



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIERÍA
INDUSTRIAL VALENCIA

Curso Académico:

Agradecimientos

Con este TFG finalizo una de las etapas más importantes de mi vida, por no decir la que más. El día que decidí venir a Valencia a estudiar en la Universidad Politécnica, sabía que estaba dando un paso importante, que me daba miedo, pero al mismo tiempo, me motivaba. Estos 4 años han sido duros, pero he adquirido una inmensa cantidad de conocimientos que espero poder aplicar en un futuro próximo en mi carrera profesional.

Quería dar las gracias especialmente a mis padres por confiar en mí y motivarme siempre a seguir formándome y estudiando. Muchas gracias también, y en especial, a la gente que he conocido estos años y que la carrera me ha proporcionado, que han vivido conmigo mano a mano cada alegría y cada sufrimiento.

Finalmente, pero no menos importante, muchas gracias al tutor de este trabajo, José Vicente Manjón y a todo el equipo del laboratorio ITACA por haberme ayudado y transmitido sus conocimientos en este tiempo trabajando juntos.

Resumen

En la actualidad, la detección precoz de anomalías en estructuras cerebrales de pacientes se ha demostrado como una manera objetiva y eficiente de ayudar en el diagnóstico más preciso y temprano. No obstante, la cuantificación automática del estado de dichas estructuras requiere del desarrollo de herramientas *software* eficientes, robustas y precisas. En este Trabajo Fin de Grado se ha llevado a cabo el desarrollo de un método de segmentación de estructuras subcorticales basado en técnicas de *Deep Learning*. El conjunto de datos con el que se ha trabajado ha sido imágenes potenciadas en T1 y T2 volumétricas de Resonancia Magnética (RM) de un grupo limitado de sujetos, que se han segmentado, en primera instancia, utilizando el Software 'Freesurfer' para la obtención de las etiquetas de las regiones de interés: Núcleo Talámico, Hipocampo, Amígdala y Tronco Cerebral (Brainstem). Posteriormente, se ha diseñado y validado una red neuronal convolucional (CNN) entrenada con las imágenes del conjunto de sujetos descrito. Los resultados de segmentación obtenidos con la red neuronal, que resulta de un modelo basado en la red U-Net, mejoran la eficiencia de la técnica *single-atlas* (usada en 'Freesurfer'), que realiza una transferencia directa de etiquetas tras el registro entre el atlas y la imagen de estudio.

Palabras clave: Imágenes de Resonancia Magnética (IRM), *Deep Learning*, Redes Neuronales Convolucionales (CNN), segmentación, estructuras cerebrales.

Resum

En l'actualitat, la detecció precoç d'anomalies en estructures cerebrals de pacients s'ha demostrat com una manera objectiva i eficient d'ajudar en el diagnòstic més precís i primerenc. No obstant això, la quantificació automàtica de l'estat d'aquestes estructures requereix del desenvolupament d'eines *software* eficients, robustes i precises. En aquest Treball Fi de Grau s'ha dut a terme el desenvolupament d'un mètode de segmentació d'estructures subcorticals basat en tècniques de *Deep Learning*. El conjunt de dades amb el qual s'ha treballat ha sigut imatges potenciades en T1 i T2 volumètriques de Ressonància Magnètica (RM) d'un grup limitat de subjectes, que s'han segmentat, en primera instància, utilitzant *software 'Freesurfer'* per a l'obtenció de les etiquetes de les regions d'interès: Nucli Talàmic, Hipocamp, Amígdala i Tronc Cerebral (*Brainstem*). Posteriorment, s'ha dissenyat i validat una xarxa neuronal convolucional (CNN) entrenada amb les imatges del conjunt de subjectes descrit. Els resultats de segmentació obtinguts amb la xarxa neuronal, que resulta d'un model basat en la xarxa U-Net, milloren l'eficiència de la tècnica single-atles (usada en '*Freesurfer*'), que realitza una transferència directa d'etiquetes després del registre entre l'atles i la imatge d'estudi.

Paraules clau: Imatges de Ressonància Magnètica (IRM), *Deep Learning*, Xarxes Neuronals Convolucionals (CNN), segmentació, estructures cerebrals.

Abstract

Nowadays, early detection of abnormalities in brain structures in patients has proven to be an objective and efficient way to aid in more accurate and earlier diagnosis. However, the automatic quantification of the state of these structures requires the development of efficient, robust and accurate software tools. In this Final Degree Project we have developed a method for segmenting subcortical structures based on Deep Learning techniques. The data set with which we have worked has been volumetric T1 and T2 enhanced images of Magnetic Resonance Imaging (MRI) of a limited group of subjects, which have been segmented, in the first instance, using the software 'Freesurfer' to obtain the labels of the regions of interest: Thalamic Nucleus, Hippocampus, Amygdala and Brainstem (Brainstem). Subsequently, a convolutional neural network (CNN) trained with the images of the described set of subjects has been designed and validated. The segmentation results obtained with the neural network, which results from a model based on the U-Net network, improve the efficiency of the single-atlas technique (used in 'Freesurfer'), which performs a direct transfer of labels after registration between the atlas and the study image.

Keywords: Magnetic Resonance Imaging (MRI), Deep Learning, Convolutional Neural Networks (CNN), segmentation, brain structures.

