presentes en pacientes cognitivamente normales con un riesgo genético de padecer la enfermedad de Alzheimer [7]. Es importante remarcar que las imágenes PET de amiloide permiten visualizar las placas de la molécula beta amiloide en el tejido neuronal dañado, por lo que, en principio, constituyen un mejor biomarcador para el diagnóstico de la enfermedad. No obstante, las imágenes ^{18}F -FDG PET permiten una ayuda al diagnóstico más temprana, lo que centra el interés de este TFG.

El objetivo de este TFG es el diseño de un sistema de aprendizaje basado en técnicas de deep-learning para el diagnóstico temprano de individuos enfermos, con deterioro cognitivo leve o cognitivamente normales a partir de imágenes ^{18}F -FDG PET. Para ello se ha desarrollado un sistema de predicción basado en redes neuronales convolucionales profundas, siguiendo la arquitectura propuesta en [1], entrenado y validado con 2092 imágenes ^{18}F -FDG PET de 831 pacientes. Estas imágenes han sido obtenidas de la base de datos ADNI.

1.2. Estado del arte

En la última década, los algoritmos de aprendizaje han tenido un gran impacto en todos los ámbitos del análisis de imágenes médicas y la ayuda al diagnóstico por imagen. En particular, se han desarrollado múltiples sistemas de clasificación con el objetivo de resolver el problema de la asistencia al diagnóstico de la enfermedad de Alzheimer en fases tempranas y lograr predecir la evolución de pacientes con deterioro cognitivo leve a Alzheimer. Los sistemas desarrollados utilizan distintas técnicas de aprendizaje automático con distintos biomarcadores para lograr resolver dicho problema. Los biomarcadores más utilizados son las imágenes de MRI y PET, aunque también se utilizan otros biomarcadores como los resultados de test cognitivos realizados a los pacientes. Se describen a continuación las técnicas y resultados obtenidos de los sistemas de predicción que forman parte del estado del arte.

- En [8] se aborda el problema predecir la conversión de MCI a AD con suficiente antelación, discriminando aquellos pacientes diagnosticados como MCI que evolucionan a AD con aquellos que se mantienen en MCI. En este trabajo, únicamente se utilizan imágenes de pacientes MCI y de aquellos que tras ser diagnosticados como MCI evolucionan a AD. Para ello se construye un modelo de clasificación basado en las intensidades de vóxeles de los volúmenes FDG-PET preprocesados. Se aplica el criterio de Mutual Information (MI) para reducir la dimensionalidad, y se utilizan dos clasificadores: Máquinas de Soporte Vectorial (SVM) y Clasificador de Bayes ingenuo gaussiano (Gaussian Naive Bayes, GNB). Con un tiempo de 24 meses previos al momento de

conversión, el clasificador obtiene un 70% de precisión a la hora de identificar pacientes que evolucionarán a AD. Conforme aumenta el tiempo de conversión, la precisión del modelo va disminuyendo.

- En [9] se vuelve a plantear el problema de predecir la conversión de MCI a AD a partir de imágenes FDG-PET. Las imágenes utilizadas son de pacientes MCI que evolucionan a AD (12.2 meses después del primer diagnóstico de media) y de pacientes MCI que no degeneran en AD. El modelo utilizado tiene un clasificador SVM en la capa final. Este modelo aprende las características de las regiones de interés del cerebro previamente extraídas. La mejor precisión obtenida en la clasificación es de un 86.6% utilizando un kernel con una función de base radial para el clasificador SVM.
- En [10] se plantea el problema de clasificar AD y MCI frente a CN. Para ello, se extraen 116 regiones de interés de las imágenes FDG-PET a partir de un atlas de etiquetado anatómico automatizado (Automated Anatomical Labeling, AAL). Con esas regiones de interés se extraen ciertas características y se aplica el método de análisis LASSO en cada set de características extraídas para seleccionar las mejores. Múltiples clasificadores SVM son entrenados, cada SVM utiliza un set concreto de características, y finalmente la predicción se hace a partir del voto mayoritario de dichos clasificadores. Con este método se obtiene una precisión de 90.48% en la clasificación de AD frente CN, y una precisión de 81.09% en la clasificación de MCI frente CN.
- En [11] se utiliza una red neuronal convolucional profunda junto con una red neuronal recurrente para la predicción de Alzheimer y MCI frente pacientes cognitivamente normales. Se utilizan imágenes FDG-PET, las cuales son preprocesadas y descompuestas en diversas secciones 2D para cada dirección de las imágenes (coronal, sagital y axial). Estas secciones son cargadas en la red convolucional que se encarga de identificar y aprender sus características más relevantes que serán pasadas como entrada a la red neuronal recurrente. Finalmente, hay dos capas completamente conectadas (Fully Connected Layer, FC) y una capa Softmax que se encargan de la predicción. La precisión obtenida es de 91.2% en la clasificación de AD vs CN y un 78.9% en la clasificación de MCI vs CN.
- En [12] se plantea de nuevo el problema de predecir la conversión de MCI a AD, se utilizan cuatro redes neuronales recurrentes y cuatro biomarcadores (rendimiento cognitivo, información demográfica, imágenes MRI y muestras de fluido cerebrospinal), cada red aprende a partir de un biomarcador concreto, se concatenan los vectores obtenidos de cada red y se aplica una regresión logística con regularización L1 para la predicción. Con este método se ha obtenido un 80% de precisión en la clasificación de AD con 24 meses previos al momento de conversión.
- En [1], se propone la predicción del diagnóstico final de un paciente, diferenciando AD vs MCI vs CN conjuntamente, distinguiéndose de los otros trabajos que hacen una predicción AD vs CN y MCI vs CN o simplemente AD vs MCI. Cabe destacar que el problema de la predicción conjunta de las tres clases

tiene una mayor dificultad que el problema de predicción por pares (AD vs CN o MCI vs CN). Se utiliza un modelo basado en redes neuronales convolucionales, más concretamente la red InceptionV3, que toma como datos de entrada imágenes FDG-PET provenientes de pacientes de los tres grupos. La arquitectura presentada en este trabajo es más sencilla que la de las RNN utilizadas en los otros trabajos. Con este método se ha obtenido un 76% de precisión en la predicción de AD con un tiempo medio de 75.8 meses (unos 6 años) de antelación, muy superior en comparación con el resto de clasificadores que forman parte del estado del arte.

En este Trabajo de Fin de Grado se propone abordar el problema de la predicción temprana de la enfermedad de Alzheimer, siguiendo el método propuesto en [1]. Se ha elegido este método debido a su capacidad de predicción temprana del diagnóstico y nuestro interés en los biomarcadores ^{18}F -FDG PET y en la metodología utilizada.

1.3. Objetivos

Los principales objetivos a conseguir con el desarrollo de este TFG son:

- Desarrollar un sistema de aprendizaje profundo a partir de imágenes ^{18}F -FDG PET capaz de predecir el diagnóstico final de un paciente, basado en la arquitectura propuesta en [1]. La implementación será desarrollada en Python 3.6.9, utilizando las bibliotecas de Tensorflow y Keras para desarrollar la red, y Scipy, scikit-image, Nibabel y Nilearn para el preprocesamiento de las imágenes.
- Reproducir los resultados obtenidos en [1], y estudiar la capacidad de predicción del modelo en pacientes que evolucionan a AD a partir de MCI o CN.
- Evaluar el modelo obtenido a partir de distintas técnicas de análisis para redes neuronales.
- Proponer distintas mejoras para conseguir un mejor sistema de predicción.

1.4. Organización de la memoria

La memoria se ha organizado en 4 secciones principales. En la sección 1 se presenta el contexto, el estado del arte y los objetivos que se pretenden conseguir en este TFG.

En la sección 2 se presentan las arquitecturas de InceptionV3 y InceptionResNetV2 y se explican los fundamentos de las técnicas de transfer-learning y fine-tuning. También se presentan los dos sistemas implementados, los datos utilizados, el preprocesamiento realizado a las imágenes PET, los parámetros utilizados para el entrenamiento y los aspectos técnicos más importantes de la implementación.

En la sección 3 se detallan los resultados obtenidos con los sistemas desarrollados y se presenta un sistema de predicción de AD vs CN junto con un análisis de cómo afecta al rendimiento del modelo al introducir la clase MCIs.

En la sección 4 se detallan las conclusiones obtenidas de este proyecto.

2. MÉTODO DESARROLLADO

En esta sección se presentan las arquitecturas de InceptionV3 y InceptionResNetV2, dos redes neuronales profundas muy utilizadas en la clasificación y detección de objetos en imágenes. Se explican también los fundamentos de las técnicas de transfer-learning y fine-tuning utilizadas en el desarrollo de los sistemas de predicción. Seguidamente, se presentan los dos sistemas implementados, los datos utilizados, el preprocesamiento realizado a las imágenes PET, los parámetros utilizados para el entrenamiento y los aspectos técnicos más importantes de la implementación. En el anexo A se pueden encontrar más detalles acerca de deep-learning y detalles del funcionamiento de las capas más importantes de las arquitecturas.

2.1. Redes neuronales convolucionales

Una red neuronal convolucional (convolutional neural network, CNN) es un tipo concreto de red neuronal diseñado para trabajar con datos de dos o más dimensiones, por lo general imágenes. Principalmente se utilizan para resolver problemas relacionados con la clasificación de imágenes, la clusterización de imágenes similares o el reconocimiento de objetos en una escena, entre otros. El auge de estas redes en los últimos años ha hecho que sean utilizadas en múltiples disciplinas.

2.1.1. InceptionV3

InceptionV3 es una red neuronal convolucional profunda desarrollada por Google. Esta red ha sido entrenada con ImageNet, un conjunto de datos con 14 millones de imágenes de 1000 clases distintas. El objetivo principal de esta red es reducir el coste computacional del sistema mediante la modificación de las arquitecturas Inception anteriores. En comparación con otras redes neuronales convolucionales profundas como VGGNet [20], se ha demostrado que InceptionV3 es más eficiente tanto en el número de parámetros de la red como en los recursos computacionales necesarios.

La arquitectura de la red se construye de forma progresiva, y hace uso de técnicas como convoluciones factorizadas, regularización, reducción de la dimensionalidad y computaciones paralelizadas [19]. A continuación se muestran los elementos principales de esta arquitectura.

- Convoluciones factorizadas: este tipo de convoluciones ayudan a reducir el coste computacional para imágenes con más de un canal (por ejemplo, imágenes RGB) [21]. En las capas convolucionales tradicionales, cada canal de salida se produce mediante la convolución de un conjunto de filtros con los datos de entrada a lo largo de las dimensiones espaciales. Esto se puede considerar como una combinación de convolución dentro de cada canal y proyección entre canales, lo que conlleva una gran cantidad de cálculos. Con las

convoluciones factorizadas se descompone la convolución y la proyección mediante la introducción de bases convolucionales. Cada canal se convoluciona primero con sus bases correspondientes, lo que da como resultado datos con la misma dimensión espacial y un mayor número de canales, que luego se proyectan a los canales de salida. En la figura 2.1 se muestra un ejemplo gráfico de esta operación.

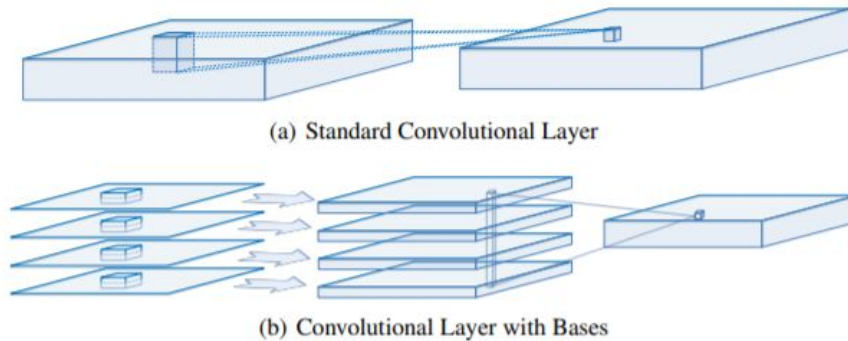


Figura 2.1: Ejemplo de convoluciones factorizadas, arriba (a) un ejemplo de una convolución tradicional, abajo (b) la operación equivalente utilizando convoluciones factorizadas. Imagen obtenida de [21].

- Convoluciones con filtros de menor tamaño: Utilizar convoluciones con un kernel pequeño en lugar de grande acelera el proceso de entrenamiento, un filtro 5x5 tiene 25 parámetros mientras que dos filtros 3x3 tienen 18.
- Convoluciones asimétricas: Consiste en separar una convolución con un kernel 3x3 por dos convoluciones, una con un kernel 1x3 seguida de otra con un kernel 3x1. En la figura 2.2 se puede ver un ejemplo de aplicar convoluciones asimétricas y convoluciones de menor tamaño.

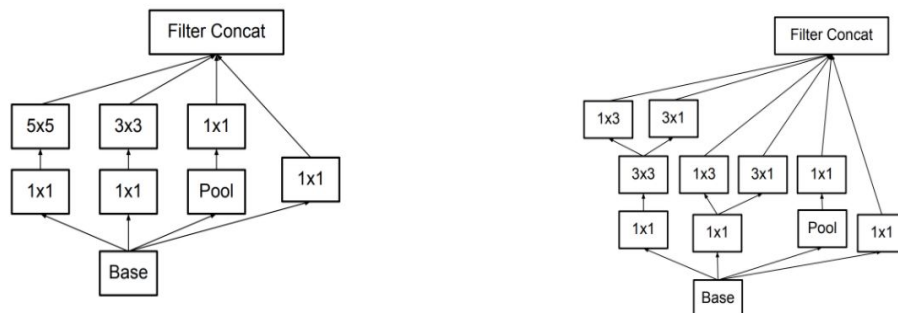


Figura 2.2: Ejemplo de la arquitectura de InceptionV3 al aplicar convoluciones asimétricas y kernels de menor tamaño. A la izquierda se encuentra la arquitectura original y a la derecha la arquitectura resultante. Imagen obtenida de [19].

- Clasificador auxiliar: Un clasificador auxiliar es una pequeña red neuronal convolucional insertada entre las capas ocultas durante el entrenamiento. En InceptionV3, este clasificador auxiliar actúa como regularizador. En la figura 2.3 se puede observar la arquitectura de un clasificador auxiliar.

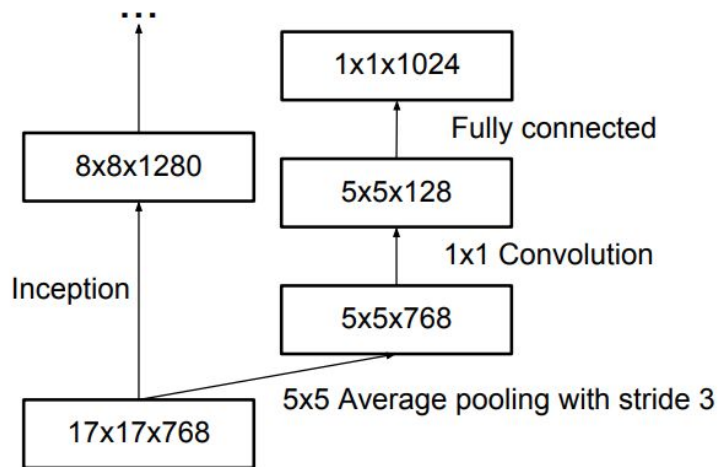


Figura 2.3: Clasificador auxiliar utilizado en InceptionV3 encima de la última capa 17x17. Imagen obtenida de [19].

- Reducción del tamaño de los datos: Esto se suele hacer con operaciones de pooling de forma eficiente.

Combinando todos estos elementos se ha construido InceptionV3. La arquitectura general de la red se muestra en la figura 2.4.

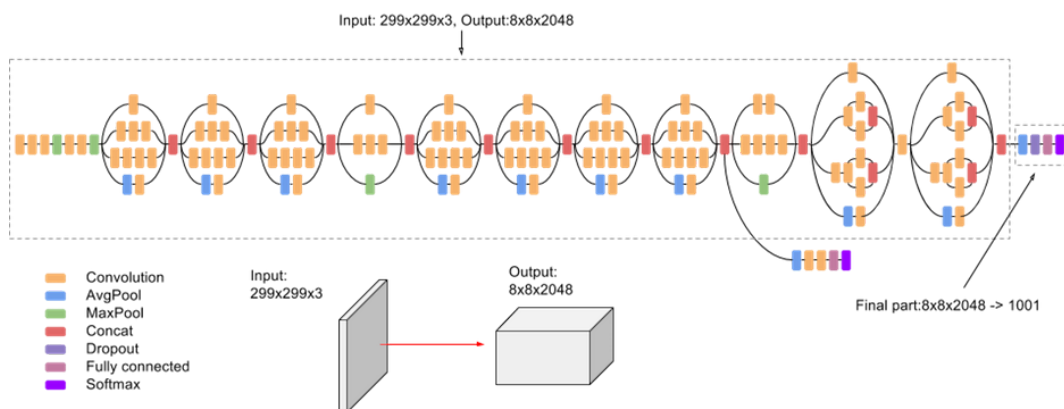


Figura 2.4: Arquitectura general de la red InceptionV3. Imagen obtenida de: <https://cloud.google.com/tpu/docs/inception-v3-advanced?hl=es-419>

Al realizar la evaluación sobre el conjunto de datos de ImageNet para establecer el benchmark de esta red, los resultados obtenidos son de una precisión de 77.9% y una precisión entre los 5 primeros (se da la predicción correcta si entre las 5 clases con más probabilidad que predice el modelo se encuentra la clase real) de un 93.7%. Estos resultados son mejores que los de otras redes como VGGNet, que consiguió una precisión de 71.3% y una precisión entre los 5 primeros de 90.1%. Otras redes como ResNet50 o ResNet101 consiguieron una precisión de 74.9 % y 76.4% respectivamente. Cabe destacar que el número de parámetros utilizados en InceptionV3 es del orden de 100 millones menor que VGGNet, lo que la hace una red mucho más potente y rápida.