Lista de acrónimos

IRM, Imagen de Resonancia Magnética

H, Hidrógeno

RF, Radiofrecuencia

GDL, Grados de Libertad

CC, Correlación Cruzada

IM, Información Mutua

SNC, Sistema Nervioso Central

ROI, *Region of Interest*

EP, Enfermedad de Parkinson

TFG, Trabajo Fin de Grado

RM, Resonancia Magnética

ANTs, *Advanced Normalization Tools*

CNN, *Convolutional Neural Network*

ITK, *Insight Segmentation and Registration Toolkit*

PET, Tomografía por Emisión de Positrones

TAC, Tomografía Axial Computarizada

SPM, *Statistical Parametric Mapping*

fMRI, *Functional Magnetic Resonance Imaging*

RNA, Redes Neuronales Artificiales

ReLU, *Rectified Linear Unit*

TTDO, *Test Time Dropout*

TTDA, *Test Time Data Augmentation*

T, Tesla

MEG, Magnetoencefalografía

rMEG, Magnetoencefalografía de estado de reposo

tMEG, Magnetoencefalografía de tarea

MNI, *Montreal Neurological Institute*

SCP, *Superior Cerebellar Peduncle*

CA, *Cornu Ammonis*

HATA, *Hippocampal to Amygdala Transition Area*

HA, *Hipocampo y Amígdala*

NT, *Núcleo Talámico*

BN, *Batch Normalization*

API, *Application Programming Interface*

GPU, *Graphics Processing Unit*

Índice general

- Documento I: Memoria
- Documento II: Presupuesto



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

Parte I

Memoria

Sánchez Soto, Paula

Manjón Herrera, José Vicente

ÍNDICE DE LA MEMORIA

1. INTRODUCCIÓN	3
1.1 Resonancia Magnética.	3
1.1.1 Física de la RM	3
1.1.2 Adquisición de imágenes potenciadas en T1 y T2	4
1.2 Técnicas de preprocesado de imagen médica	5
1.2.1 Registro	6
1.2.2 Segmentación	7
1.3 Presentación de las estructuras cerebrales de interés	8
1.4 Problema actual de las enfermedades mentales	9
2. ESTADO DEL ARTE.....	11
2.1 Herramientas de registro de imagen médica	11
2.2 Técnicas de segmentación de imagen	12
2.3 Redes Neuronales Artificiales (RNA)	13
2.3.1 Redes neuronales convolucionales (CNN).....	14
2.3.2 Modelo U-Net	16
2.3.3 Capas de la CNN	17
3. OBJETIVOS	18
4. MATERIALES Y MÉTODOS	19
4.1 Database	19
4.2 <i>Software</i> Freesurfer	20
4.2.1 Manejo del <i>software</i>	20
4.2.2 Segmentación de las estructuras.....	22
4.3 Postprocesado de las segmentaciones	29
4.4 Entrenamiento de la Red Neuronal Convolucional (CNN)	30
4.4.1 Data Augmentation	30
4.4.2 Data preparation	31
4.4.3 Training	32
4.4.4 Test.....	34
4.4.5 Modelo de red.....	35

5. RESULTADOS.....	37
5.1 Librerías Keras y Tensor Flow	37
5.2 Resultados del <i>training</i>	37
5.3 Resultados del <i>test</i>	39
5.4 Unión de las redes	42
5.5 Implementación para la mejora de resultados de la red.....	44
6. DISCUSIÓN Y CONCLUSIONES.....	48
7. LÍNEAS FUTURAS	49
BIBLIOGRAFÍA:.....	50

1. INTRODUCCIÓN

La imagen médica ha revolucionado la forma de diagnosticar patologías que afectan al interior de nuestro organismo, ya que han proporcionado una herramienta para la visualización del cuerpo humano, haciendo uso de distintas técnicas de imagen, entre las que se encuentran las imágenes por rayos X, imágenes por ecografía con el uso de ultrasonidos, imágenes de TAC y PET, con radiaciones ionizantes, y, la modalidad que se va a tratar en este TFG, y, que resulta de las más importantes, la Imagen por Resonancia Magnética.

1.1 Resonancia Magnética.

La imagen por resonancia magnética (IRM) es un método tomográfico cuyas principales ventajas sobre otros métodos de adquisición de imagen son: la posibilidad que ofrece para extraer cortes en cualquier dirección del espacio, su gran resolución de contraste, superior a cualquier otra técnica de imagen y su ausencia de efectos nocivos sobre el organismo, al no usar radiaciones ionizantes (Lafuente Martínez & Hernández Moreno, 2017).

1.1.1 Física de la RM

El funcionamiento se basa en la producción de un campo magnético B_0 por parte del imán y la emisión de pulsos de radiofrecuencia (ondas electromagnéticas) por parte de las bobinas de gradiente de campo magnético, que constituyen el aparataje de resonancia. La combinación de ambos componentes va a excitar los núcleos de H del organismo del paciente en la zona de exposición.

Los núcleos tienen un vector M de magnetización, por el simple hecho de estar bajo la influencia de un campo magnético, que se puede separar en dos componentes, M_z , componente longitudinal y M_{xy} , componente transversal (Meléndez Cuauro, 2016). Las partículas nucleares describen un movimiento de precesión por el hecho de no poder alinearse por completo con el campo magnético, presentan una cierta angulación con el eje longitudinal que les inclina a precesar o rotar a una frecuencia determinada. Esta frecuencia de precesión juega un papel fundamental en el fenómeno de resonancia.

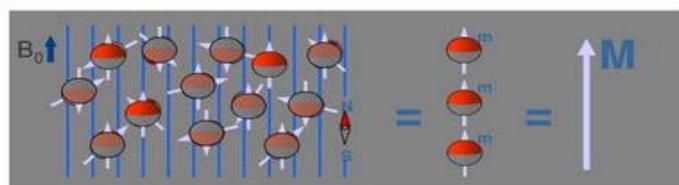


Figura 1: Alineación de los vectores de magnetización de los núcleos de H con el campo magnético B_0 (Meléndez Cuauro, 2016)

Los núcleos que se excitan y absorben energía a la frecuencia de precesión, debido a las ondas de radiofrecuencia, emitirán de vuelta esa energía en forma de señal eléctrica oscilante (eco) (Lafuente Martínez & Hernández Moreno, 2017), quedando en estado de relajación, cuando cese el pulso de RF. Inicialmente, antes del pulso de RF, el vector de magnetización tiene su máxima amplitud en el eje longitudinal, en alineación con el campo magnético del imán. Cuando se activa la onda electromagnética, el vector M en el eje longitudinal pierde amplitud, que comienza a verse reflejada en el eje transversal. El pulso de RF modifica la posición del vector en el espacio, según la angulación que se le dé. Así, en el proceso de relajación tras la excitación, la componente transversal del vector M irá desapareciendo, a medida que adopta su posición inicial de reposo en el eje longitudinal.

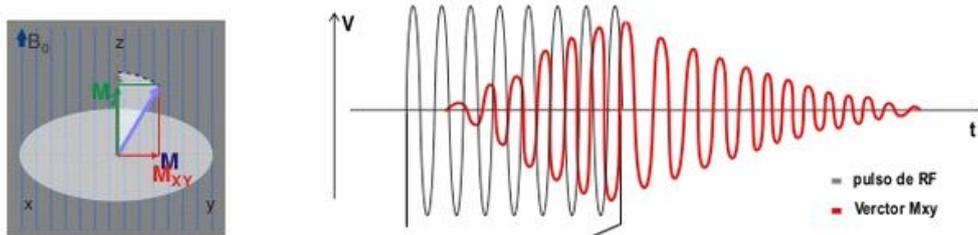


Figura 2: Proceso de relajación de los núcleos. Movimiento sinusoidal descrito en el eje transversal por el vector de magnetización en función del tiempo (Meléndez Cuauro, 2016) .

La IRM es una técnica multiplanar y multidireccional. Se puede seleccionar el corte de interés de manera que sólo se excite un plano concreto. Los volúmenes que se adquieren, como, por ejemplo, un órgano, se suelen obtener mediante la excitación de distintos planos o cortes, normalmente en el eje z, que supone el plano axial, hasta completar la totalidad de la estructura. La selección del plano se produce gracias a la presencia de gradientes. Al gradiente aplicado en el eje z se le conoce como gradiente de corte. Materiales ferromagnéticos en la estructura del imán y la modificación de las corrientes que se hacen pasar por las bobinas (imanes electromagnéticos) de RF, hacen posible que se generen gradientes espaciales en el área de exploración. Así, si se emite una onda a una determinada frecuencia, que coincide con la frecuencia de precesión que tienen los núcleos en un plano, afectado por el gradiente, se consigue excitar sólo la región de interés, dado que los núcleos únicamente absorben energía que posea su misma frecuencia.

1.1.2 Adquisición de imágenes potenciadas en T1 y T2

El contraste obtenido en las imágenes tras su reconstrucción dependerá del tipo de imagen que queramos extraer. En nuestro caso de estudio, vamos a trabajar con imágenes potenciadas en T1 y T2, por ello nos centramos en ellas. Ambas se fundamentan en los tiempos de relajación de los núcleos. En imágenes T1, el tiempo de recuperación de la componente longitudinal T_z es el que más pesa en el valor de intensidad de cada píxel (Meléndez Cuauro, 2016). Por otro lado, en la imagen T2, prima la influencia de la componente transversal T_{xy} . Los tiempos de relajación son exclusivos de cada tejido, ya que dependen del entorno con el que interactúan en el proceso de liberación de energía. A mayor compatibilidad en la frecuencia de precesión entre moléculas, más rápido se produce la transferencia energética. Estos tiempos les aportan un determinado valor de contraste en la imagen a cada tejido.

El tiempo de relajación T1 se define como el tiempo que tarda la componente longitudinal del vector M en llegar al 63% del valor inicial de magnetización, antes de aplicar los pulsos de radiofrecuencia, que hacen que la componente longitudinal disminuya, al mismo tiempo que introducen la componente transversal. El tiempo de relajación T2 es, opuestamente, el tiempo que tarda la componente transversal del vector en decaer al 37% respecto de su valor mientras el pulso de radiofrecuencia se estaba emitiendo.

Las diferencias principales entre imágenes T1 y T2, residen en que, si los tejidos tienen un tiempo de relajación longitudinal T1 largo, se verán hipointensos (cerca del negro o valores bajos de gris) en imágenes T1. Por el contrario, en el caso de tener tiempos de relajación T1 cortos, se verán hiperintensos. En contraposición, para imágenes T2, los tiempos de relajación largos se verán hiperintensos y los tiempos breves de relajación se verán hipointensos. De esta forma, y a modo de ejemplo, el agua libre, al poseer unos tiempos de relajación largos (tanto T1 como T2) se verá hipointensa en imágenes tipo T1 e hiperintensa en imágenes potenciadas en T2 (Gil & Udiat).

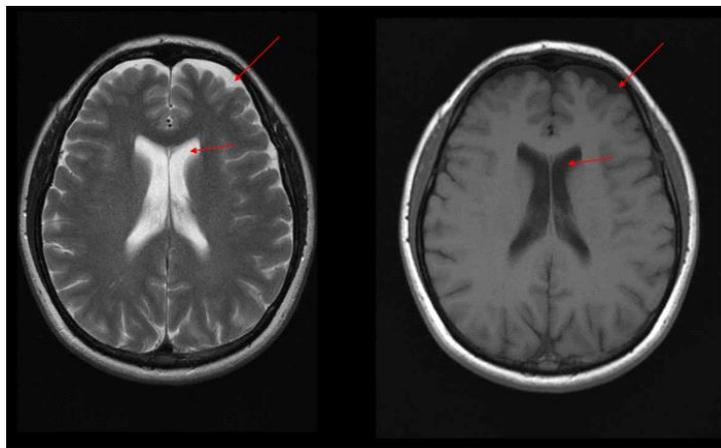


Figura 3: Imagen potenciada en T1 (derecha). Imagen potenciada en T2 (izquierda). Las flechas rojas marcan el agua, hiperintensa en T2 e hipointensa en T1 (Gil & Udiat).

Para la obtención de las imágenes, la energía emitida por los núcleos es recogida por las bobinas receptoras, que pasan esta señal eléctrica por un convertidor analógico digital, de manera que, se muestrea la señal. Estas muestras, codificadas en fase y frecuencia gracias a los gradientes, se almacenan en el espacio-k. El espacio-k es una imagen en el dominio frecuencial, que requiere de la aplicación de la transformada de Fourier para su visualización. Esta conversión o proceso de reconstrucción de la imagen se lleva a cabo en el ordenador, obteniendo así la IMR.

Extraer imágenes de tipo T1 o T2, o cualquier otra modalidad, va a depender del momento en el que se activen las bobinas receptoras para captar la señal en el proceso de relajación de los núcleos.

1.2 Técnicas de preprocesado de imagen médica

Para analizar las imágenes hace falta, previamente, aplicar una serie de técnicas que permiten, por ejemplo, mejorar su calidad o hacerlas comparables mediante un proceso de

estandarización. Estos métodos se agrupan en una clase que describe el preprocesado de imágenes, entre las que se encuentra el registro de las mismas. La segmentación, por su parte, es en sí una técnica de procesado, que se realiza a posteriori del registro.