2.1.2. InceptionResNetV2

InceptionResNetV2 es una red neuronal convolucional con una profundidad de 164 capas que puede clasificar imágenes de 1000 clases distintas. Esta red es un híbrido que combina dos de las ideas más recientes en el diseño de arquitecturas de redes convolucionales: conexiones residuales, utilizadas en las redes ResNet [23], y la última versión de la arquitectura de Inception (InceptionV3). Existen evidencias de las ventajas de usar la fusión aditiva de señales para reconocimiento de imágenes y especialmente en detección de objetos [22]. En la figura 2.5 se muestra un ejemplo de esta idea utilizada en las conexiones de las redes ResNet.

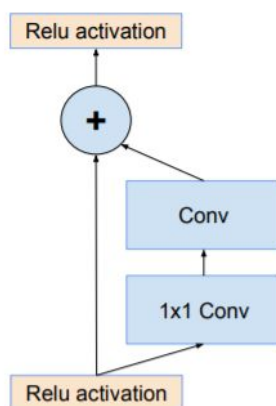


Figura 2.5: Versión optimizada de las conexiones de ResNet. Imagen obtenida de [22].

Las conexiones residuales tienen una importancia inherente a la hora de entrenar arquitecturas muy profundas. Dado que las arquitecturas de Inception tienen un gran número de capas, se planteó la idea de reemplazar la etapa de concatenación de filtros de Inception por conexiones residuales. Esto permite que esta nueva arquitectura se beneficie de las conexiones residuales mientras se mantiene la eficiencia computacional de Inception [22]. En la figura 2.6 se muestra la arquitectura de la red InceptionResNetV2.

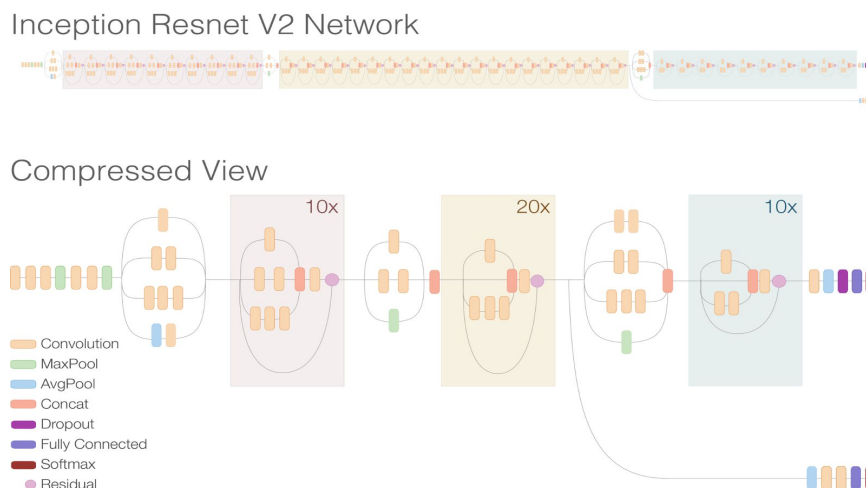


Figura 2.6: Arquitectura general de la red InceptionResNetV2. Imagen obtenida de: <https://ai.googleblog.com/2016/08/improving-inception-and-image.html>

Cabe destacar que esta arquitectura tiene muchas más capas y parámetros que InceptionV3, por lo que la hace una arquitectura más demandante en cuanto a recursos computacionales. Sin embargo, los resultados obtenidos al entrenar y validar esta red sobre el conjunto de datos de ImageNet son superiores a los de la arquitectura de InceptionV3, con una precisión de 80.3% y una precisión entre los 5 primeros de 95.3%.

2.2. Transfer-learning y fine-tuning

Transfer-learning es una técnica de aprendizaje automático donde un modelo que ha sido pre-entrenado para una tarea concreta es reutilizado para una segunda tarea relacionada. El transfer-learning se suele utilizar cuando el dataset del que se dispone para hacer una tarea es demasiado pequeño como para entrenar un modelo a gran escala desde cero. De este modo se aprovechan las características que han sido aprendidas por un modelo al ser entrenado con muchos más datos para una tarea relacionada, haciendo que el entrenamiento de este sistema sea mucho más rápido y haya menos probabilidades de que se produzca overfitting. Esta técnica puede ser de gran utilidad en aplicaciones médicas debido a la insuficiente cantidad de datos disponible para el entrenamiento de grandes modelos desde cero. El principal problema se encuentra en que la mayoría de modelos disponibles para transfer-learning se encuentran customizados para aplicaciones 2D en imágenes RGB. El proceso que se suele seguir para aplicar transfer-learning es el siguiente:

1. Tomar las capas de un modelo que ha sido entrenado previamente.
2. Congelar las capas, es decir, dejar fijos los parámetros aprendidos por la red para evitar perder la información aprendida en futuros entrenamientos.
3. Añadir nuevas capas entrenables seguidamente después de las capas congeladas, estas capas aprenderán cómo convertir las características aprendidas por el modelo congelado en predicciones en un nuevo dataset.
4. Entrenar las nuevas capas con el nuevo dataset.

Por otro lado, la técnica de fine-tuning utiliza un modelo pre-entrenado de la misma forma que en transfer-learning, pero en este caso se descongelan las capas del modelo (o solo algunas de ellas) y se re-entrena con los nuevos datos con un learning rate muy bajo. Potencialmente, esto puede lograr mejoras significativas al adaptar gradualmente las funciones previamente entrenadas a los nuevos datos.

2.3. Sistemas desarrollados

La primera red neuronal desarrollada en este TFG tiene como entrada de datos matrices 3D de dimensión 512x512x3. Los datos de entrada serán las imágenes ^{18}F -FDG PET preprocesadas de la misma forma que se hace en [1]. La arquitectura utilizada será la propuesta en [1]. A partir de la imagen ^{18}F -FDG PET de un paciente, la red calculará las probabilidades de pertenecer a las clases AD, MCI y CN, de modo que la predicción

final corresponderá la probabilidad más alta obtenida. En la figura 2.7 se puede observar una visión general de la arquitectura utilizada.

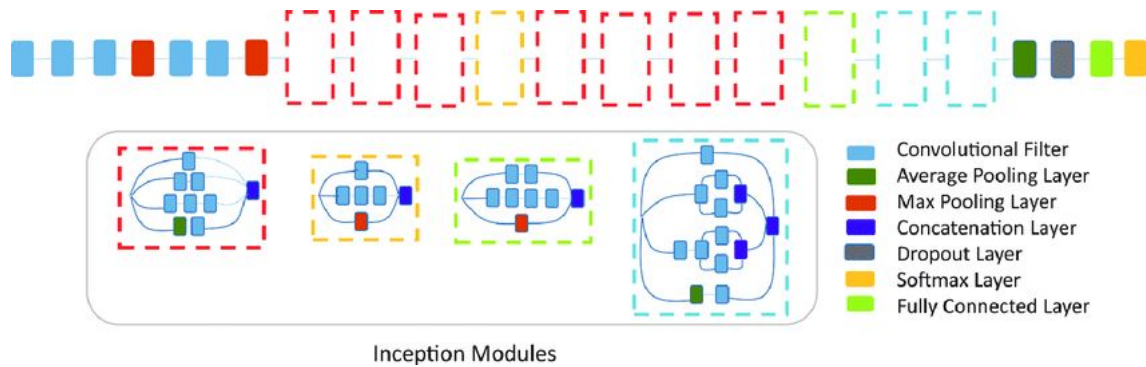


Figura 2.7: Arquitectura utilizada, utilizando InceptionV3 como modelo base. Cada módulo de Inception está compuesto por capas convolucionales y operaciones de pooling. La entrada del modelo son imágenes RGB de tamaño 512x512. Una capa Dropout y tres capas completamente conectadas de 1024, 512 y 3 neuronas respectivamente han sido añadidas encima de InceptionV3. Imagen obtenida de [1].

Las imágenes en escala de grises son transformadas a imágenes RGB concatenando tres veces cada valor de la matriz 512x512 para obtener un volumen 512x512x3. Éstas son cargadas en la red InceptionV3 que previamente ha sido pre-entrenada con el conjunto de datos de ImageNet. A través de los 11 módulos de Inception, compuestos por capas convolucionales y operaciones de pooling, se extraen las características más relevantes de las imágenes. La salida Inception será un volumen de dimensiones 14x14x2048, a esta salida se ha acoplado una capa que utiliza una operación de pooling denominada Global average pooling. La principal diferencia entre Average pooling y Global average pooling es que en lugar de utilizar un kernel que calcula la media de los valores abarcados por el kernel, calcula la media de todos los valores de cada dimensión. De este modo, la salida de esta capa será un vector unidimensional de tamaño 2048.

Seguidamente se ha añadido una capa Dropout como medida de regularización para evitar el overfitting. El ratio de probabilidad elegido es de 0.6, es decir, un 60% de las neuronas serán inhabilitadas en la capa siguiente durante el entrenamiento. Finalmente, se han añadido tres capas completamente conectadas con 1024, 512 y 3 neuronas respectivamente. En cada capa completamente conectada se ha hecho uso de la operación de Batch Normalization justo antes de aplicar la función de activación. Esto reduce en gran medida el overfitting producido durante el entrenamiento, observado en caso de no utilizarla. La función de activación utilizada en las capas completamente conectadas es la función ReLu (Rectified Linear unit) a excepción de la última capa que se ha utilizado la función de activación Softmax para obtener las predicciones.

La segunda red neuronal desarrollada reemplaza la red InceptionV3 por InceptionResNetV2, pre-entrenada también con el conjunto de datos de ImageNet. Se ha decidido utilizar esta red puesto que los resultados obtenidos a la hora de validar la red con el conjunto de datos de ImageNet son superiores a los obtenidos con InceptionV3. Además se ha añadido una capa Dropout con un ratio de probabilidad de 0.1 entre la capa completamente conectada de 1024 neuronas con la de 512 neuronas.

2.4. Datos utilizados

El conjunto de datos utilizado está compuesto por 2092 imágenes ^{18}F -FDG PET de 831 pacientes, 475 hombres y 356 mujeres, de entre 55 y 91 años. Los participantes están divididos en tres grupos, pacientes enfermos de Alzheimer (AD), pacientes con un deterioro cognitivo leve (MCI) y pacientes cognitivamente normales (CN).

Las imágenes han sido obtenidas de la base de datos de ADNI. ADNI dispone de cuatro tipos de preprocesado realizados con el objetivo de uniformizar las características de las imágenes PET obtenidas de los distintos sistemas de adquisición de las instituciones clínicas que las proporcionan. Los detalles de los tipos de preprocesado disponibles pueden encontrarse en <http://adni.loni.usc.edu/methods/pet-analysis-method/pet-analysis/#pet-pre-processing-container>. Para este TFG, se eligió el preprocesado (4), que después de alinear las imágenes a un sistema de coordenadas común uniformiza la resolución de las imágenes.

En la tabla 2.1 se presentan los detalles demográficos de los datos utilizados en este trabajo.

Diagnóstico clínico	No. Pacientes	No. Imágenes	Edad media (Hombres)	Edad media (Mujer)	No. Hombres por paciente	Follow-up medio (meses)
AD	369	884	75.77 ± 7.41	73.52 ± 8	220/369 (59.6%)	36.44 ± 24.82
MCI	246	606	74.68 ± 6.89	72.70 ± 8	155/246 (63%)	53.21 ± 38.09
CN	213	602	73.41 ± 6.44	73.01 ± 6	97/213 (45.53%)	39.51 ± 24.31
Total	831	2092	74.94 ± 7.15	73.14 ± 7	475/831 (57.16%)	41.81 ± 29.34

Tabla 2.1: Resumen de la información demográfica del conjunto de datos.

En este punto nos gustaría añadir que se realizó una búsqueda exhaustiva de los datos utilizados en el estudio en el que nos basamos [1] sin que pudiésemos encontrar un número de pacientes e imágenes con una distribución similar a la del artículo. Se contactó con los autores de [1] solicitando esta información para trabajar con un conjunto de datos lo más similar posible al estudio. Uno de los autores, Jae Ho Sohn, que tuvo la amabilidad de contestarnos, nos informó de que los identificadores de los pacientes del estudio no estaban disponibles. Tampoco pudo aclararnos el tipo de preprocesado utilizado, ni si el preprocesado disponible en ADNI se hizo con posterioridad al trabajo presentado en [1].

2.5. Preprocesado de las imágenes

El preprocesado de las imágenes ^{18}F -FDG PET, realizado para alimentar las redes neuronales 2D, es el propuesto en [1]. En primer lugar, las imágenes son remuestreadas a vóxeles isotrópicos de 2mm y recortadas en una cuadrícula de 100x100x90 píxeles, obteniendo así un volumen de 200x200x180 mm³. El método de segmentación Otsu's thresholding [30] es aplicado al volumen resultante para seleccionar los vóxeles correspondientes al cerebro, es decir, descartar partes del cráneo irrelevantes que puede hacer que la red se confunda al entrenar. Seguidamente, se realiza un análisis de conectividad sobre los volúmenes para reconectar aquellas partes del cerebro que hayan podido ser desconectadas tras aplicar Otsu. Para ello, se hace uso de operaciones de morfología matemática como erosión y dilatación. Con ellas se consigue rellenar los agujeros en el interior del cerebro.

A continuación, se calcula el centro de masas de cada volumen y se seleccionan 16 secciones del cerebro separadas equidistantemente. Estas secciones se distribuyen en una cuadrícula 4x4 con la sección más craneal en la parte superior izquierda y la sección más caudal en la parte inferior derecha. Después de este preprocesamiento, las imágenes son matrices 512x512 que son normalizadas para que todos los valores estén entre 0 y 1. En la figura 2.8 se muestra la sección central del cerebro de la imagen ^{18}F -FDG PET original de un paciente con AD. En la figura 2.9 se muestra un ejemplo de la imagen resultante al aplicar este preprocesado al mismo volumen de datos de la figura 2.8. Se puede observar que en la figura 2.8 aparece el cráneo del paciente en la imagen mientras que en la figura 2.9 se ha filtrado para mostrar únicamente el cerebro y reducir el ruido de las imágenes.

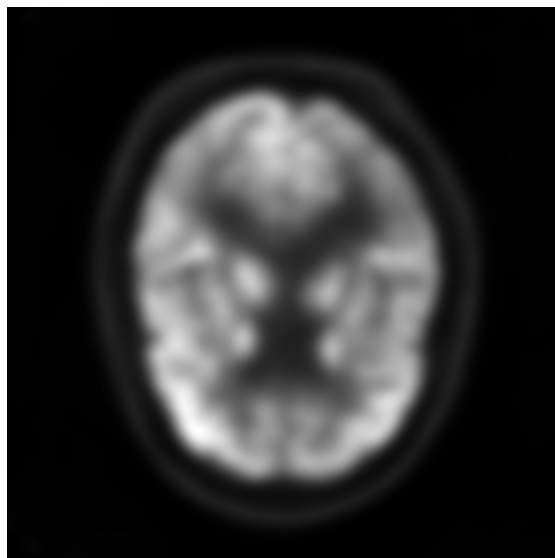


Figura 2.8: Vista axial de la sección central del cerebro de la imagen ^{18}F -FDG PET de un paciente con AD.