1.2.1 Registro

Las imágenes médicas, entre ellas las de resonancia magnética, se usan cada vez más para el diagnóstico de enfermedades. Por esta razón, resulta de gran interés relacionar con gran exactitud la información que nos aportan las distintas imágenes, ya que, actualmente, los diagnósticos a partir de imágenes multimodales son los más precisos. En esta tarea de extraer información de las imágenes, se hace necesaria la técnica de registro. Este recurso comprende una alta variabilidad de necesidades, como pueden ser, la alineación de imágenes del mismo sujeto de modalidades distintas; la combinación de imágenes de la misma modalidad separadas temporalmente del mismo paciente o la alineación de imágenes de múltiples pacientes.(Hajnal et al., 2001)

El proceso de registro entra dentro del paquete de preprocesado de los datos, antes de ser analizados, y se lleva a cabo mediante la aplicación de una transformada geométrica a la imagen. Debemos tener en cuenta que siempre registramos respecto de otra imagen, por lo que la aplicación de la transformada se le imputa a la imagen en cuestión que queremos registrar. Existen 4 transformaciones geométricas aplicables:

- Transformación rígida (de tipo lineal): presenta 6 grados de libertad (GDL) y admite las operaciones de translación y rotación en los 3 ejes.
- Transformación global (de tipo lineal): presenta 9 GDL y admite las operaciones de translación, rotación y escalado en los 3 ejes.
- Transformación afin (de tipo lineal): presenta 12 GDL y admite las operaciones de translación, rotación, escalado y *shearing*.
- Transformación *warping* (de tipo no lineal): presenta miles de GDL. Permiten cualquier cambio geométrico en el espacio. Tienen un coste computacional muy elevado.

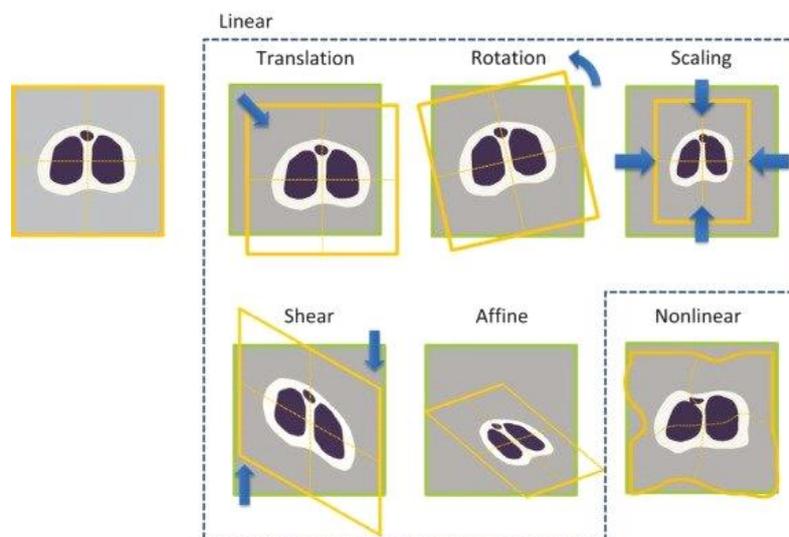


Figura 4: Transformaciones geométricas para el registro de imágenes(Uchida, 2013).

Una forma automática de alinear dos imágenes es basarse en el criterio de intensidad de los píxeles. En este caso, se calcula la transformada geométrica en cuestión, que maximice una función de coste. En relación al tipo de estudio que se esté realizando, si es de tipo intra modal o de tipo intermodal, las funciones de coste que se encuentran son distintas. En imágenes pertenecientes a un mismo sujeto y de una única modalidad, el cálculo de la Correlación Cruzada (CC) es un buen punto de partida, ya que las características de las imágenes serán muy similares y se podrá maximizar ese valor de correlación para el alineamiento.

De manera opuesta, si se dispone de imágenes multimodales, el cálculo del parámetro CC no resulta coherente, dado que las imágenes presentan unas características totalmente dispares y no están correladas (al menos no totalmente). A pesar de ello, internamente, las imágenes sí que guardan cierta relación, ya que representan la misma región de estudio del paciente. En estos casos, la métrica de Información Mutua (IM) (1), resulta de gran interés.

$$I(A,B) = H(A) + H(B) - H(A,B) \quad (1)$$

La IM se basa en el concepto de entropía. Para maximizar el valor del parámetro, la entropía conjunta de las imágenes a registrar, interesa que sea lo mínima posible. Se define entropía de la imagen como el nivel de orden de dicha imagen, siendo la entropía de valor 0, si se tuviera una imagen monocromática, con todos los vóxeles del mismo color (máximo orden). Según la fórmula, cuanto menor sea la entropía conjunta, y por tanto, la información conjunta contenida en la imagen A y la contenida en la imagen B, mayor será la información que comparten las imágenes A y B (Bonsignore, 2004). Si se maximizara el parámetro $H(A,B)$, la información mutua sería muy baja, pues toda la información de las imágenes vendría explicada en cada una de ellas por separado, y no compartirían información. Los datos de entropía se obtienen a partir de los histogramas de las imágenes.

Para finalizar la parte de registro, cabe mencionar que, este procedimiento también puede llevarse a cabo con un atlas, alineando la imagen de estudio con un mapa probabilístico de tejidos, bien obtenido por promedio de imágenes de una población, o basado en la segmentación de un caso concreto (Manjón Herrera, J.V., 2021).

1.2.2 Segmentación

La segmentación, por su parte, constituye una de las tareas más críticas del análisis de imágenes. Se trata de subdividirla en las diferentes partes que la constituyen y extraer aquellas que resulten de interés (Zhang, 1996).

El proceso de segmentación puede realizarse de forma manual o de forma automática. Llevar a cabo este trabajo manualmente resultaría verdaderamente tedioso, ya que se estaría hablando de clasificar cada uno de los píxeles dentro de un grupo o cluster, hasta completar la totalidad de la imagen. En el caso de tener un volumen considerable de imágenes, esta tarea resulta imposible. Por ello, actualmente, se trabaja con algoritmos de segmentación automática, que son capaces de diferenciar cada una de las partes de una imagen basándose en un conocimiento a priori del que se le dota.

En la práctica, para evaluar la calidad de un algoritmo de segmentación se miden los errores en la imagen segmentada, ya que todo algoritmo automático, es susceptible de cometer errores. La

diferencia entre la imagen segmentada automáticamente y la segmentada por un experto con la ayuda de la inspección visual, nos dará el rendimiento de dicho método. (Zhang, 1996) Si la distancia entre la imagen segmentada y la de referencia es aceptable, tendremos un buen algoritmo de segmentación automática.

Con el tiempo, se ha comprobado que la segmentación de imágenes basada en un único parámetro, como es la intensidad de los píxeles, resulta deficiente. Numerosos píxeles pueden poseer el mismo valor de intensidad y no por ello pertenecer a la misma región de la imagen, siendo así píxeles que no deben ser incluidos en ese grupo en concreto. A modo de respuesta a esta carencia, se introdujo también la información espacial. La localización de los píxeles de una imagen, junto a su valor de brillo, representa un dato mucho más completo y valioso, capaz de clasificar un píxel correctamente.

Las tecnologías de clustering basadas en la distribución de intensidades para la clasificación de píxeles como k-means o mixtura de gaussianas, se pueden combinar con la segmentación basada en atlas, que aporta información geométrica. Estos métodos se tratarán posteriormente.

1.3 Presentación de las estructuras cerebrales de interés

En el desarrollo de este trabajo, el foco de interés es segmentar, concretamente 4 regiones del cerebro: tronco encefálico (brainstem), hipocampo, amígdala y núcleo talámico. Hipocampo y amígdala se segmentarán juntos por proximidad espacial y conexión funcional.

El tronco encefálico es un área que está en la base del cerebro, entre las estructuras profundas de los dos hemisferios cerebrales y la médula espinal, y que juega un papel muy importante en el ordenamiento de acciones involuntarias, como la respiración (Joynt). Se divide en 3 secciones principales: mesencéfalo, protuberancia y bulbo raquídeo.

El hipocampo, por su parte, está ubicado en la parte interior del lóbulo temporal, situado a su vez, en el lateral inferior del encéfalo. Cada encéfalo tiene dos hipocampos, uno para el hemisferio derecho y otro para el hemisferio izquierdo. A la par, el hipocampo forma parte del sistema límbico, encargado de regular los instintos humanos, las emociones y la personalidad, entre otras. De esta manera, las funciones del hipocampo están muy relacionadas con las recién comentadas: consolidación de aprendizajes, integrando en nuestra memoria informaciones que recibimos a lo largo de nuestra vida; vinculación de los recuerdos con las emociones; almacén de recuerdos; regulación de la percepción de los espacios (navegación espacial) (Ruíz, 2021).

Siguiendo con la amígdala, esta estructura es un grupo de neuronas situada en la parte más profunda del lóbulo temporal. Al igual que en el hipocampo, encontramos un cuerpo amigdalino por hemisferio. Forma parte también, del sistema límbico, y su función está relacionada con la generación y regulación de las emociones. Está constituida por un núcleo central, un núcleo medial y un complejo basolateral (Triglia).

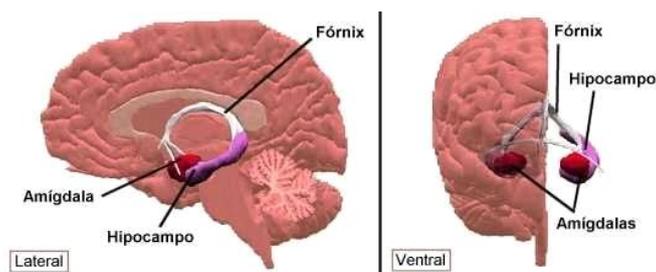


Figura 5: Estructuras del hipocampo y amígdala (Triglia, n.d.)

Finalmente, el núcleo talámico, incluye 4 regiones principales, el núcleo anterior, el medial, el lateral y el ventral. También podemos encontrar los núcleos intralaminares, núcleo de la línea media y núcleo reticular. Las funciones del tálamo son numerosas, entre ellas encontramos el procesamiento de la información sensitiva, memoria, emociones y funciones motoras (Bereciartua Pérez, 2016). Además, recoge una gran parte de la información sensorial, que pasa por esta estructura antes de llegar a la corteza cerebral. Se puede definir como un centro integrador del SNC.



Figura 6: Estructura del tálamo (Bereciartua Pérez, 2016) .

Estas 4 áreas definidas constituyen las regiones de interés (ROIs).

1.4 Problema actual de las enfermedades mentales

Tanto las enfermedades neurodegenerativas, como el simple envejecimiento saludable cerebral, están relacionados frecuentemente con cambios en las estructuras cerebrales. Dichos cambios, provocan cambios en las imágenes que se toman del cerebro, tanto a nivel de propiedades del tejido, como a nivel morfométrico de las estructuras. Los cambios morfométricos hacen referencia a alteraciones de volumen, área, forma e incluso patrón de plegamiento de la corteza (Fischl et al., 2004)

En el caso de este TFG, que pretende centrarse en las estructuras subcorticales mencionadas, un análisis volumétrico resulta de mayor interés que un estudio del área, ya que éste sería más conveniente si se quisiesen medir los cambios en la corteza. La razón por la que son interesantes las estructuras subcorticales y sus volúmenes, es porque existe correlación entre los cambios volumétricos y la aparición de enfermedades neurológicas (Fischl et al., 2004) (Fischl et al., 2002). Precisamente, estructuras como el tálamo, el hipocampo y la amígdala, son susceptibles de cambios en distintas patologías cerebrales (Fischl et al., 2002).

Enfermedades como el Parkinson o el Alzheimer, siendo esta segunda más estudiada a lo largo de los años por científicos y médicos, son diagnosticadas en estadios tardíos a su aparición, razón por la que, en la mayoría de ocasiones, no es posible ralentizar/curar este tipo de enfermedades y la única solución son cuidados paliativos.

La sintomatología inicial de los enfermos de Alzheimer, en este caso, es una pérdida leve de memoria, que puede perdurar durante años y va progresando crecientemente (Fischl et al., 2002). Dado que, las deficiencias leves de memoria son bastante comunes en la población de mayor edad, la tarea de distinguir entre personas con indicios de pérdida de memoria propios del envejecimiento y personas en las que progresará la enfermedad con el paso del tiempo, se hace muy difícil.

Una de las vías de diagnóstico de esta enfermedad es mediante el uso de biomarcadores. Estos biomarcadores son parámetros que se pueden medir y que permiten, gracias al establecimiento de umbrales, determinar si hay una situación potencialmente patológica o no. El problema que esta técnica lleva consigo, es que, en el caso del Alzheimer, hay biomarcadores que también se presentan en personas sanas con valores muy cercanos, como es la acumulación de placas β -amiloides en el área cortical.