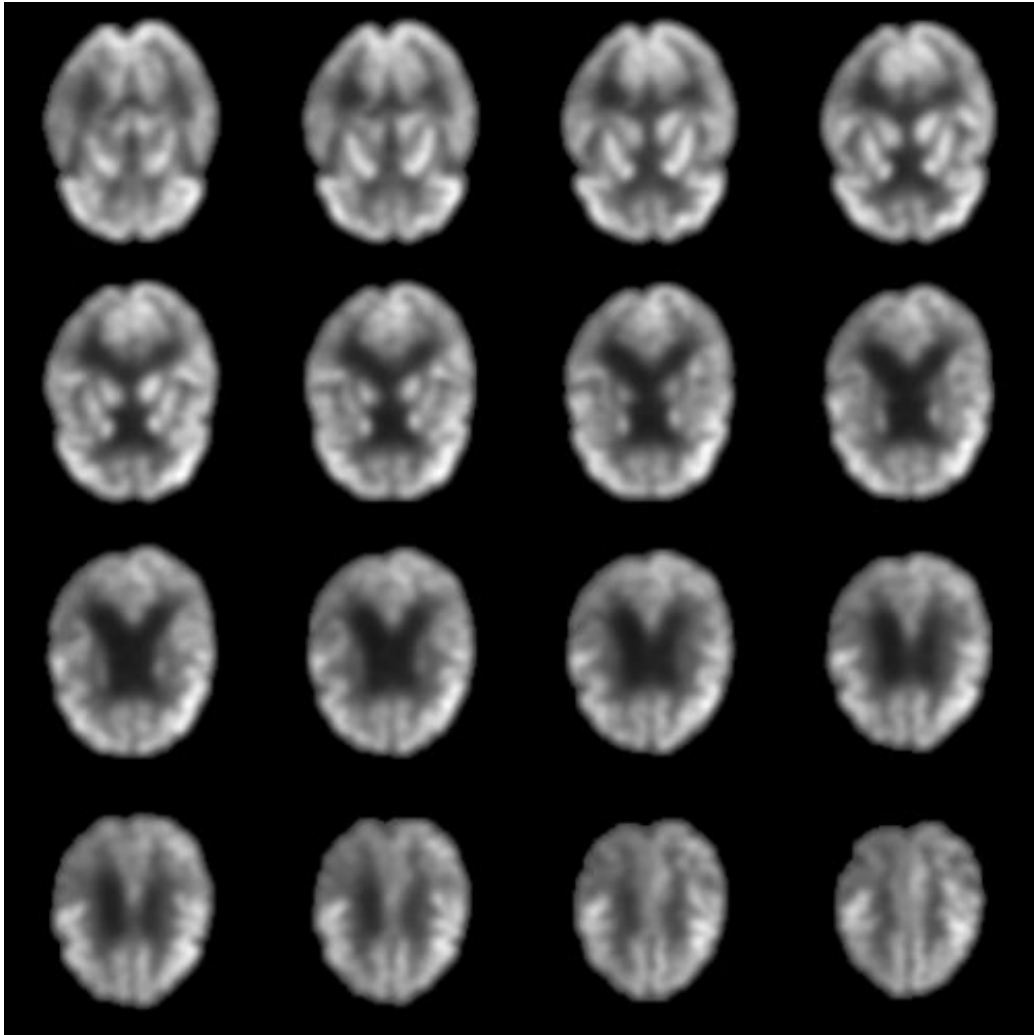


Figura 2.9: Resultado del preprocesado de la imagen ^{18}F -FDG PET de un paciente con AD.

2.6. Entrenamiento de la red

Los parámetros de entrenamiento utilizados son los propuestos en [1]. Durante el desarrollo de este TFG se realizaron múltiples experimentos con parámetros diferentes a éstos, pero los resultados obtenidos no mejoraron la precisión del sistema.

Así, los parámetros utilizados son los siguientes:

- Un tamaño de batch de ocho muestras.
- Categorical cross-entropy como función de coste, esta función es utilizada cuando se tiene un problema de clasificación multiclase.
- Optimizador Adam, con una tasa de aprendizaje igual a 0.0001.

Los sistemas fueron entrenados durante cuatro épocas debido al gran overfitting producido a partir de la cuarta época, incluso aplicando las técnicas de estabilización y regularización Batch Normalization y Dropout. Se utilizó también la

técnica conocida como aumento de datos (Data Augmentation) con el objetivo de reducir este overfitting.

Los parámetros utilizados para el aumento de datos son:

- Rotation range: Rango de grados para realizar rotaciones aleatorias de la imagen. El valor utilizado es 8, por lo que el rango será de 0 a 8 grados.
- Shear range: Determina el ángulo de corte en sentido antihorario en grados. El valor utilizado es $\frac{\pi}{16}$.
- Width shift range: Determina el desplazamiento horizontal de la imagen. El valor elegido es 0.1, por lo que el rango será entre un 0% y un 10% de desplazamiento.
- Height shift range: Determina el desplazamiento vertical de la imagen. El valor elegido es 0.1, por lo que el rango será entre un 0% y un 10% de desplazamiento.
- Zoom range: Rango para aplicar un zoom aleatorio. El valor elegido es 0.08, es decir, un zoom entre el 0% y 8%.

2.7. Detalles de la implementación

Todo el código desarrollado se ha implementado en Python 3.6.9. El preprocesamiento de las imágenes $^{18}\text{F-FDG}$ PET se ha realizado con el uso de las librerías NiBabel (versión 3.0.2), Nilearn (versión 0.6.2), SciPy (versión 1.4.1) y Scikit-image (versión 0.16.2). Para el desarrollo de las redes neuronales convolucionales se ha hecho uso de la librería Keras (versión 2.2.4), utilizando Tensorflow (versión 1.12) como backend.

NiBabel [24] es un paquete de código abierto que proporciona acceso de lectura y escritura a algunos formatos de archivo médicos y de neuroimagen comunes. Nilearn [25] es un módulo de Python para el aprendizaje estadístico rápido e intuitivo con datos de neuroimagen.

SciPy [26] es una biblioteca libre y de código abierto desarrollada para Python que está compuesta de módulos para optimización, álgebra lineal, integración, interpolación funciones especiales, la transformada rápida de fourier (FFT), procesamiento de señales y de imagen y otras tareas para la ciencia e ingeniería. Scikit-image [27] es una librería de código abierto que proporciona una colección de algoritmos para el procesamiento de imágenes.

Tensorflow [28] es una plataforma de código abierto end-to-end para aprendizaje automático. Otorga una gran flexibilidad a la hora de desarrollar modelos y facilita en gran medida la ejecución de los mismos en múltiples unidades de procesamiento y en tarjetas gráficas (GPU).

Keras [29] es una API para el desarrollo de redes neuronales, capaz de utilizar distintas bibliotecas como backend, incluyendo Tensorflow, CNTK y Theano. Keras facilita la implementación de modelos de deep learning complejos y es fácilmente

extensible para la inclusión de módulos customizables, como capas, funciones de coste, optimizadores etc.

El proceso de entrenamiento de todos los experimentos se ha ejecutado sobre una tarjeta gráfica de NVidia Titan RTX, que cuenta con 24 GB de VRAM. Los drivers utilizados para ésta incluyen la versión 9.0 de CUDA y la versión 7.5.0 de cuDNN. El resto de especificaciones del equipo utilizado incluyen una RAM de 32 GB y un procesador Intel(R) Core(TM) i7-7700, que dispone de cuatro núcleos a 3,60 Ghz.

El coste temporal del entrenamiento del primer sistema de aprendizaje desarrollado es de 266.97 segundos (66.74 por época), con una utilización de 3.037 GB de memoria. Para el segundo sistema desarrollado, el coste temporal del entrenamiento es de 376.50 segundos (94.125 por época), con una utilización de 7.622GB de memoria. En ambos modelos, el tiempo empleado en realizar la predicción de una única muestra es inferior a un segundo.

3. RESULTADOS

En esta sección se presentan los resultados obtenidos con ambos sistemas de predicción desarrollados. Posteriormente se presenta un sistema de predicción adicional desarrollado capaz de predecir AD vs CN con gran precisión y se realiza un análisis del impacto en el rendimiento al introducir la clase MCI en la predicción.

3.1. Arquitectura InceptionV3

Los resultados obtenidos del modelo con la misma arquitectura que la propuesta en [1] se pueden observar en la tabla 3.1. Se muestran todas las métricas utilizadas para la evaluación del modelo: sensibilidad, especificidad, precisión, F1-score y Área bajo la curva ROC (AUC).

Resultados de este TFG					
Clase	Sensibilidad	Especificidad	Precisión	F1 Score	AUC
AD	45.2% (24/53)	95.62% (153/160)	77.41% (24/31)	57.14%	0.84
MCI	62.0% (54/87)	65.87% (83/126)	55.67% (54/97)	58.69%	0.70
CN	78.0% (57/73)	80.00% (112/140)	67.05% (57/85)	72.15%	0.88
Resultados de [1]					
AD	81% (29/36)	94% (143/152)	76% (29/38)	78%	0.92
MCI	54% (43/79)	68% (74/109)	55% (43/78)	55%	0.63
CN	59% (43/73)	75% (86/115)	60% (43/72)	59%	0.73

Tabla 3.1: Resultados obtenidos en este TFG a partir del modelo con la arquitectura basada en la propuesta en [1] sobre el conjunto de datos de test, junto con los resultados obtenidos en [1]. En el anexo B se encuentran la explicación de qué representa cada métrica.

De la tabla 3.1 se concluye que el modelo obtenido permite clasificar pacientes enfermos con una precisión razonable y discriminar correctamente pacientes que evolucionan a AD frente a aquellos que permanecen como MCI o CN. Los valores de precisión y especificidad obtenidos son muy similares a los obtenidos en [1], siendo incluso ligeramente superiores en la clase de mayor interés (AD). Sin embargo, la sensibilidad del modelo para la clase AD es muy baja en comparación con la obtenida

en [1], lo que indica que el sistema no siempre es capaz de identificar aquellos pacientes que evolucionarán a AD. No obstante, aquellos pacientes identificados como AD serán clasificados correctamente con un 77.41% de precisión. Estas discrepancias en los resultados con los obtenidos en [1] pueden deberse a la diferencia en los conjuntos de datos utilizados, puesto que la población de pacientes y las imágenes utilizadas en este trabajo son diferentes a los utilizados en [1].

Para hacer un análisis en profundidad del sistema de predicción desarrollado se generaron diversas gráficas. Entre estas gráficas se incluyen, una gráfica de las curvas ROC para cada clase, una gráfica de las muestras de entrenamiento reducidas a dos dimensiones con la técnica t-distributed Stochastic Neighbor Embedding (t-SNE) para ver si se produce alguna clusterización entre las clases y un mapa de saliencia para poder evaluar la atención de la red neuronal. A continuación se muestran en las figuras 3.1, 3.2 y 3.3 las gráficas generadas con el modelo desarrollado.

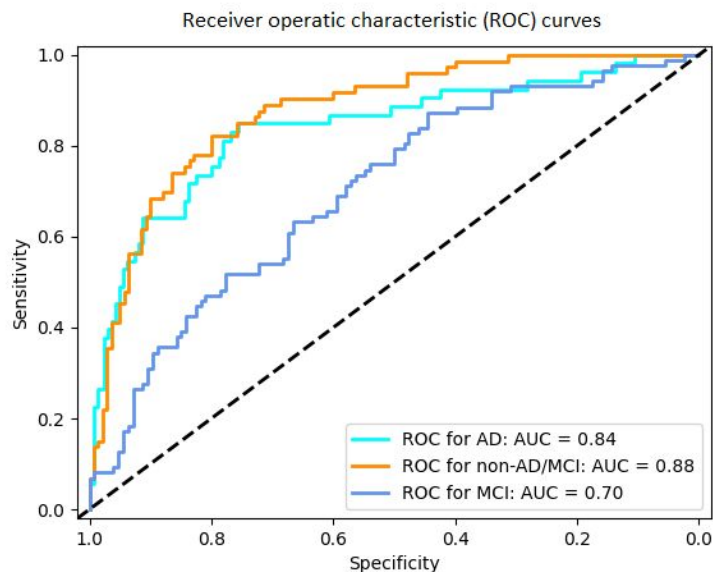


Figura 3.1: Curvas ROC obtenidas al entrenar el modelo con la arquitectura basada en la propuesta en [1]. La línea azul clara es para la clase AD, la naranja para la clase CN y la azul oscura para la clase MCI.

Los AUC que se observan en la figura 3.1 indican que la red neuronal tiene una capacidad razonable de distinguir pacientes que finalmente evolucionan a AD en el momento de la obtención de la imagen frente aquellos que se mantienen en las clases MCI o CN. Se puede apreciar que la red neuronal es más precisa a la hora de identificar pacientes cognitivamente normales (CN) que pacientes de las otras dos clases. El AUC obtenido para la clase MCI es significativamente menor que para las otras dos clases. Esto puede ser debido a que, al tratarse el deterioro cognitivo leve de un estado intermedio entre un paciente cognitivamente normal y otro que pueda padecer la enfermedad de Alzheimer, las imágenes de la clase MCI no muestran evidencias claras que permitan realizar un diagnóstico igual de preciso que el de un paciente normal u otro con Alzheimer. De este modo, la red neuronal presenta una mayor confusión a la hora de clasificar la clase MCI.

El objetivo del gráfico mostrado en la figura 3.2 es el de poder visualizar si la red es capaz de discriminar distintos grupos entre las muestras tras realizar la forward propagation. Para ello se obtiene la salida de la red en la capa completamente conectada de 1024 neuronas, pues es la capa inmediata a la salida producida por InceptionV3. Debido a la alta dimensionalidad de la salida producida por esta capa (1024) es necesario aplicar algunas técnicas de reducción de dimensionalidad para poder mostrar los datos en una gráfica. En primer lugar se aplica la técnica de Análisis de componentes principales (Principal component analysis, PCA) para reducir la dimensionalidad de los datos de 1024 a 30 dimensiones. PCA es una técnica de reducción de dimensionalidad que permite mantener la varianza global de las muestras del conjunto de datos. Las dimensiones de las muestras son ordenadas por la cantidad de varianza original que describen. Entre ellas se seleccionan las dimensiones más relevantes [31].

Una vez reducida la dimensionalidad de los datos a 30 es cuando se aplica la técnica t-distributed Stochastic Neighbor Embedding (t-SNE) para reducir nuevamente la dimensionalidad de los datos a 2 dimensiones. t-SNE es una técnica de reducción de dimensionalidad no lineal muy efectiva a la hora de reducir datos de alta dimensión en datos de menor dimensión (2D o 3D) que facilita la visualización de los mismos. Específicamente, modela cada objeto de alta dimensión por un punto bidimensional o tridimensional de tal manera que objetos similares son modelados por puntos cercanos y objetos diferentes son modelados por puntos distantes con alta probabilidad [32]. Este algoritmo tiene dos parámetros que hay especificar y cuyos valores corresponden a los utilizados en [1], el primero de ellos es el número de iteraciones, igual a 1000, y el segundo es el factor de aprendizaje, establecido en 200.

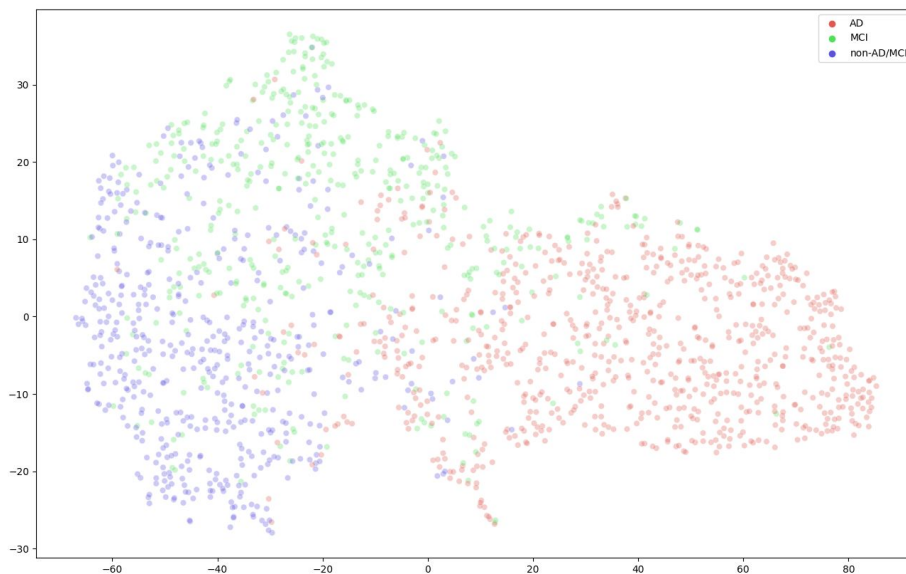


Figura 3.2: Visualización de los datos de entrenamiento tras realizar una reducción de dimensionalidad con la técnica t-SNE. Cada punto representa las 1024 características producidas por la capa completamente conectada de 1024 neuronas del modelo con la arquitectura basada en la propuesta en [1]. Los puntos rojos pertenecen a muestras de la clase AD, los puntos azules a muestras de la clase CN y los puntos verdes a muestras de la clase MCI.