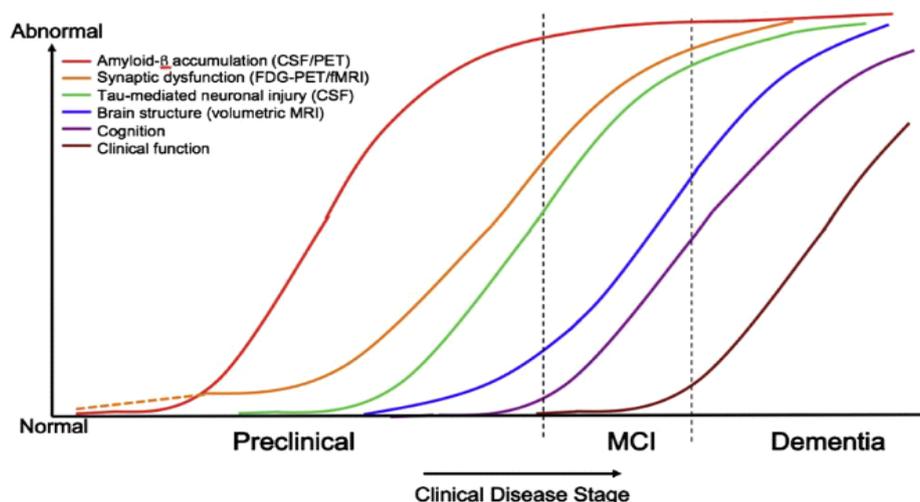


Figura 7: Gráfico de valores de biomarcadores en fases: normal, preclínica, deterioro cognitivo leve, demencia (Manjón Herrera, J.V., 2021).

Por otro lado, en el caso del Parkinson, encontrar biomarcadores que reflejen cambios cognitivos tempranos ha resultado más difícil en todos los años de estudio, ya que los cambios en algunos dominios cerebrales suelen ser más pronósticos de la demencia. El Parkinson es una enfermedad más de carácter motor. No existen muchos estudios que hayan investigado si la resonancia magnética estructural (IRM) se correlaciona con el deterioro de las funciones motoras del enfermo, dado que muchos de los artículos que han tratado el tema han sacado conclusiones contrapuestas (Filoteo et al., 2014).

Un artículo publicado en 2014 por un equipo de la Universidad de San Diego, California, enuncia que, tras el estudio de una serie de sujetos, sometidos a pruebas de fluidez de palabras, aprendizaje verbal, cognición visuoespacial y visoconstrucción, los volúmenes de las estructuras de los dominios responsables de estas funciones (entre los que se encuentran hipocampo y amígdala) están muy relacionados con su capacidad de funcionamiento en pacientes con la enfermedad del

Parkinson (EP) sin demencia (Filoteo et al., 2014). Desmintiendo así, las contradicciones emergidas hasta el momento en artículos anteriores, por posibles diferencias en los métodos de procesamiento de imágenes, tamaño de muestras y características de los pacientes (Filoteo et al., 2014).

Este TFG se centra, por tanto, en la rama de la neurología que investiga el poder identificar a las personas en fase prodrómica de la enfermedad (síntomas iniciales que preceden al desarrollo de la enfermedad) a través de la segmentación eficiente y precisa de imágenes de resonancia magnética y su análisis volumétrico, para conseguir un diagnóstico temprano (Killiany et al., 2000).

2. ESTADO DEL ARTE

2.1 Herramientas de registro de imagen médica

En este apartado, se va a hacer un recopilatorio de algunas de las herramientas disponibles a día de hoy para el registro de imagen médica. Los distintos *software* que se van a tratar constituyen el estado del arte y el conjunto de útiles del que disponen los médicos e investigadores para el proceso de registro de imagen.

- **ITK** (*Insight Segmentation and Registration Toolkit*). Es una librería para la segmentación y registro de imágenes médicas, de tipo MRI, PET, TAC, entre otras. Es un software de código abierto muy utilizado en el ámbito biomédico. El programa permite realizar el registro de imágenes de manera manual o automática. Contiene un paquete con diferentes métodos de registro basados en intensidad. Las funciones de coste a maximizar o minimizar que contempla el programa son correlación cruzada, información mutua y diferencia de intensidad. Por su parte, las transformaciones que ofrece son de tipo afín y rígida (Bereciartua Pérez, 2016).
- **SPM**. Mapeo paramétrico estadístico. Al igual que en el caso anterior, se trata de un software de código abierto, para el análisis de imágenes cerebrales funcionales. Su algoritmo se basa en 4 factores: función objetivo, separación, tolerancia y suavizado del histograma. Para la función objetivo, si se trata de un caso multimodal, el programa utiliza el parámetro de información mutua (normalizada) o el coeficiente de correlación de entropía (Abderrahim et al., 2020). En caso de registro monomodal, correlación cruzada normalizada. La separación o distancia de los puntos de las imágenes debe ser como máximo, de 8 mm en fMRI y 12 mm en PET (Abderrahim et al., 2020). Adicionalmente, el programa realiza un suavizado del histograma, de tipo gaussiano. Para medir la similitud, SPM aplica interpolación trilineal y B-spline (Abderrahim et al., 2020).
- **ANTs** (**Advanced Normalization Tools**). Este *software* es considerado a día de hoy, el estado del arte en registro de imágenes médicas. ANTs comprende las dos aplicaciones generales del registro de imágenes: el mapeo de datos etiquetados de una imagen plantilla en un espacio individual (el del paciente en concreto a analizar), donde es la plantilla o atlas el que se deforma para adaptarse al caso concreto, y la transformación geométrica de la imagen del individuo a la plantilla, que resulta el proceso inverso.

El conjunto de herramientas ANTs, permite al usuario ajustar una diversidad de parámetros, que le permiten establecer una jerarquía de transformaciones con niveles ajustables de complejidad, regularización, grados de libertad y comportamiento como optimizadores (Avants et al., 2002). El tiempo de cálculo depende de la complejidad de la transformación. ANTs permite seleccionar: número de iteraciones de manera convergente; parámetro de similitud y tipo de transformada a realizar, entre otras.

2.2 Técnicas de segmentación de imagen

Eventualmente, existen *software* de segmentación manuales que permiten al profesional seleccionar y distinguir entre regiones diferentes de una imagen, marcando el área distintiva píxel a píxel con herramientas como ITK-snap. El problema que muestran es el gran coste e inversión de tiempo que esto conlleva, y más cuando se trata de volúmenes grandes de imágenes. Se podría considerar que, para casos puntuales, la utilización de estas herramientas es una opción, pero en el día a día de la segmentación de imágenes médicas estas técnicas están ciertamente obsoletas.

Se busca la automatización del método. En base a esta idea, se pueden considerar, por un lado, las técnicas basadas en atlas y por otro, las redes neuronales. Dentro de la ciencia de los atlas, distinguimos entre: *single-atlas*, atlas probabilísticos y multi-atlas.

En la técnica de *single-atlas*, el mapa es una única imagen, tal y como indica su nombre. Carece de información probabilística, pues es la transferencia de etiquetas de la imagen de referencia, el atlas, al caso a segmentar, al no existir mayor comparación con otros ejemplos. Se podría decir que es un caso de registro con una inmediata transferencia de etiquetas. Presenta una gran simplicidad y rapidez en su cómputo, pero carece de variabilidad biológica, únicamente tiene en cuenta la información aportada por el sujeto sobre el que se ha obtenido ese atlas, que pertenecerá a una población concreta, de rango de edad y sexo, entre otras, que pueden no ajustarse del todo bien a las necesidades y características fisiológicas del caso en concreto a segmentar, introduciendo así errores en la estimación.

Por su cuenta, las técnicas de atlas probabilísticos, son un promedio del resultado de un análisis estadístico aplicado a un conjunto de atlas. El mapa final obtenido, contiene información promedio de múltiples atlas distintos, integrando mayor información y dando más cabida a la variabilidad anatómica. Un ejemplo de mapa probabilístico es el atlas MNI152 (resultado del promedio de los atlas de 152 sujetos) (Enrique & Gómez, 2018). La imagen a segmentar y el atlas se registran y se produce la transferencia de etiquetas del atlas a la imagen, de manera probabilística (generalmente usando el teorema de Bayes). Al igual que ocurría con la técnica de *single-atlas*, el traspaso de etiquetas es unívoco de una imagen a otra, pues solo se dispone de una única imagen etiquetada donde la probabilidad de que cada uno de los píxeles pertenezca a una etiqueta es entre 0 y 1.

Las técnicas multi-atlas tiene un enfoque distinto. En esta ocasión se dispone de un conjunto ‘p’ de atlas, en el que cada uno de ellos va a ser partícipe en la transferencia de etiquetas al caso a segmentar. Existen distintos métodos para ello, como es la técnica de *Majority Voting* (Enrique & Gómez, 2018), en la que, cada uno de los atlas presenta un valor de etiqueta para un píxel dado. El valor de etiqueta que salga más votado en el recuento de todos los atlas es la que se asignará al píxel correspondiente en la imagen. Se trata de una transferencia de etiquetas votada, una democracia en la elección, en contradicción a lo que ocurría en *single-atlas*.

Otra técnica de fusión de etiquetas es *Weighted Majority Voting* (Manjón Herrera, J.V., 2021), una implementación del procedimiento recién explicado, en la que cada voto es ponderado de

forma local o global según la similitud del atlas y el caso a segmentar. Esta similitud se mide en base a la maximización de parámetros como CC o IM. El problema que presentan estas técnicas es que requieren de un registro no lineal de la imagen con los atlas, con deformación, pues existe una correspondencia directa en la asignación de etiquetas a cada vóxel, es decir, cada vóxel del atlas se debe corresponder al mismo vóxel de la imagen a segmentar, para que la asignación sea correcta (Enrique & Gómez, 2018). Existen otros métodos más sofisticados como el método Joint Label Fusion, que intentan corregir esta dependencia (Enrique & Gómez, 2018).

Algunos ejemplos de software de segmentación que utilizan las metodologías explicadas son: 3D Slicer, Freesurfer, Volbrain y SPM. Éste último utiliza atlas probabilísticos para la segmentación. Volbrain, por su parte, dispone de una biblioteca de imágenes craneales de tipo T1 segmentadas manualmente por un experto en resonancia magnética cerebral. Esta base de datos de multi-atlas combinada con la técnica de non-local means label fusion para la transferencia de etiquetas de manera local, realizan la segmentación automática de los casos (Manjón & Coupé, 2016).

Un enfoque muy distinto para segmentación es hacer uso de redes neuronales, que en estos últimos años ha ganado terreno a las otras técnicas de segmentación explicadas. En este trabajo precisamente, se aborda una combinación de segmentación con *single-atlas* usando Freesurfer y segmentación con *Deep Learning* usando CNN.

2.3 Redes Neuronales Artificiales (RNA)

Las redes neuronales artificiales (RNA) son una técnica de aprendizaje automático, en concreto, aprendizaje profundo. Presentan analogía en su estructura con las redes neuronales biológicas del cerebro humano. Se puede enunciar que, cada uno de los componentes de la red es una neurona, que se conecta a otras que tiene a su alrededor, con las que intercambia información. De manera superficial, la red se compone de niveles jerárquicos. El nivel jerárquico inicial aprende información simple de la entrada introducida en la red, que se la enviará al siguiente nivel que la compone. Éste nivel tomará la información recibida y compondrá una nueva información de mayor complejidad, para pasársela al tercer nivel, siguiéndose así la dinámica hasta completar todas las capas (Ros García, 2019)

El elemento básico de la red, como se comentaba, es la neurona. Es una unidad que recibe como entrada el sumatorio de todas las entradas multiplicadas por unos pesos y ofrece como salida, una modificación de la entrada en base a una función de transferencia o función de coste que hay que minimizar. La solución óptima al problema tendrá el menor costo posible, por lo que a no ser que la red sea perfecta, la función de coste indicará como de lejos la solución particular al problema propuesta por la red, se encuentra de la óptima.

Entre los parámetros esenciales de la red, y por ello, los que la definen, se encuentran el número de neuronas que la componen, el número de parámetros que baraja y que optimiza en cada iteración, las interconexiones que presenta entre sus neuronas y el número de capas o profundidad de la misma.

Entre las principales ventajas que presentan las redes como método de segmentación, se tiene que, aprenden de la experiencia (Ros García, 2019) a través de ejemplos, de los que extraen las características principales. Generalizan en ejemplos nuevos, lo aprendido de los anteriores, (Ros García, 2019).

2.3.1 Redes neuronales convolucionales (CNN)

La red que se va a utilizar en este TFG es de tipo convolucional, que se compone de la sucesión en cascada de capas simples, donde la salida de una capa es la entrada de la siguiente (Ros García, 2019). Las redes neuronales convolucionales son un tipo de modelos profundos que se utilizan, especialmente, para la clasificación y segmentación de imágenes. En ellas se encuentran 3 tipos de capas: convolucionales, de reducción de muestreo y de clasificación. Adicionalmente, se encuentra la capa de partida, que es la que recibe las matrices de píxeles de las diferentes imágenes y las transfiere al resto de capas.