En la figura 3.2 se muestra la salida producida por la capa fully-connected de 1024 neuronas tras realizar la forward propagation de los datos de entrenamiento. La dimensionalidad de la salida de esta capa ha sido reducida mediante las técnicas de PCA y t-SNE para poder ser visualizable. Cada punto representa las 1024 características producidas por dicha capa. Se puede observar cómo los puntos rojos (AD) están claramente separados de los puntos azules (CN), lo cual indica que el modelo es capaz de diferenciar AD vs CN correctamente. Sin embargo, se aprecia que los puntos verdes (MCI) se encuentran mezclados entre las dos clases, esto es un claro indicio de que el modelo tiene mayor dificultad a la hora de identificar la clase MCI, reflejado en los resultados obtenidos a la hora de realizar las predicciones.

Finalmente, la figura 3.3 muestra el mapa de saliencia medio obtenido sobre el conjunto de datos de test. Estos mapas son utilizados para ver qué partes de las imágenes están teniendo una mayor influencia en el modelo a la hora de realizar las predicciones. Esta influencia es normalmente codificada con un código de colores, en la figura 3.3 los colores más cálidos representan una influencia mayor. Se puede observar que la región que incluye el lóbulo parietal del cerebro es la región que aporta más información al modelo y tiene más influencia a la hora de realizar las predicciones (<https://radiopaedia.org/cases/brain-lobes-annotated-mri-1>). Se ha demostrado que el hipometabolismo en la región parietal está asociado con la AD (<https://www.ncbi.nlm.nih.gov/medgen/868400>). Por tanto, el modelo desarrollado ha sido capaz de identificar esa relación y utilizarla para realizar las predicciones de AD. Como posible mejora al sistema, después de analizar este mapa de saliencia, se podría restringir el preprocesado con una teselación más fina en un entorno de la región parietal en lugar de utilizar todo el cerebro.

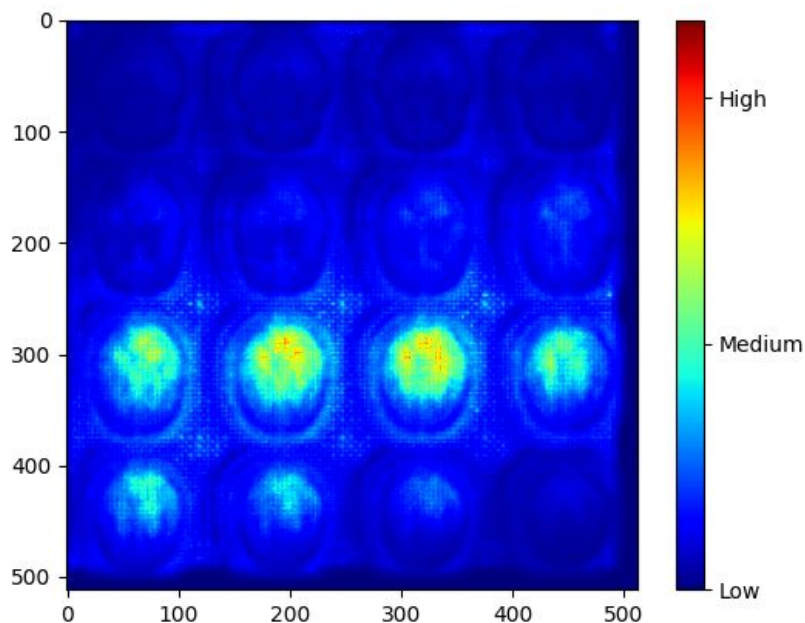


Figura 3.3: Mapa de saliencia medio sobre el conjunto de datos de test obtenido con la arquitectura basada en la propuesta en [1]. Los colores más cercanos al rojo indican una influencia mayor en el modelo a la hora de realizar las predicciones. Los colores más azulados representan una influencia baja en el modelo.

3.2. Arquitectura InceptionResNetV2

Los resultados obtenidos del modelo con la arquitectura que utiliza la red InceptionResNetV2 se pueden observar en la tabla 3.2. Se muestran todas las métricas utilizadas para la evaluación del modelo: sensibilidad, especificidad, precisión, F1-score y Área bajo la curva ROC (AUC).

Clase	Sensibilidad	Especificidad	Precisión	F1 Score	AUC
AD	62.2% (33/53)	96.25% (154/160)	84.61% (33/39)	71.73%	0.89
MCI	75.8% (66/87)	46.82% (59/126)	49.62% (66/133)	43.85%	0.68
CN	34.2% (28/73)	88.57% (124/140)	60.97% (25/41)	60.00%	0.81

Tabla 3.2: Resultados obtenidos del modelo con la arquitectura que utiliza la red InceptionResNetV2 sobre el conjunto de datos de test.

A partir de los resultados de la tabla 3.2 se puede observar que al utilizar esta arquitectura se consiguen resultados significativamente mejores que con la arquitectura anterior a la hora de clasificar pacientes enfermos de Alzheimer. La mejora más significativa corresponde a la sensibilidad: este modelo es mejor a la hora de identificar qué pacientes evolucionarán a AD (62.2 vs 45.2 %), además con una mayor precisión (84.61 vs 77.41 %). Sin embargo, la sensibilidad obtenida sigue siendo inferior al modelo obtenido en [1] (81 %). Una razón de la mejora en los resultados es que InceptionResNetV2 es una red más potente y actual que InceptionV3, por lo que para el mismo problema cabe esperar que se obtengan mejores resultados con esta nueva arquitectura.

Para determinar la calidad de este sistema de predicción, se ha realizado el mismo análisis en profundidad que para la arquitectura anterior, generando las mismas gráficas. En la figura 3.4 se muestra la gráfica de las curvas ROC para cada clase. En la figura 3.5, la gráfica de las muestras de entrenamiento reducidas a dos dimensiones con la técnica t-SNE. Por último, en la figura 3.6 se muestra el mapa de saliencia obtenido.

Los AUC que se observan en la figura 3.4 indican que la red neuronal tiene una capacidad superior al modelo anterior de distinguir pacientes que finalmente evolucionan a AD en el momento de la obtención de la imagen frente aquellos que se mantienen en las clases MCI o CN. A pesar de que se ha producido una ligera bajada en el rendimiento a la hora de identificar las clases de MCI y CN, el objetivo primordial es el identificar pacientes enfermos de AD, por lo que es preferible utilizar este modelo al de la arquitectura que usa InceptionV3 para la predicción de la evolución a Alzheimer.

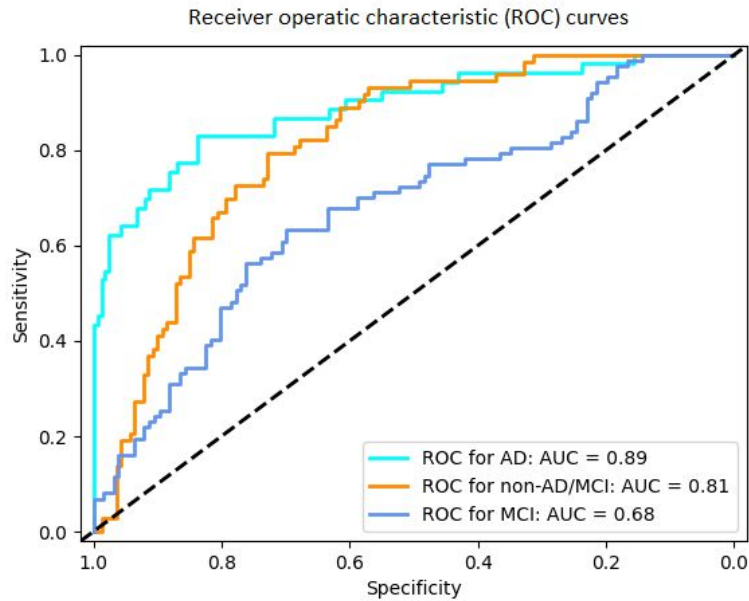


Figura 3.4: Curvas ROC obtenidas al entrenar el modelo con la arquitectura que utiliza la red InceptionResNetV2. La línea azul clara es para la clase AD, la naranja para la clase CN y la azul oscura para la clase MCI.

En la figura 3.5 se muestra la salida producida por la capa fully-connected de 1024 neuronas tras realizar la forward propagation de los datos de entrenamiento. La dimensionalidad de la salida de esta capa ha sido reducida mediante las técnicas de PCA y t-SNE de la misma forma que con la arquitectura anterior. Se observa que en este caso los puntos verdes (AD) están mucho más separados de las otras dos clases, lo que explica la gran mejora de este nuevo modelo a la hora de identificar AD respecto al modelo anterior. Sin embargo, los puntos rojos (MCI) y azules (CN) se siguen encontrando entremezclados, lo cual es una clara evidencia de la dificultad que tiene el modelo a la hora de diferenciar la clase MCI del resto.

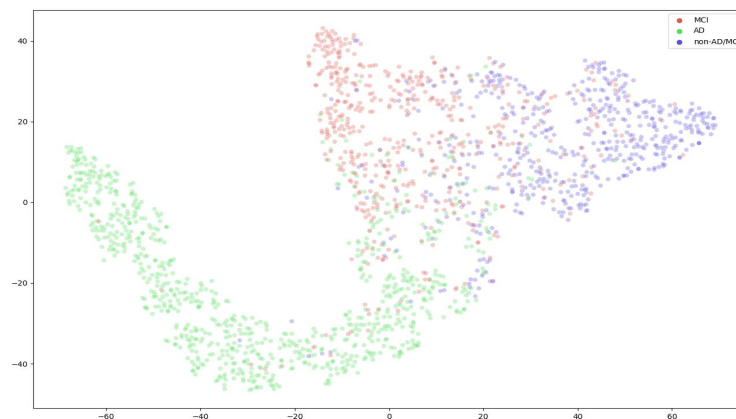


Figura 3.5: Visualización de los datos de entrenamiento tras realizar una reducción de dimensionalidad con la técnica t-SNE. Cada punto representa las 1024 características producidas por la capa completamente conectada de 1024 neuronas del modelo con la arquitectura que utiliza la red InceptionResNetV2. Los puntos verdes pertenecen a muestras de la clase AD, los puntos azules a muestras de la clase CN y los puntos rojos a muestras de la clase MCI.

La figura 3.6 muestra el mapa de saliencia medio obtenido sobre el conjunto de datos de test. Se puede observar que la mayor influencia a la hora de hacer las predicciones sigue estando en la región del lóbulo parietal del cerebro de un paciente. Cabe destacar que con este modelo se presentan más puntos con una alta influencia (puntos rojos) que con el modelo anterior, esto se puede deber a la mayor capacidad de InceptionResNetV2 para reconocer ciertas características.

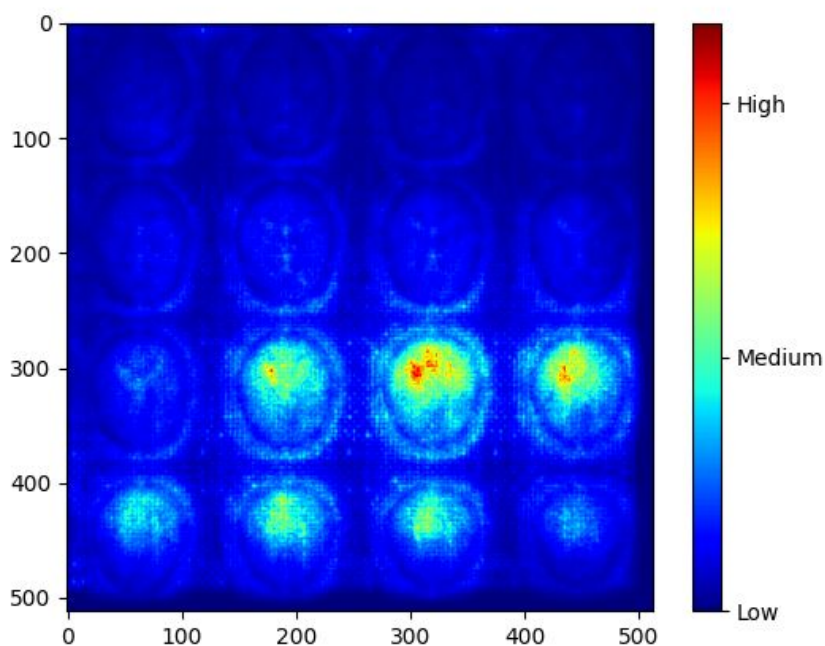


Figura 3.6: Mapa de saliencia medio sobre el conjunto de datos de test obtenido con la arquitectura que utiliza la red InceptionResNetV2. Los colores más cercanos al rojo indican una influencia mayor en el modelo a la hora de realizar las predicciones. Los colores más azulados representan una influencia baja en el modelo.

3.3. Sistema de predicción AD vs CN

Tras observar los resultados obtenidos con las dos arquitecturas anteriores, se propuso desarrollar un sistema en el que únicamente se tuvieran en cuenta pacientes cognitivamente normales (CN) y pacientes que evolucionarán a Alzheimer (AD). La clase MCI fue descartada. El objetivo de este sistema es observar si se obtiene un modelo robusto capaz de predecir correctamente aquellos pacientes que evolucionarán a AD al no introducir pacientes con un deterioro cognitivo leve en el sistema. Finalmente, se incluyó la clase MCI en el entrenamiento y se analizó el impacto de la misma en el modelo. La arquitectura utilizada es la misma que la propuesta en [1], sin embargo, la última capa es ahora una única neurona con la función de activación sigmoide, utilizada para la clasificación binaria. Además la función de coste pasa de ser categorical cross entropy a binary cross entropy.

Los resultados obtenidos al entrenar este modelo únicamente con imágenes de pacientes CN y AD son los siguientes:

- Sensibilidad: 92.45%
- Especificidad: 90.41%
- Precisión: 87.50%
- F1-Score: 89.90%
- AUC: 0.97

Los resultados obtenidos con este modelo son muy superiores a los obtenidos con los otros dos, debido a que la red neuronal presenta una menor confusión a la hora de realizar las predicciones dada la ausencia de la clase MCI, la cual se trata de un estado intermedio entre CN y AD. Este modelo es capaz de identificar con gran precisión las diferencias más evidentes entre los pacientes cognitivamente normales y aquellos que evolucionan a Alzheimer y realizar predicciones muy precisas. Este gran rendimiento de la red se puede apreciar en la figura 3.7, se observa que el AUC obtenido es muy cercano a 1 (0.97), esto indica que el modelo es muy eficiente a la hora de discriminar entre pacientes enfermos y sanos.

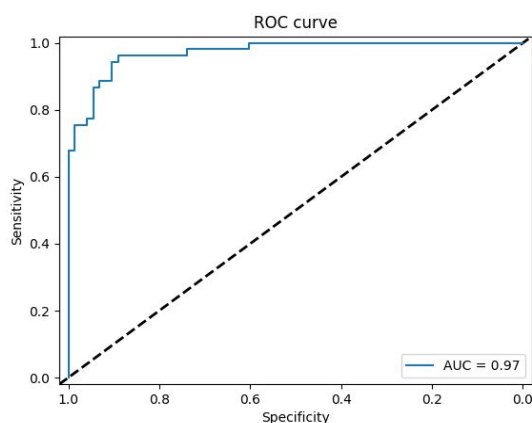


Figura 3.7: Curva ROC obtenidas al entrenar el modelo con la arquitectura basada en la propuesta en [1] pero únicamente con imágenes de pacientes AD y CN.

Para analizar mejor el funcionamiento de este modelo se han generado las mismas gráficas que para los otros modelos. En la figura 3.8 se muestra la gráfica de las muestras de entrenamiento reducidas a dos dimensiones con la técnica t-SNE. En la figura 3.9 se muestra el mapa de saliencia medio obtenido con el conjunto de datos de test.

En la figura 3.8 se muestra la salida producida por la capa fully-connected de 1024 neuronas tras realizar la forward propagation de los datos de entrenamiento. Se observa que los puntos rojos (AD) y los puntos azules (CN) están claramente separados entre sí, a excepción de casos que pueden confundir a la red, visibles en la zona donde se mezclan los puntos de las dos clases. Esto es un claro indicio de la gran capacidad de la red de diferenciar pacientes AD de los pacientes CN y de la alta precisión con la que es realizada la clasificación.

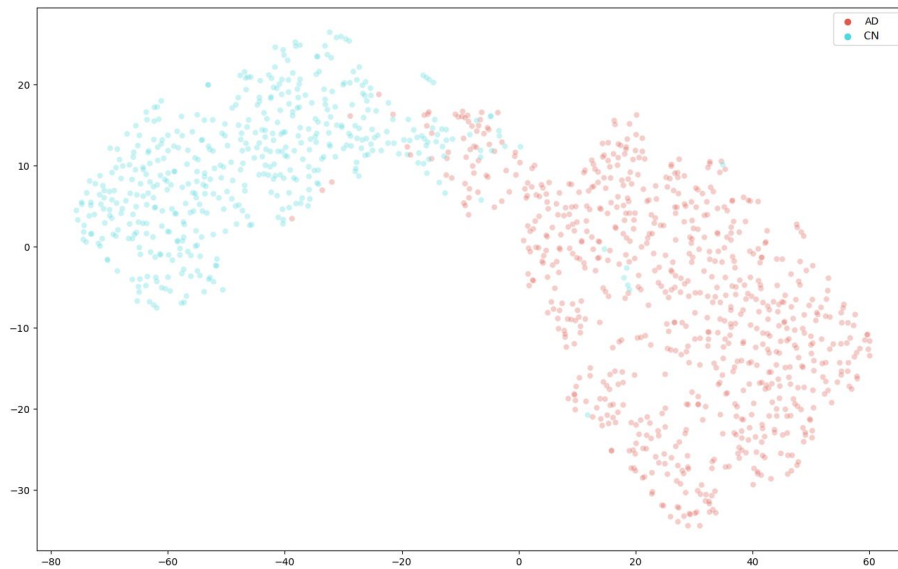


Figura 3.8: Visualización de los datos de entrenamiento tras realizar una reducción de dimensionalidad con la técnica t-SNE. Cada punto representa las 1024 características producidas por la capa completamente conectada de 1024 neuronas del modelo AD vs CN. Los puntos rojos pertenecen a muestras de la clase AD, los puntos azules a muestras de la clase CN.

La figura 3.9 muestra el mapa de saliencia medio obtenido sobre el conjunto de datos de test. Se puede observar que los puntos de interés están mucho más concentrados que en los mapas de saliencia anteriores, esto se puede deber a la ausencia del ruido introducido por la clase MCI. De modo que los puntos de interés en este modelo, aunque más concentrados, son mucho más relevantes a la hora de realizar las predicciones correspondientes. Estos puntos de interés se encuentran en la parte superior de los ventrículos cerebrales y la mayor influencia se encuentra en la cola izquierda del núcleo caudado. Es conocido que las zonas medias del giro cingulado y el núcleo caudado presentan un mayor grado de atrofia en los niveles de glucosa en pacientes con AD [37,38].

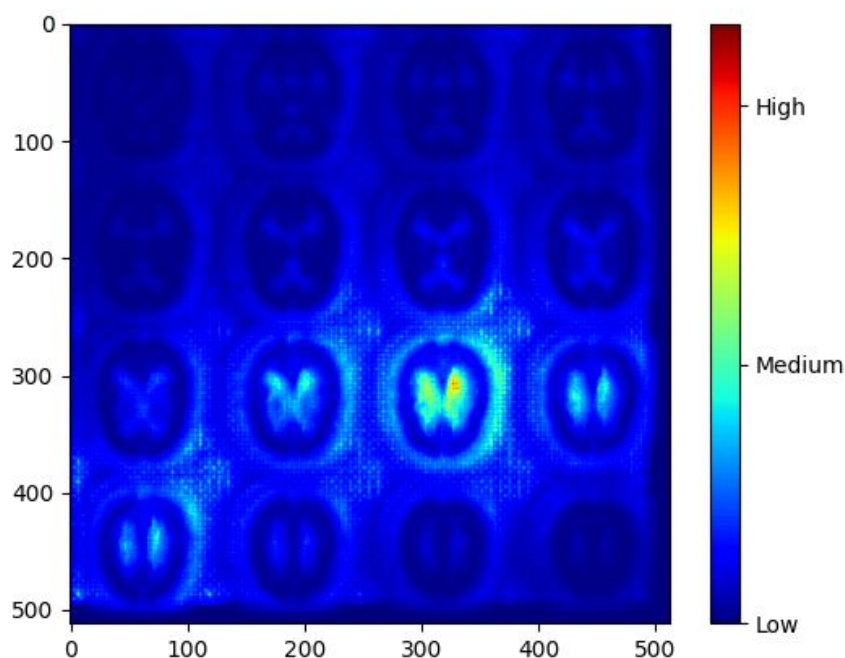


Figura 3.9: Mapa de saliencia medio sobre el conjunto de datos de test obtenido con el modelo AD vs CN. Los colores más cercanos al rojo indican una influencia mayor en el modelo a la hora de realizar las predicciones. Los colores más azulados representan una influencia baja en el modelo.

Finalmente, se analizó cómo afectaba la clase MCI al rendimiento de este modelo y se propuso re-entrenar este modelo añadiendo los pacientes MCI en la misma clase que los pacientes CN, realizando así una clasificación binaria AD vs no-AD (MCI+CN). Los resultados obtenidos fueron los siguientes:

- Sensibilidad: 75.47%
- Especificidad: 83.75%
- Precisión: 60.60%
- F1-Score: 67.22%
- AUC: 0.88

Se observa que se ha producido una bajada considerable en el rendimiento de la red al introducir las imágenes de pacientes MCI, el resultado obtenido era el esperado. La red sufre más confusiones debido a las similitudes que tiene MCI con AD y con CN y esto produce un deterioro en la sensibilidad y especificidad del modelo. La red, al ver imágenes de pacientes MCI, no tiene indicios tan claros como cuando es entrenada únicamente con imágenes de pacientes AD y CN para poder discriminar con la misma precisión los pacientes enfermos de los sanos.

4. CONCLUSIONES

El principal objetivo de este TFG ha consistido en desarrollar un sistema de aprendizaje profundo a partir de imágenes ^{18}F -FDG PET capaz de realizar la predicción temprana del diagnóstico final de un paciente, basado en la arquitectura propuesta en [1]. El problema consiste en la clasificación de individuos entre pacientes que padecen o evolucionarán a la enfermedad de Alzheimer, un deterioro cognitivo leve o son cognitivamente normales.

En primer lugar, se ha llevado a cabo un estudio del funcionamiento de las redes neuronales convolucionales, así como de las arquitecturas más modernas como InceptionV3 y InceptionResNetV2. Se ha realizado una búsqueda exhaustiva de los datos utilizados en el estudio en el que nos basamos [1] sin que pudiésemos encontrar un número de pacientes e imágenes con una distribución similar a la del artículo, la búsqueda y descarga de los datos se hizo a través de la base de datos ADNI. A continuación se ha realizado la implementación de los tres sistemas de aprendizaje descritos en la memoria, entre los que cabe destacar el sistema que utiliza la red InceptionResNetV2 por haber sido el sistema que mejores resultados ha obtenido en la predicción de AD. La implementación se ha realizado en Python 3.6.9 mediante el uso de librerías como Keras y Tensorflow. Las imágenes ^{18}F -FDG PET han sido preprocesadas siguiendo el mismo esquema utilizado en [1]. Para este preprocesamiento se han hecho uso de librerías como Nibabel, Nilearn, Scipy y scikit-image. Finalmente, se han realizado una serie de experimentos que incluyen los descritos en la memoria.

De los resultados obtenidos cabe destacar que no se han logrado reproducir los obtenidos en [1], debido a la imposibilidad de encontrar el mismo conjunto de datos que fue utilizado en [1]. Los modelos desarrollados con nuestro conjunto de datos son menos sensibles que el de [1] a la hora de identificar pacientes que evolucionarán a AD. No obstante, han sido capaces de predecir AD hasta 66 meses (5 años y medio) antes del diagnóstico final con una precisión razonable.

Los mejores resultados se han conseguido con el modelo que utiliza la red InceptionResNetV2, con una sensibilidad de 62.2%, una especificidad de 96.25%, una precisión de 84.61%, un F1-Score de 71.73% y un AUC de 0.89. Adicionalmente, se ha desarrollado un modelo para resolver el problema de la clasificación AD vs CN obteniendo unos grandes resultados, una sensibilidad de 92.45%, una especificidad de 90.41%, una precisión de 87.5%, un F1-Score de 89.9% y un AUC de 0.97. Con este modelo se ha hecho un análisis del impacto producido al introducir la clase MCI y se ha observado una gran bajada en el rendimiento del modelo debido a la gran confusión que presenta la red a la hora de diferenciar pacientes AD de MCI.

Los mapas de saliencia obtenidos a partir de los modelos resaltan que la región cerebral que tiene una mayor influencia a la hora de realizar las predicciones es la que contiene el lóbulo parietal. Se ha demostrado que el hipometabolismo en la región parietal está asociado con la AD, por lo que parece que la red es capaz de aprender los criterios utilizados en la práctica clínica y utilizarlos para realizar el diagnóstico.

En la figura 4.1 se muestra un diagrama de Gantt con las tareas desarrolladas en la realización del TFG:

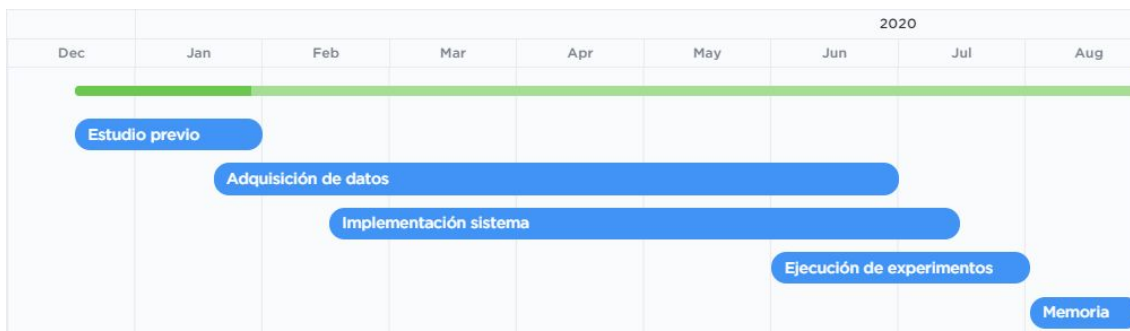


Figura 4.1: Diagrama de Gantt del proyecto.

Como líneas futuras de trabajo, se podría restringir el preprocesado con una teselación más fina en un entorno de la región parietal con el objetivo de conseguir un sistema de predicción más robusto. También, se podrían añadir las secciones sagital y coronal a la teselación axial que se ha utilizado para alimentar la red con más información relevante. Las imágenes PET se podrían combinar con imágenes de distinta modalidad como MRI o DTI. De este modo, el modelo podría nutrirse de distintas fuentes de información y realizar predicciones más precisas. En lugar de trabajar con teselaciones 2D de las imágenes, se podría abordar el problema directamente en 3D y comparar el rendimiento de ambos modelos.

Adicionalmente, se podría completar este estudio con otros métodos de interpretabilidad más sofisticados que el mapa de saliencia como Integrated Gradient, presentado en [39] y disponible en Tensorflow. Integrated Gradient tiene como objetivo explicar la relación entre las predicciones de un modelo en términos de sus características. Layerwise Relevance Propagation (LRP) es otra técnica interesante que se podría utilizar. LRP permite determinar qué características en un vector de entrada particular contribuyen más fuertemente a la salida de una red neuronal. Esta técnica fue introducida en [40] y se encuentra disponible en el paquete Keras Innvestigate (<https://github.com/albermax/innvestigate>). El objetivo de utilizar estas técnicas es que la explicabilidad del modelo pueda mejorar un sistema de ayuda al diagnóstico precoz de AD basado en la arquitectura InceptionResNetV2.

Otra línea futura interesante sería el desarrollo de arquitecturas que, en lugar de trabajar con las imágenes adquiridas en las diferentes visitas como datos independientes, tuvieran en cuenta la información de la evolución temporal de los datos adquiridos en dichas visitas.

Como conclusión final, a pesar de no haber podido reproducir el estudio realizado en [1] debido a una falta de sistematización por parte de los autores del almacenamiento de los datos y resultados del estudio, se ha conseguido implementar un sistema capaz de identificar pacientes que evolucionarán a AD con una antelación de hasta 5 años y medio al diagnóstico final, siendo el preprocesado y la arquitectura de la red mucho más sencillas que otros modelos que forman parte del estado del arte. Cabe destacar la gran utilidad que puede tener este modelo y las posibles mejoras del mismo como herramienta de ayuda al diagnóstico precoz de la enfermedad de Alzheimer.

ANEXO A: Fundamentos de Deep-learning

Deep-learning

Deep learning es un conjunto de algoritmos de aprendizaje automático que, a partir de un dato de entrada, normalmente expresado en forma matricial o tensorial, predice un cierto valor de salida. En el aprendizaje supervisado, se parte de un conjunto de datos de entrada y un vector con los valores de salida, y el algoritmo trata de realizar las aproximaciones necesarias para que los valores de salida predichos correspondan a los valores esperados (aquellos del vector de salida). La forma de hacer esto se consigue a partir de las redes neuronales, que permiten realizar transformaciones no lineales múltiples e iterativas de los datos de entrada. El término deep hace referencia al elevado número de capas ocultas que tienen estas redes, en comparación con las redes neuronales convencionales.

Redes neuronales convencionales

Las redes neuronales son estructuras compuestas de capas de neuronas conectadas entre sí, el número de capas puede variar desde una única capa a cientos o incluso miles. La primera capa se conoce como capa de entrada ya que es la capa que recibe todos los datos de entrada, el tamaño de esta capa dependerá de la dimensionalidad de los datos, y la última capa recibe el nombre de capa de salida, puesto que es la capa que hace el cálculo de las predicciones, esta capa puede estar compuesta de una única neurona o de múltiples neuronas. Entre la capa de entrada y de salida puede haber una o más capas intermedias, estas son denominadas como las capas ocultas de la red. En la figura A.1 se puede observar un ejemplo de una red neuronal convencional.

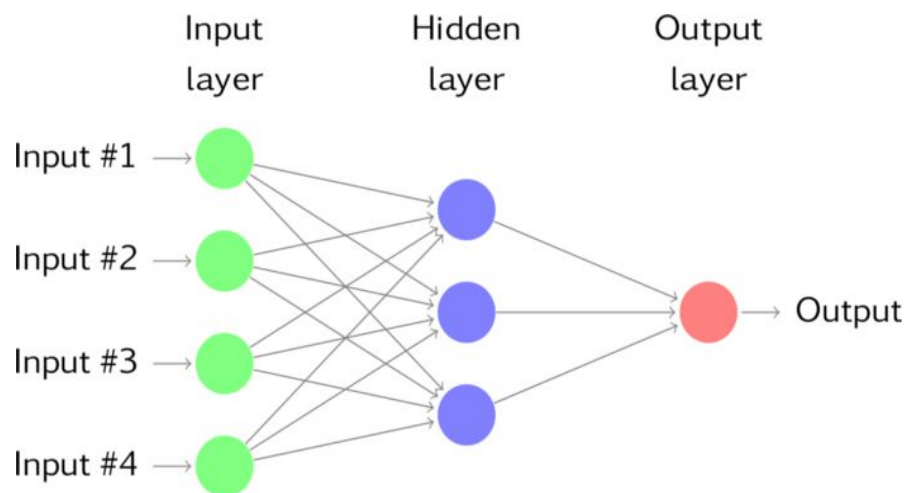


Figura A.1: Red neuronal convencional con una capa oculta y una única neurona en la capa de salida. Imagen obtenida de: <https://becominghuman.ai/artificial-neural-networks-and-deep-learning-a3c9136f2137>.

Una neurona artificial (figura A.3) toma como entrada un vector de números reales x de tamaño m , y produce un único valor de salida y . Cada neurona está compuesta de un vector de pesos w del mismo tamaño que los datos de entrada, un valor denominado bias b y una función no lineal conocida como función de activación. La salida producida de una neurona viene determinada por la siguiente ecuación:

$$y = \phi\left(\sum_{i=0}^m x_i \cdot w_i + b\right)$$

La función de activación es el elemento que permite al modelo resolver problemas no triviales y aprender funciones complejas. Únicamente se utilizan como funciones de activación aquellas que no son lineales, esto es porque con una función lineal la red, por muchas capas ocultas que tenga, es equivalente a un modelo con una única capa [13]. Existen multitud de funciones de activación, en la figura A.2 se muestran ejemplos de las más utilizadas.

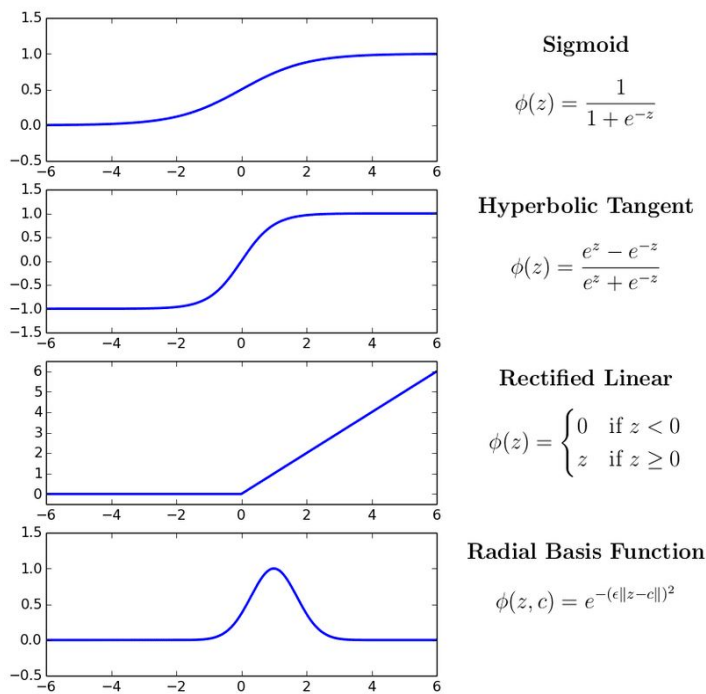


Figura A.2: Funciones de activación más comunes. Imagen obtenida de: https://www.researchgate.net/figure/Common-neural-network-activation-functions_fig7_305881131.

Una red neuronal necesita ser entrenada para que ésta pueda hacer correctamente las predicciones necesarias, es decir, necesita ajustar los vectores de pesos y bias de cada neurona a los valores apropiados a través de un algoritmo iterativo.

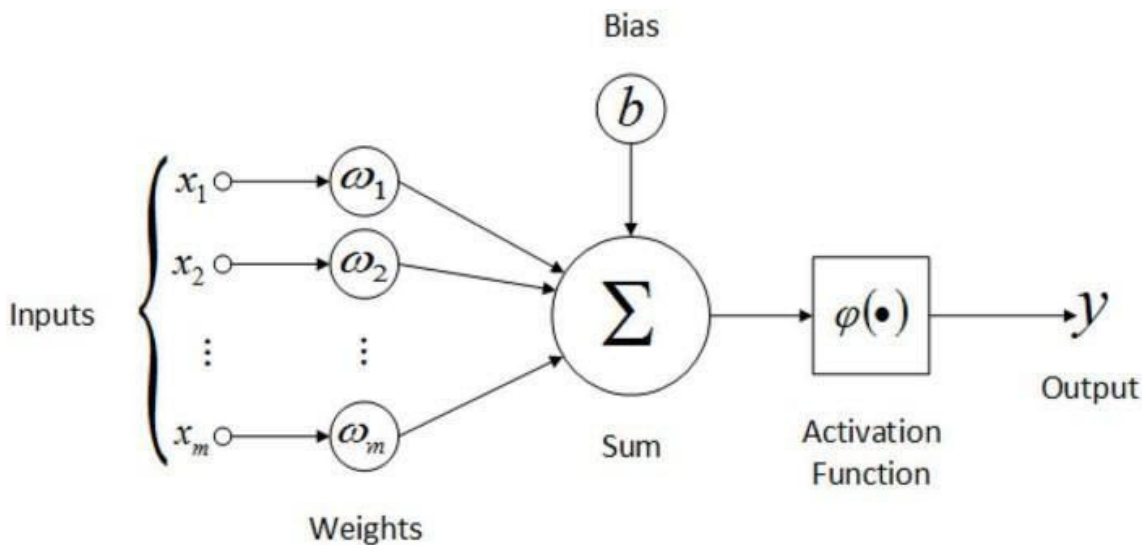


Figura A.3: Estructura básica de una neurona artificial. Imagen obtenida de: <https://medium.com/@jayeshbahire/the-artificial-neural-networks-handbook-part-4-d2087d1f583e>.

Entrenamiento de una red neuronal

Con el entrenamiento de una red neuronal se quiere conseguir que la red realice predicciones más precisas conforme el entrenamiento “avanza”. El cómo de buena es una red viene definido por una función de coste o función de error, de este modo, el entrenamiento de una red se modela como un problema de minimización de dicha función de coste, esto se consigue ajustando los parámetros de la red (los pesos y bias de las neuronas). La más utilizada se conoce como *cross entropy* y viene definida por la siguiente ecuación:

$$J(W) = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)$$

Donde $J(w)$ es la función de coste de los pesos de la red, N es el número de muestras, y es el vector de los valores correctos y \hat{y} es el vector de los valores predichos por la red.

El entrenamiento de la red se realiza mediante un algoritmo de optimización basado en la mayoría de los casos en el método iterativo de gradiente descendente estocástico (Stochastic gradient descent, SGD). El algoritmo calcula en cada iteración el gradiente de la función de coste de un subconjunto de datos llamado batch. El gradiente da información sobre la dirección y la tasa de aumento más rápido de la función de coste. Esto se calcula con el algoritmo de backpropagation. Con backpropagation se calcula el gradiente de la función de coste con respecto a cada peso de la red siguiendo la regla de la cadena. Se calcula el gradiente de una capa cada vez, empezando por la última capa, y se itera hacia las capas más superiores para evitar cálculos redundantes de términos intermedios [14]. Una vez calculado el gradiente, se actualizan los pesos de la red según la fórmula de cada algoritmo, en el caso de descenso de gradiente se actualizan según la siguiente fórmula:

$$W_{i+1} = W_i - \alpha \frac{\partial J(W)}{\partial W_i}$$

donde W_i son los pesos de la i -ésima capa, y el factor de aprendizaje del entrenamiento. El factor de aprendizaje determina el tamaño de los pasos hacia el mínimo de la función de coste en cada iteración. Si es muy pequeño, el entrenamiento puede durar demasiado y no converger en un tiempo razonable, si es demasiado grande se puede omitir el mínimo lo que supone un fallo en la convergencia o incluso divergir.

Con los pesos actualizados, se vuelve a repetir el proceso iterativamente hasta que la función de coste sea lo suficientemente pequeña, es decir, haya convergido. A modo de resumen, en cada iteración se realizan estos cuatro pasos:

1. Calcular las predicciones a partir de los datos de entrenamiento.
2. Calcular la función de coste con las predicciones calculadas.
3. Calcular el gradiente de la función de coste.
4. Actualizar los pesos de cada capa de la red.

Una vez la función de coste ha convergido se para el entrenamiento. No obstante, no siempre se obtiene una red bien entrenada. Se pueden dar ocasiones en las que la red se ha ajustado excesivamente a los datos que ha visto durante el entrenamiento, por lo que no es capaz de generalizar correctamente, es decir, cuando ve una muestra que no había sido utilizada en el entrenamiento no suele hacer una buena predicción. Este fenómeno se denomina como *overfitting*. En las figuras A.4 y A.5 se presentan los tres casos que se pueden dar al terminar el entrenamiento de una red.

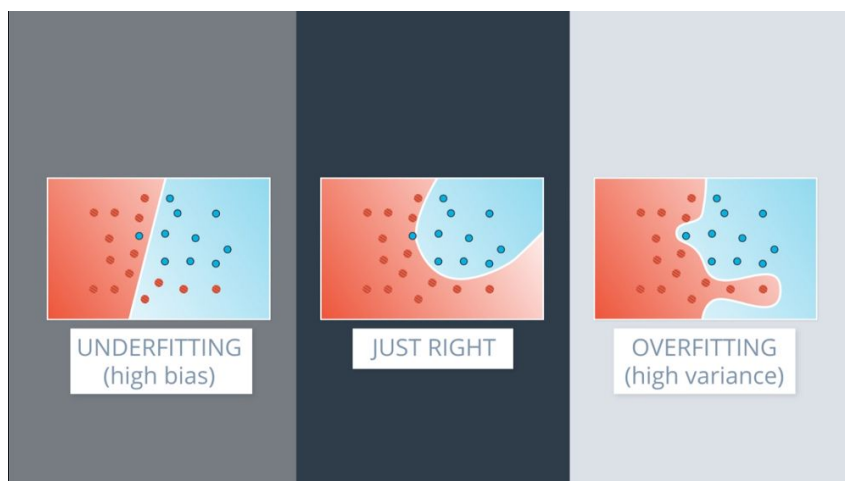


Figura A.4: Underfitting, overfitting y buen entrenamiento al hacer la clasificación de dos clases (roja y azul). Imagen obtenida de: <https://towardsdatascience.com/overfitting-vs-underfitting-ddc80c2fc00d>.

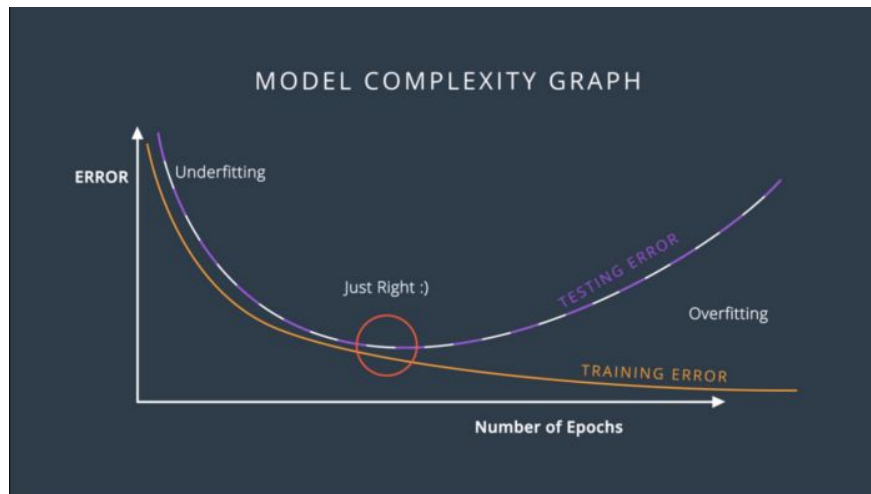


Figura A.5: Underfitting, overfitting y buen entrenamiento, función de coste en función del número de épocas. Imagen obtenida de: <https://towardsdatascience.com/overfitting-vs-underfitting-ddc80c2fc00d>.

Las técnicas que se suelen utilizar para resolver este problema son las siguientes:

- Aumentar el número de muestras disponibles para el entrenamiento, esta es la mejor opción a utilizar si se disponen de los suficientes datos, lo que no suele ser el caso general. Conforme más elevado sea el número de muestras más capaz será el modelo de generalizar correctamente.
- Regularización, se suele aplicar la regularización L2 a los pesos de la red, penalizando al modelo al utilizar pesos grandes. Esto se hace sumando en la función de coste el cuadrado de todos los pesos multiplicados por un parámetro. El valor de dicho parámetro puede influir en la efectividad de la regularización, por lo que para la elección del mismo se suele realizar una búsqueda exhaustiva de un valor óptimo o se estima como hiper-parámetro del modelo.
- Early stopping, esta técnica consiste en parar el entrenamiento antes de lo programado, se suele parar cuando se empieza a detectar overfitting, es decir, cuando el error con los datos de validación no desciende (o incluso aumenta) durante un número prefijado de épocas.
- Dropout, consiste en inhabilitar aleatoriamente un porcentaje de las neuronas de una o más capas concretas de la red durante el entrenamiento, al inhabilitar una neurona ésta se desconecta de la red. Esto permite que no se aprendan detalles redundantes de las muestras de los datos de entrada y no se prioricen ciertos pesos que no benefician al modelo [15].

Convolución

La convolución es la operación principal de estas redes, esta operación necesita un dato de entrada (imagen o volumen), una matriz denominada kernel o filtro que tiene la misma dimensionalidad que el dato de entrada, aunque un tamaño mucho menor, y un parámetro que determina el desplazamiento del kernel por la imagen denominado stride. El número de filas y de columnas del kernel es casi siempre impar e iguales, de este modo se asegura que siempre exista un elemento central. Esta

operación es muy útil para detectar ciertos patrones en las imágenes, por ejemplo contornos verticales u horizontales. El patrón que será reconocido dependerá de los valores almacenados en el kernel.

El resultado de aplicar la convolución es otro volumen de igual dimensionalidad, donde cada valor almacena el resultado de sumar los valores obtenidos al hacer la multiplicación elemento a elemento entre el kernel y una submatriz del volumen de datos de entrada, esta submatriz es del mismo tamaño que el kernel.

Si el volumen de datos es una imagen en escala de grises (N filas x N columnas x 1 dimensión), la convolución se realizará de la siguiente manera. El kernel se colocará en la posición (0,0) de la imagen, se realizará la multiplicación elemento a elemento del kernel con la submatriz correspondiente de la imagen (la submatriz será aquella que vaya desde [0,0] hasta [$krows$, $kcols$], donde $krows$ y $kcols$ son el número de filas y de columnas del kernel respectivamente), estos valores obtenidos se sumarán dando como resultado un único valor, que será almacenado en la posición (0,0) de la imagen de salida. Después se avanzará el kernel un número de columnas igual al parámetro stride por la imagen y se repetirá este mismo proceso, cuando se llega a la última columna de la imagen, se avanza una fila y se vuelve a empezar desde la primera columna.

Si en lugar de una imagen fuera un volumen de datos de dos o más dimensiones, se aplicaría esta operación para cada dimensión. En la figura A.6 se muestra un ejemplo gráfico de una convolución sobre una imagen 7x7 con un kernel 3x3 y stride = 1.

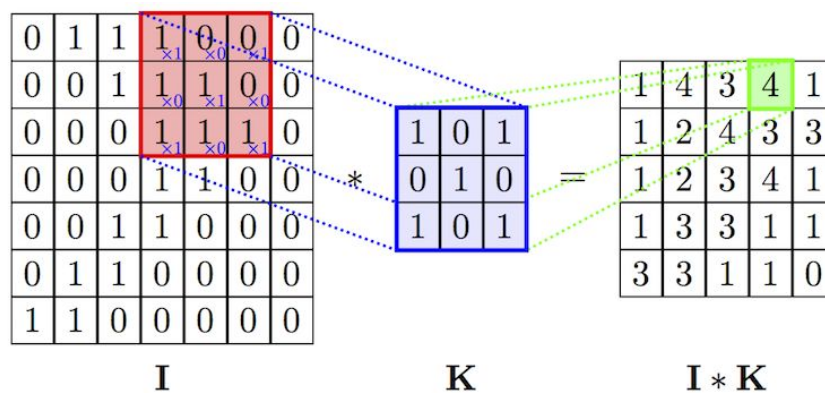


Figura A.6: Ejemplo de una convolución 2D, I corresponde a la imagen de entrada, K al kernel y I*K el resultado obtenido al aplicar esta operación. Imagen obtenida de: <https://www.cnblogs.com/gogoSandy/p/8684501.html>.

El tamaño de la imagen de salida depende del tamaño de la imagen de entrada, el tamaño del kernel, el parámetro stride y de otro parámetro adicional conocido como padding. El padding consiste en aumentar el tamaño de la imagen de entrada añadiendo tantas filas y columnas como se especifiquen en los bordes de la imagen, normalmente se añaden ceros. El objetivo de este parámetro es que el tamaño de la imagen de salida sea igual que el de la de entrada al hacer la convolución. El tamaño de la imagen de salida viene determinado por la siguiente ecuación:

$$d = \frac{n + 2p - k}{s} + 1$$

Donde d es el número de filas o columnas final, n es el número de filas o columnas de la imagen de entrada, p es el parámetro padding, k es el tamaño del kernel y s es el parámetro stride.

Capa convolucional

Una red neuronal convolucional está compuesta por múltiples capas convolucionales. Cada capa es una agrupación de convoluciones, compuesta por un cierto número de filtros K , de modo que cada filtro es aplicado a la entrada de esa capa. La salida de la capa será la concatenación de esas K convoluciones. En la arquitectura de la red se debe especificar el número de filtros, su tamaño, y los parámetros stride y padding de cada capa convolucional. Por lo general, se suelen utilizar kernels 3×3 , 5×5 o 7×7 , y en las capas más profundas se suelen utilizar más filtros que en las capas más superficiales.

La red tratará de aprender los valores de los kernels, además de un valor adicional denominado bias, que es sumado al resultado de aplicar la convolución para cada kernel. Adicionalmente, al igual que en las capas de una red neuronal convencional, se suele aplicar una función de activación no lineal a cada valor obtenido tras aplicar las convoluciones. En la figura A.7 se muestra el resultado de aplicar una convolución sobre una imagen RGB con un filtro 3×3 , $\text{stride} = 1$ y $\text{padding} = 0$.

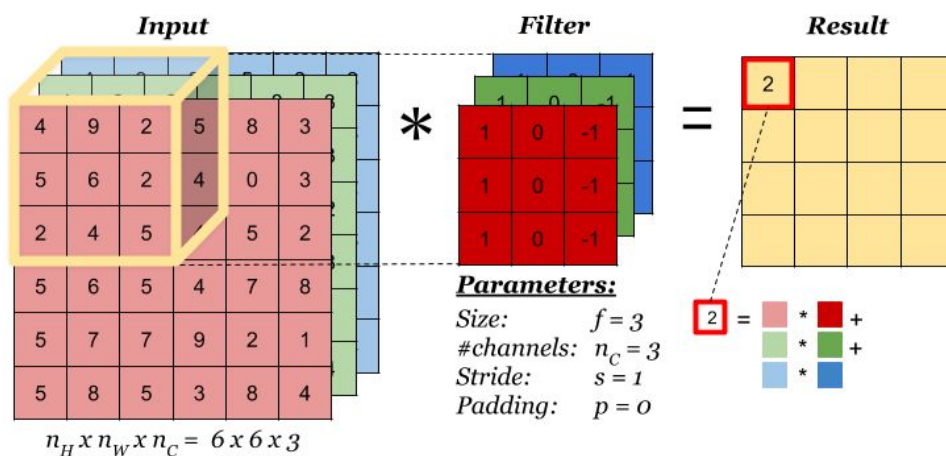


Figura A.7: Ejemplo de una convolución sobre una imagen RGB, se puede observar a la izquierda la imagen de entrada, en el medio el filtro a utilizar y a la derecha el resultado obtenido. Imagen obtenida de: <https://www.cnblogs.com/qogoSandy/p/8684501.html>.

Pooling

Esta operación es muy utilizada en las redes neuronales convolucionales, el objetivo de ésta es reducir el tamaño de las convoluciones obtenidas en las capas convolucionales, disminuyendo así el número de parámetros y operaciones de la red. Existen dos tipos de pooling: Max pooling y average pooling. El funcionamiento de estas operaciones es igual que en las convoluciones, consiste en desplazar un kernel por el volumen de datos al igual que en las convoluciones pero, en lugar de aplicar una convolución, el resultado obtenido es el máximo de los valores cubiertos por el kernel para el caso de Max pooling, o la media de los mismos para el caso de Average pooling.

Es importante destacar que con estas operaciones no hay parámetros a aprender por la red, y que únicamente reducen el tamaño de los datos, pero no la dimensión. En la figura A.8 se muestra un ejemplo al aplicar Max y Average pooling a una imagen 4x4.

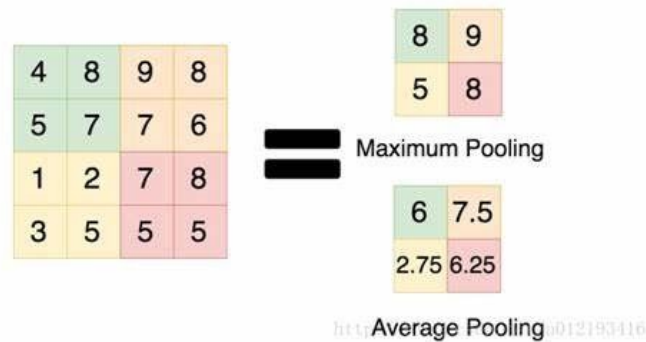


Figura A.8: Operaciones de Max y Average pooling aplicadas sobre una imagen 4x4 (izquierda), el resultado serán imágenes 2x2 (arriba a la derecha tras aplicar Max pooling y abajo a la derecha tras aplicar Average pooling). Imagen obtenida de: <https://blog.csdn.net/u012193416/article/details/79432668>.

Batch normalization

Esta operación se aplica sobre la salida de la capa anterior. Para cada salida (activación) producida en la capa anterior, se tienen dos parámetros γ y β que la red debe aprender. La particularidad de esta operación es que trata todos los datos de cada batch conjuntamente.

Batch normalization empieza calculando la media y la varianza de cada activación del batch de datos. Posteriormente se estandarizan estas activaciones, restando su media y dividiendo por la desviación estándar (raíz cuadrada de la varianza) calculadas anteriormente. Finalmente, se obtiene el resultado a partir de la siguiente ecuación:

$$y = \gamma \cdot \hat{x} + \beta$$

Donde \hat{x} son las activaciones estandarizadas de cada batch, γ y β son los parámetros a aprender por la red. Se ha demostrado que añadir estas capas tras las capas convolucionales o capas completamente conectadas acelera el proceso de entrenamiento, puesto que permite aumentar el valor del factor de aprendizaje, sin que se produzca overfitting tan rápidamente [18].

Capas completamente conectadas

Las capas completamente conectadas (Fully connected layers, FCL) son la base de las redes neuronales convencionales, aunque también son muy utilizadas en las capas finales de las redes neuronales convolucionales para hacer predicciones. Estas capas están compuestas por neuronas artificiales, y toman como entrada un vector unidimensional.

ANEXO B: Métricas de análisis

En este anexo se presenta una explicación de las métricas de análisis utilizadas en este TFG. Entre ellas se encuentran, sensibilidad, especificidad, precisión, F1-Score y AUC. La sensibilidad representa la proporción de pacientes enfermos correctamente identificados, con esta medida se mide la capacidad de un modelo de predicción para dar como casos positivos los casos realmente enfermos [41]. La sensibilidad se define como:

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

Donde TP , son los verdaderos positivos, pacientes enfermos que han sido identificados correctamente como enfermos; y FN , los falsos negativos, pacientes enfermos que han sido identificados como pacientes sanos.

La especificidad representa la proporción de pacientes sanos correctamente identificados, con esta medida se mide la capacidad de un modelo de predicción para dar como casos negativos los casos realmente sanos [41]. La especificidad se define como:

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

Donde TN , son los verdaderos negativos, pacientes sanos que han sido identificados correctamente como sanos; y FP , los falsos positivos, pacientes sanos que han sido identificados como pacientes enfermos.

La precisión se define como la fracción de instancias relevantes entre las instancias recuperadas [42]. En el contexto de pacientes enfermos y sanos, la precisión de un modelo para clasificar pacientes enfermos se define como la fracción de los pacientes enfermos entre los pacientes que el modelo ha clasificado como enfermos. La precisión se calcula a partir de la siguiente fórmula:

$$\text{Precisión} = \frac{TP}{TP + FP}$$

Donde TP , son los verdaderos positivos, y FP , los falsos positivos.

La medida F1-Score es la medida de precisión que tiene un modelo. Se define como la media armónica de la precisión y el recall (equivalente a la sensibilidad). El valor máximo de esta medida es 1, lo que indica una precisión y sensibilidad perfectas, y el valor más bajo posible es 0, se obtiene si la precisión o la sensibilidad son cero [43]. La F1-Score se calcula a partir de la siguiente fórmula:

$$F1 = 2 \cdot \frac{\text{Precisión} \cdot \text{Sensibilidad}}{\text{Precisión} + \text{Sensibilidad}}$$

Finalmente, el área bajo la curva ROC (AUC) es igual a la probabilidad de que un modelo clasifique una instancia positiva (paciente enfermo) elegida aleatoriamente por encima de una negativa (paciente sano) elegida aleatoriamente. Esta medida es muy utilizada en aprendizaje automático para comparar distintos modelos y se calcula obteniendo el área que cubre la curva ROC. Esta curva representa la capacidad de diagnóstico de un clasificador binario a medida que varía su umbral de discriminación, y se crea trazando la sensibilidad contra la especificidad en varios valores de umbral [44].

ANEXO C: Proyecto ADNI

Todos los datos utilizados en este TFG han sido obtenidos de la base de datos de la Alzheimer's Disease Neuroimaging Initiative (ADNI). La Iniciativa ADNI es un estudio multicéntrico longitudinal diseñado para desarrollar biomarcadores clínicos, de imágenes, genéticos y bioquímicos para la detección temprana y el seguimiento de la enfermedad de Alzheimer (AD). Desde su lanzamiento hace más de una década, la histórica asociación público-privada ha realizado importantes contribuciones a la investigación de AD, permitiendo el intercambio de datos entre investigadores de todo el mundo [45]. El acceso a los datos de ADNI es gratuito, aunque es necesario realizar una solicitud que es concedida en función de la investigación propuesta. Hasta la fecha, ADNI ha resultado clave en el estudio de la enfermedad de Alzheimer, con un total de 1800 publicaciones científicas donde se ha hecho uso de sus datos.

ADNI comenzó en 2004 bajo el liderazgo del Dr. Michael W. Weiner, financiado como una asociación público-privada con \$27 millones aportados por 20 empresas y dos fundaciones a través de la Fundación para los Institutos Nacionales de Salud y \$40 millones del Instituto Nacional sobre el Envejecimiento. El estudio inicial de cinco años (ADNI-1) se extendió por dos años en 2009 mediante una subvención de Grand Opportunities (ADNI-GO), y en 2011 y 2016 mediante nuevas renovaciones competitivas de la subvención ADNI-1 (ADNI-2, y ADNI-3, respectivamente).

Los participantes de ADNI son seleccionados entre ciudadanos de Estados Unidos y Canadá, con un rango de edad entre 50 y 90 años. En estos pacientes se incluyen individuos sanos, que sirven como datos de control, pacientes que padecen la enfermedad de Alzheimer y pacientes con un deterioro cognitivo leve. En la actualidad ADNI sigue estando activo, y ya cuenta con un total de más de 1800 participantes, divididos a lo largo de cuatro fases de captación de los mismos, ADNI1, ADNI-GO, ADNI2 y ADNI3. Cada una de estas fases incluye protocolos actualizados y un aumento de los datos recopilados.

De estos participantes se realiza un seguimiento de hasta cuatro años para ver la evolución de la enfermedad, realizando evaluaciones cada tres meses aproximadamente. En estas evaluaciones se recopilan multitud de datos que se consideran podrían servir como biomarcadores de la enfermedad. Entre estos datos encontramos gran variedad de datos clínicos, datos genéticos e imágenes de los participantes tanto de tipo MRI como PET, aunque con menos frecuencia que las evaluaciones periódicas.

ANEXO D: PET en ADNI2

Uno de los principales objetivos de ADNI es realizar imágenes de amiloide Florbetapir F 18 (AV-45) e imágenes de fluorodesoxiglucosa (FDG) en todos los participantes recién inscritos en ADNI2, así como en los participantes de seguimiento de ADNI1 y ADNI GO. Se realizan imágenes de PET AV-45 y FDG PET en todos los participantes recién inscritos en 2 días separados (se requiere un mínimo de 12 horas entre exploraciones). Las exploraciones se pueden realizar en cualquier orden, pero ambas deben completarse dentro de las 2 semanas antes o 2 semanas después de las evaluaciones en la clínica al inicio (Baseline) y en la segunda visita anual, 24 meses después del Baseline [46].

Dado que el escaneo PET no se realizó bajo ADNI1, los participantes que continúan inscritos originalmente bajo ADNI1 que van directamente a ADNI2 tendrían el escaneo FDG y AV-45 en la visita inicial ADNI2 y 24 meses después de la visita inicial, ya que estas las exploraciones se consideran un par y deben ocurrir durante el mismo punto de tiempo. Para los participantes de ADNI1 que continuaron en ADNI GO y luego en ADNI2, el momento de su exploración PET inicial bajo ADNI2 será de 2 años a partir de la fecha de la última exploración PET AV-45 y FDG exitosa bajo ADNI-GO [46]. En la figura D.1 se muestran las reglas temporales de las exploraciones PET.

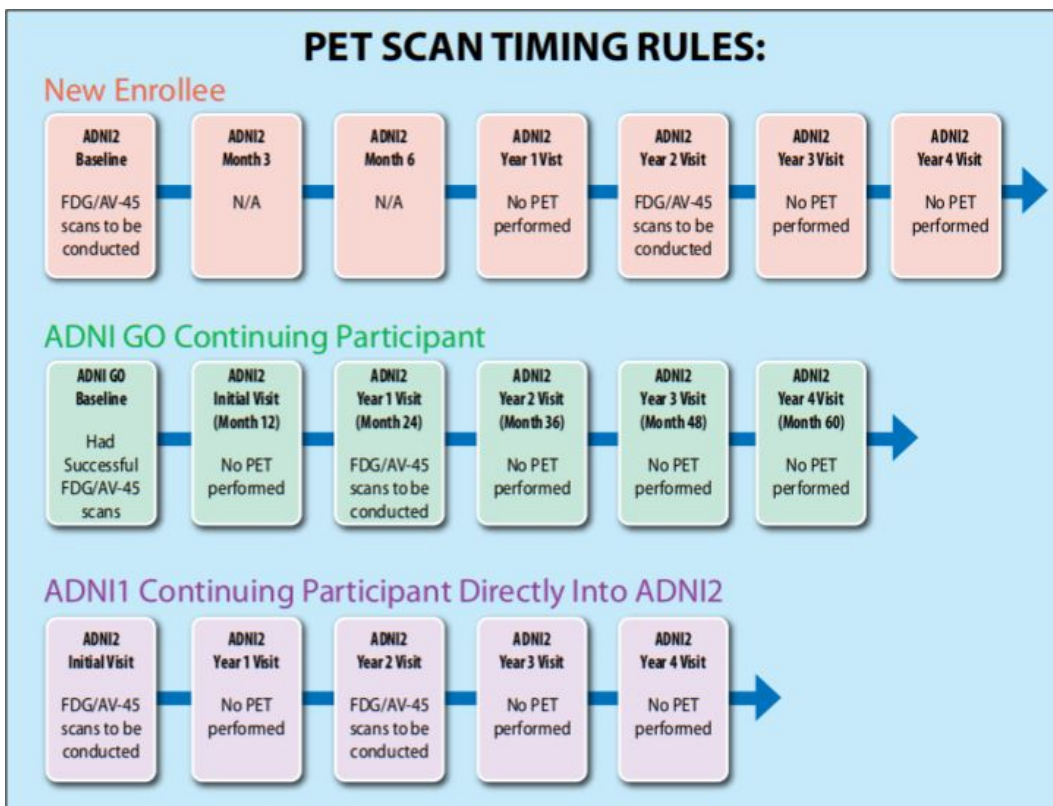


Figura D.1: Reglas temporales de las exploraciones PET. En color naranja corresponden a las reglas para nuevos participantes inscritos en ADNI2, en verde para pacientes provenientes de ADNI-GO y en morado para participantes de ADNI1 que continúan directamente en ADNI2. Imagen obtenida de: <http://adni.loni.usc.edu/wp-content/uploads/2008/07/adni2-procedures-manual.pdf>

El proceso en la exploración FDG PET es el siguiente. Los participantes deberán ayunar al menos cuatro horas antes de la sesión de exploración. La glucosa en sangre de los participantes se controla antes de la exploración y debe ser <180 mg / dL. Cada participante recibirá un único bolo (5 mCi +/- 10%) de fluorodesoxiglucosa antes de la obtención de imágenes de FDG PET, un bolo es la administración de una cantidad discreta de medicamento, fármaco u otro compuesto dentro de un tiempo específico, generalmente entre 1 y 30 minutos, para elevar su concentración en sangre a un nivel eficaz [47]. Las imágenes de FDG PET del cerebro tendrán una duración de 30 minutos, comenzando 30 minutos después de la inyección. Las condiciones ambientales de los escáneres FDG PET son las siguientes:

- Durante la fase de captación, se debe pedir a los sujetos que permanezcan quietos y despiertos con los ojos abiertos mirando al frente (no hacia las luces). Las luces deben atenuarse a un nivel similar al crepúsculo.
- La posición de los sujetos (por ejemplo, sentados o acostados), su entorno visual y la luz ambiental de la habitación deben ser los mismos durante todo el estudio longitudinal.
- El paciente debe ser monitoreado periódicamente para estar seguro de su cumplimiento y para asegurarse de que los ojos no se cierren y el paciente permanezca despierto.

Es muy importante que la posición de los sujetos durante el período de captación, su ambiente, y las condiciones de luz ambiental de la habitación sean las mismas en todos los escaneos del estudio longitudinal. Es importante estandarizar estas condiciones ya que las tomografías por emisión de positrones se realizan durante varios años [46, 48].

El protocolo de obtención de las imágenes FDG PET se realiza siguiendo los siguientes pasos [48]:

- Comprobación de las condiciones del paciente previas a la realización de la exploración mencionadas anteriormente. Si el paciente no cumple las condiciones se deberá esperar a que se cumplan.
- Se debe asegurar que el paciente vacíe su vejiga.
- Permitir que el paciente se acueste cómodamente en una cama o silla reclinable en una habitación en la que el ruido ambiental sea mínimo y el grado de iluminación pueda controlarse y minimizarse como se describió anteriormente. Brindar mantas / almohadas según sea necesario para maximizar su comodidad.
- Obtener acceso intravenoso utilizando una pequeña aguja de mariposa o angiocath. Obtener el nivel de glucosa en sangre de referencia si aún no lo ha realizado.
- Extraer 185 MBq (Becquerel) (5 mCi +/- 10%) de ^{18}F -FDG y analizar con un calibrador de dosis. Registrar el tiempo del ensayo al minuto más cercano.
- Inyectar el ^{18}F -FDG. Enjuagar la jeringa y enjuagar la línea con al menos 10 cc de solución salina normal. Registrar el tiempo de inyección al minuto más cercano. La vía intravenosa se puede suspender en este momento.
- Volver a analizar la jeringa de dosis. Si la actividad residual es 0,1 mCi o mayor, registrar la cantidad y corregir la cantidad de la dosis inyectada para la actividad residual.

- Dejar que el sujeto descanse cómodamente en la habitación durante 20 minutos para la incorporación de ^{18}F -FDG en el cerebro. Durante el período de incorporación, los ojos del paciente deben estar abiertos y los oídos deben permanecer sin ocluir.
- Al final del período de incorporación de 20 minutos, hacer que el paciente use el baño y vacíe su vejiga.
- Colocar y asegurar al sujeto en el escáner usando los métodos descritos anteriormente.
- Adquirir un escaneo 3D dinámico que consta de frames de seis a cinco minutos.
- Todas las imágenes deberán corregirse utilizando la atenuación medida.
 - Adquirir un escaneo de corrección de atenuación utilizando fuentes de varillas durante 5-6 minutos después de la adquisición del escaneo de emisión.
 - Se aplicarán rutinas de segmentación y reproyección para la corrección de atenuación.
- Una vez finalizado, el sujeto puede retirarse del escáner y alentarse a orinar. También se le debe indicar al sujeto que beba muchos líquidos y orine con frecuencia durante el día para ayudar a reducir la exposición a la radiación.
- Reconstruir imágenes utilizando parámetros específicos del sistema utilizado para escanear.
- Una vez completada la reconstrucción, revisar todas las imágenes para evaluar los artefactos y el movimiento.
- Archivar todos los datos del estudio sin procesar y procesados, incluidas copias de la normalización y escaneos en blanco. Es necesario archivar y almacenar datos sin procesar y procesados en el sitio de imágenes durante la duración del proyecto ADNI (aproximadamente 5 años).
- Transferir datos de imágenes al Laboratorio de Neuroimagen (LONI) de UCLA. Cargar sólo el conjunto de imágenes completamente corregido.

Actualmente, hay cuatro tipos de procesamiento de las imágenes PET disponibles en la base de datos ADNI. Los tipos de procesamiento que se enumeran a continuación son incrementales. Los escaneos de pasos anteriores se procesan aún más para proporcionar conjuntos de datos cada vez más uniformes [49]:

1. Co-Registered Dynamic: Las imágenes sin procesar se convierten a un formato de archivo estándar. Se extraen frames separados del archivo de imagen para fines de registro. Se adquieren seis fotogramas de cinco minutos (ADNI1) o cuatro fotogramas de cinco minutos (ADNI GO / 2) de 30 a 60 minutos después de la inyección. Cada fotograma extraído se registra conjuntamente en el primer fotograma extraído del archivo de imagen sin procesar (fotograma adquirido entre 30 y 35 minutos después de la inyección). La imagen del fotograma base y los cinco fotogramas registrados conjuntamente (o todos los fotogramas registrados conjuntamente para los estudios cuantitativos) se recombinan en un conjunto de imágenes dinámicas registradas conjuntamente. Estos conjuntos de imágenes tienen el mismo tamaño de imagen (por ejemplo, $128 \times 128 \times 63$) y dimensiones de vóxeles (por ejemplo, $2,0 \times 2,0 \times 2,0$ mm) y permanecen en la misma

orientación espacial que los datos de imagen PET originales. A esto se le llama espacio "nativo".

2. Co-Registered, Averaged: Este tipo de conjunto de imágenes procesadas se genera simplemente promediando los 6 fotogramas de cinco minutos (o los últimos 6 fotogramas para los estudios cuantitativos) del conjunto de imágenes "Co-registrado dinámico" descrito anteriormente. Esto crea una única imagen PET de 30 minutos fijada todavía en el espacio "nativo".
3. Co-Reg, Avg, Standardized image and voxel size: La imagen promediada co-registrada de cada sujeto de su exploración PET de línea de base se reorienta luego en una cuadrícula de imagen de vóxeles estándar de $160 \times 160 \times 96$, que tiene vóxeles cúbicos de 1,5 mm. Esta cuadrícula de imagen está orientada de manera que el eje anteroposterior del sujeto sea paralelo a la línea AC-PC (figura D.2). Esta imagen estandarizada luego sirve como imagen de referencia para todos los escaneos PET sobre ese tema. Los fotogramas individuales de cada exploración PET (el estudio de línea de base y todos los estudios posteriores, la exploración de 6 meses, la exploración de 12 meses, etc.) se registran conjuntamente en esta imagen de referencia. Al realizar el co-registro de los datos de la imagen original en bruto a un espacio estandarizado en un solo paso, solo se requiere una interpolación de los datos de la imagen y, por lo tanto, la degradación de la resolución por interpolación se mantiene al mínimo y es la misma para todos los escaneos. Se genera una imagen promediada a partir de los fotogramas co-registrados "AC-PC" y luego se normaliza la intensidad utilizando una máscara específica del sujeto, de modo que el promedio de vóxeles dentro de la máscara es exactamente uno. Tanto la orientación espacial (AC-PC) como la normalización de la intensidad de la imagen están pensadas como punto de partida para análisis posteriores. Con una matriz de imágenes estandarizada, los datos PET de diferentes modelos de escáner se pueden comparar más fácilmente. Cabe destacar que en estos conjuntos de imágenes solo se ha producido una reorientación espacial y una normalización de la intensidad de los escaneos. No se ha aplicado a las imágenes ninguna deformación no lineal o incluso escalado lineal de las dimensiones del cerebro

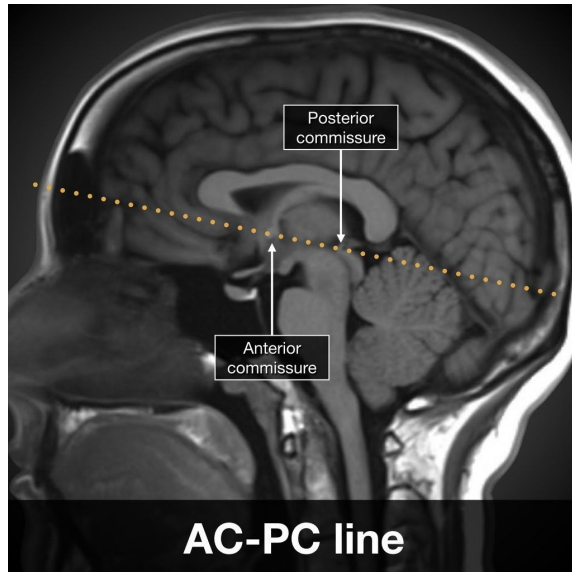


Figura D.2: Vista sagital de una imagen cerebral donde se muestra la línea AC-PC. Imagen obtenida de:
<https://pbs.twimg.com/media/C2wVEuKUsAAxmmM.jpg>

4. Co-Reg, Avg, Std img and vox siz, Uniform Resolution: Estas imágenes son el resultado del suavizado de las imágenes mencionadas anteriormente. Cada conjunto de imágenes se filtra con una función de filtro específica del escáner (puede ser un filtro no isotrópico) para producir imágenes con una resolución isotrópica uniforme de 8 mm FWHM (Anchura a media altura), la resolución aproximada de los escáneres de menor resolución utilizados en ADNI. Los conjuntos de imágenes de los escáneres de mayor resolución se han suavizado más que los conjuntos de imágenes de los escáneres de menor resolución. Las funciones específicas del filtro se determinaron a partir de las exploraciones PET de Hoffman que se adquirieron durante el proceso de certificación.

BIBLIOGRAFÍA

- [1]: Ding, Y., Sohn, J. H., Kawczynski, M. G., Trivedi, H., Harnish, R., Jenkins, N. W., et al. (2019). A deep learning model to predict a diagnosis of Alzheimer disease by using 18F-FDG PET of the brain. *Radiology*, 290(2), 456-464. doi:10.1148/radiol.2018180958
- [2]: World Health Organization. (2019). *Dementia fact sheet*. Retrieved Aug 9, 2020, from <https://www.who.int/news-room/fact-sheets/detail/dementia>
- [3]: Burns, A., & Iliffe, S. (2009). Alzheimer's disease. *BMJ (Clinical Research Ed.)*, 338, b158. doi:10.1136/bmj.b158
- [4]: Morris, J. C., Storandt, M., Miller, J. P., McKeel, D. W., Price, J. L., Rubin, E. H., et al. (2001). Mild cognitive impairment represents early-stage Alzheimer disease. *Archives of Neurology*, 58(3), 397-405. doi:10.1001/archneur.58.3.397
- [5]: Wikipedia contributors. (2020, July 28). Magnetic resonance imaging. In *Wikipedia, The Free Encyclopedia*. Retrieved 10:46, August 9, 2020, from https://en.wikipedia.org/w/index.php?title=Magnetic_resonance_imaging&oldid=970013695
- [6]: Wikipedia contributors. (2020, August 4). Positron emission tomography. In *Wikipedia, The Free Encyclopedia*. Retrieved 10:45, August 9, 2020, from https://en.wikipedia.org/w/index.php?title=Positron_emission_tomography&oldid=971209734
- [7]: Nordberg, A., Rinne, J. O., Kadir, A., & Långström, B. (2010). The use of PET in Alzheimer disease. *Nature Reviews Neurology*, 6(2), 78-87. doi:10.1038/nrneurol.2009.217
- [8]: Cabral, C., Morgado, P. M., Campos Costa, D., & Silveira, M. (2015). Predicting conversion from MCI to AD with FDG-PET brain images at different prodromal stages. *Computers in Biology and Medicine*, 58, 101-109. doi:10.1016/j.combiomed.2015.01.003
- [9]: Shen, T., Jiang, J., Lu, J., Wang, M., Zuo, C., Yu, Z., et al. (2019). Predicting Alzheimer disease from mild cognitive impairment with a deep belief network based on 18F-FDG-PET images. *Molecular Imaging*, 18 doi:10.1177/1536012119877285
- [10]: Pan, X., Adel, M., Fossati, C., Gaidon, T., & Guedj, E. (October 1, 2018). Alzheimer's disease diagnosis with FDG-PET brain images by using multi-level features. Paper presented at the pp. 366-370. doi:10.1109/ICIP.2018.8451824
- [11]: Liu, M., Cheng, D., Yan, W., & Initiative, A. D. N. (2018). Classification of Alzheimer's disease by combination of convolutional and recurrent neural networks using FDG-PET images. *Frontiers in Neuroinformatics*, 12 doi:10.3389/fninf.2018.00035
- [12]: Lee, G., Nho, K., Kang, B., Sohn, K., & Kim, D. (2019). Predicting Alzheimer's disease progression using multi-modal deep learning approach. *Scientific Reports*, 9(1), 1-12. doi:10.1038/s41598-018-37769-z
- [13]: Wikipedia contributors. (2020, August 2). Activation function. In *Wikipedia, The Free Encyclopedia*. Retrieved 16:23, August 15, 2020, from https://en.wikipedia.org/w/index.php?title=Activation_function&oldid=970733589
- [14]: Wikipedia contributors. (2020, August 7). Backpropagation. In *Wikipedia, The Free Encyclopedia*. Retrieved 18:20, August 15, 2020, from <https://en.wikipedia.org/w/index.php?title=Backpropagation&oldid=971681738>
- [15]: Srivastava, Nitish & Hinton, Geoffrey & Krizhevsky, Alex & Sutskever, Ilya & Salakhutdinov, Ruslan. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*. 15. 1929-1958.
- [16]: Bao, H., Li, R., He, M., Kang, D., & Zhao, L. (2019). DTI study on brain structure and cognitive function in patients with chronic mountain sickness. *Scientific Reports*, 9 doi:10.1038/s41598-019-55498-9
- [17]: Huettel, S. A.; Song, A. W.; McCarthy, G. (2009), *Functional Magnetic Resonance Imaging* (2 ed.), Massachusetts: Sinauer, ISBN 978-0-87893-286-3
- [18]: Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift.

- [19]: Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). Rethinking the inception architecture for computer vision.
- [20]: Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition.
- [21]: Wang, M., Liu, B., & Foroosh, H. (2016). Factorized convolutional neural networks.
- [22]: Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2016). Inception-v4, inception-ResNet and the impact of residual connections on learning
- [23]: He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition.
- [24]: Neuroimaging in Python — NiBabel 3.1.1 documentation. <https://nipy.org/nibabel/>
- [25]: Nilearn: Machine learning for NeuroImaging in Python — Machine learning for NeuroImaging. <https://nilearn.github.io/>
- [26]: SciPy.org — SciPy.org. <https://www.scipy.org/>
- [27]: scikit-image: Image processing in Python — scikit-image. <https://scikit-image.org/>
- [28]: TensorFlow. <https://www.tensorflow.org/>
- [29]: Keras: the Python deep learning API. <https://keras.io/>
- [30]: Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62-66. doi:10.1109/TSMC.1979.4310076
- [31]: Wikipedia contributors. (2020, August 26). Principal component analysis. In *Wikipedia, The Free Encyclopedia*. Retrieved 18:35, August 26, 2020, from https://en.wikipedia.org/w/index.php?title=Principal_component_analysis&oldid=974995555
- [32]: Wikipedia contributors. (2020, August 26). T-distributed stochastic neighbor embedding. In *Wikipedia, The Free Encyclopedia*. Retrieved 18:49, August 26, 2020, from https://en.wikipedia.org/w/index.php?title=T-distributed_stochastic_neighbor_embedding&oldid=975053741
- [33]: Spasov, S., Passamonti, L., Duggento, A., Liò, P., & Toschi, N. (2019). A parameter-efficient deep learning approach to predict conversion from mild cognitive impairment to Alzheimer's disease. *NeuroImage*, 189, 276-287. doi:10.1016/j.neuroimage.2019.01.031
- [34]: Liu, Y., Li, Z., Ge, Q., Lin, N., & Xiong, M. (2019). Deep feature selection and causal analysis of Alzheimer's disease. *Frontiers in Neuroscience*, 13, 1198. doi:10.3389/fnins.2019.01198
- [35]: Ramzan, F., Khan, M. U. G., Rehmat, A., Iqbal, S., Saba, T., Rehman, A., et al. (2019). A deep learning approach for automated diagnosis and multi-class classification of Alzheimer's disease stages using resting-state fMRI and residual neural networks. *Journal of Medical Systems*, 44(2), 37. doi:10.1007/s10916-019-1475-2
- [36]: Sarraf, S., & Tofighi, G. (December 2016). Deep learning-based pipeline to recognize Alzheimer's disease using fMRI data. Paper presented at the pp. 816-820. doi:10.1109/FTC.2016.7821697
- [37]: Dukart, J., Kherif, F., Mueller, K., Adaszewski, S., Schroeter, M. L., Frackowiak, R. S. J., et al. (2013). Generative FDG-PET and MRI model of aging and disease progression in Alzheimer's disease. *PLoS Computational Biology*, 9(4), e1002987. doi:10.1371/journal.pcbi.1002987
- [38]: Shivamurthy, V. K. N., Tahari, A. K., Marcus, C., & Subramaniam, R. M. (2015). Brain FDG PET and the diagnosis of dementia. *AJR. American Journal of Roentgenology*, 204(1), 76. doi:10.2214/AJR.13.12363
- [39]: Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks.
- [40]: Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7) doi:10.1371/journal.pone.0130140
- [41]: Wikipedia contributors. (2020, August 24). Sensitivity and specificity. In *Wikipedia, The Free Encyclopedia*. Retrieved 11:30, August 29, 2020, from https://en.wikipedia.org/w/index.php?title=Sensitivity_and_specificity&oldid=974676207
- [42]: Wikipedia contributors. (2020, August 24). Precision and recall. In *Wikipedia, The Free Encyclopedia*. Retrieved 11:47, August 29, 2020, from https://en.wikipedia.org/w/index.php?title=Precision_and_recall&oldid=974612208
- [43]: Wikipedia contributors. (2020, August 27). F1 score. In *Wikipedia, The Free Encyclopedia*. Retrieved 12:01, August 29, 2020, from https://en.wikipedia.org/w/index.php?title=F1_score&oldid=975166309

- [44]: Wikipedia contributors. (2020, August 19). Receiver operating characteristic. In *Wikipedia, The Free Encyclopedia*. Retrieved 12:14, August 29, 2020, from https://en.wikipedia.org/w/index.php?title=Receiver_operating_characteristic&oldid=973746260
- [45]: ADNI | Alzheimer's Disease Neuroimaging Initiative. <http://adni.loni.usc.edu/>
- [46]: ADNI 2 Procedures Manual. <http://adni.loni.usc.edu/wp-content/uploads/2008/07/adni2-procedure-manual.pdf>
- [47]: Wikipedia contributors. (2020, April 28). Bolus (medicine). In *Wikipedia, The Free Encyclopedia*. Retrieved 17:11, August 30, 2020, from [https://en.wikipedia.org/w/index.php?title=Bolus_\(medicine\)&oldid=953680571](https://en.wikipedia.org/w/index.php?title=Bolus_(medicine)&oldid=953680571)
- [48]: ADNI 2 PET Technical Procedures Manual AV-45 (Florbetapir F 18) & FDG V1.0 January 14, 2011. http://adni.loni.usc.edu/wp-content/uploads/2010/05/ADNI2_PET_Tech_Manual_0142011.pdf
- [49]: ADNI | PET Analysis. <http://adni.loni.usc.edu/methods/pet-analysis-method/pet-analysis/#pet-pre-processing-container>