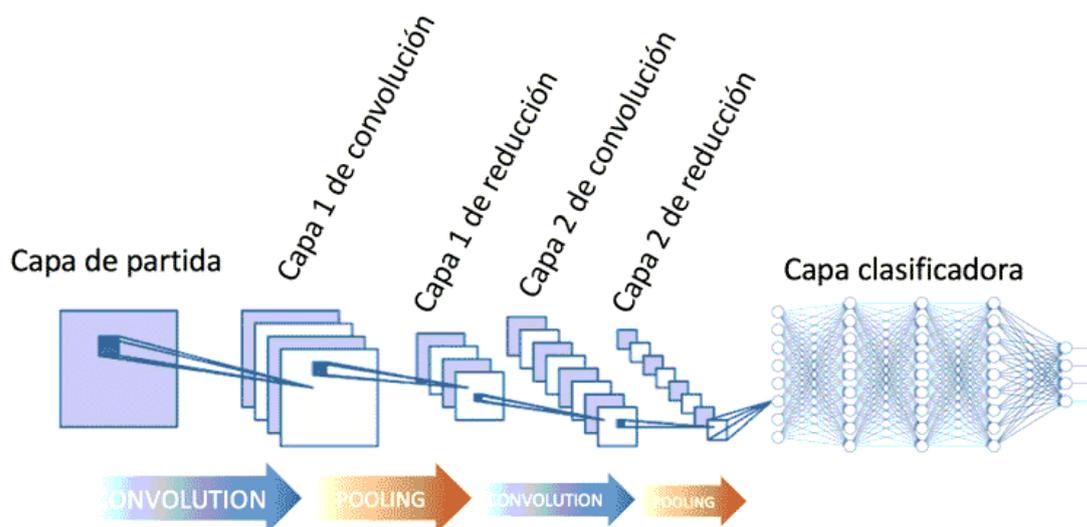


Figura 8: Ejemplo de una red neuronal convolucional (Calvo, 2017).

Las neuronas de convolución aplican filtros entrenables y de vecindad local a las imágenes de entrada. Estos *kernel* actúan como pesos que transforman los datos, de manera que ciertas características se convierten en dominantes en la imagen de salida, por tener un valor mayor que el resto de píxeles de la imagen. El valor de los filtros se va actualizando y optimizando en cada iteración (Wikipedia, 2021).

A estos filtros de convolución se les puede variar la dimensionalidad. Se debe encontrar un equilibrio en el tamaño del *kernel*, ya que unas dimensiones pequeñas del mismo, supondrían un bajo nivel de detalle a la hora de detectar patrones pequeños; por otro lado, si las dimensiones del *kernel* son muy grandes, el coste computacional será excesivo. Generalmente, para el filtrado de imágenes se usan máscaras de orden 3x3x3 o 5x5x5 (Penadés Migallón, 2020). El filtro actúa como una ventana deslizante que va tomando regiones de la imagen de su mismo tamaño, realizando el producto, y, posteriormente, la suma de los resultados, para obtener un único valor numérico representativo de cada región. Este proceso se puede observar en la figura 3.3.2. Un parámetro importante a definir es el valor de paso o *stride*, que indica el número de píxeles que el filtro se desplaza para tomar la siguiente región. El movimiento de avance se realiza de izquierda a derecha y de arriba abajo (Penadés Migallón, 2020) (en 2D).

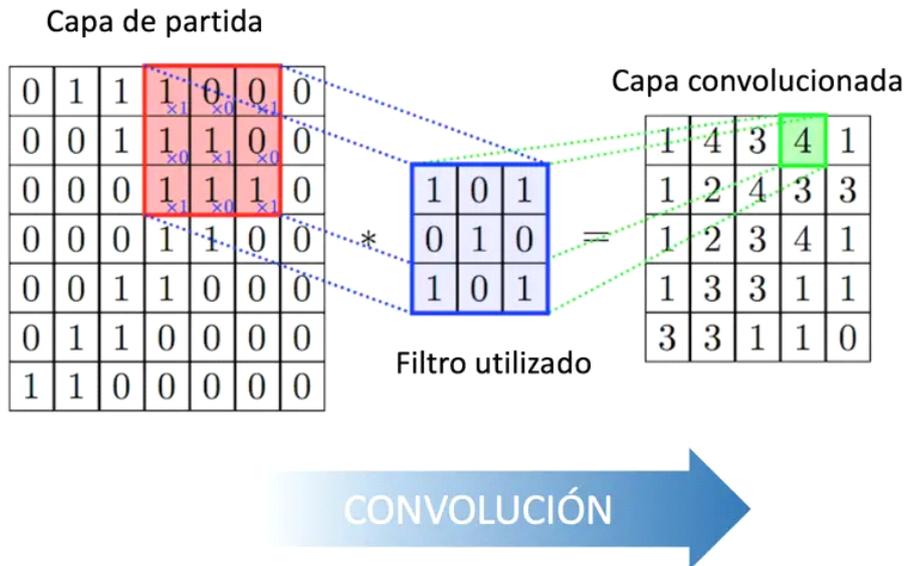


Figura 9: Ejemplo de filtrado de una imagen en una capa de convolución (Calvo, 2017).

Cuando el proceso de convolución acaba, se obtiene una matriz de convolución que representa un mapa de características de la imagen, de dimensiones inferiores a la matriz inicial, al que se le aplica una función de activación (Penadés Migallón, 2020).

Por su parte, las capas de reducción o pooling minimizan el tamaño de los mapas de características, quedándose únicamente con aquellas características realmente relevantes. De esta manera, en las capas más profundas de la red, se encuentra información resumida pero mucho más específica que en las capas iniciales. Se puede definir también, el número de mapas de características o número de filtros que se quiere que la neurona proporcione en cada capa. A mayor número de filtros de salida, mayor información estará extrayendo la red neuronal de la imagen.

Finalmente, a las neuronas de clasificación, llega una serie de características únicas de la imagen de entrada y su trabajo consiste en clasificar estas características con una etiqueta u otra. Habrá tantas neuronas de clasificación como número de etiquetas o clases posibles pueda adoptar el vóxel.

La mejora del rendimiento de las redes se basa en aumentar los conjuntos de datos de entrenamiento, generar modelos de red más potentes y utilizar mejores técnicas para prevenir el sobreajuste (Krizhevsky et al., 2020).

2.3.2 Modelo U-Net

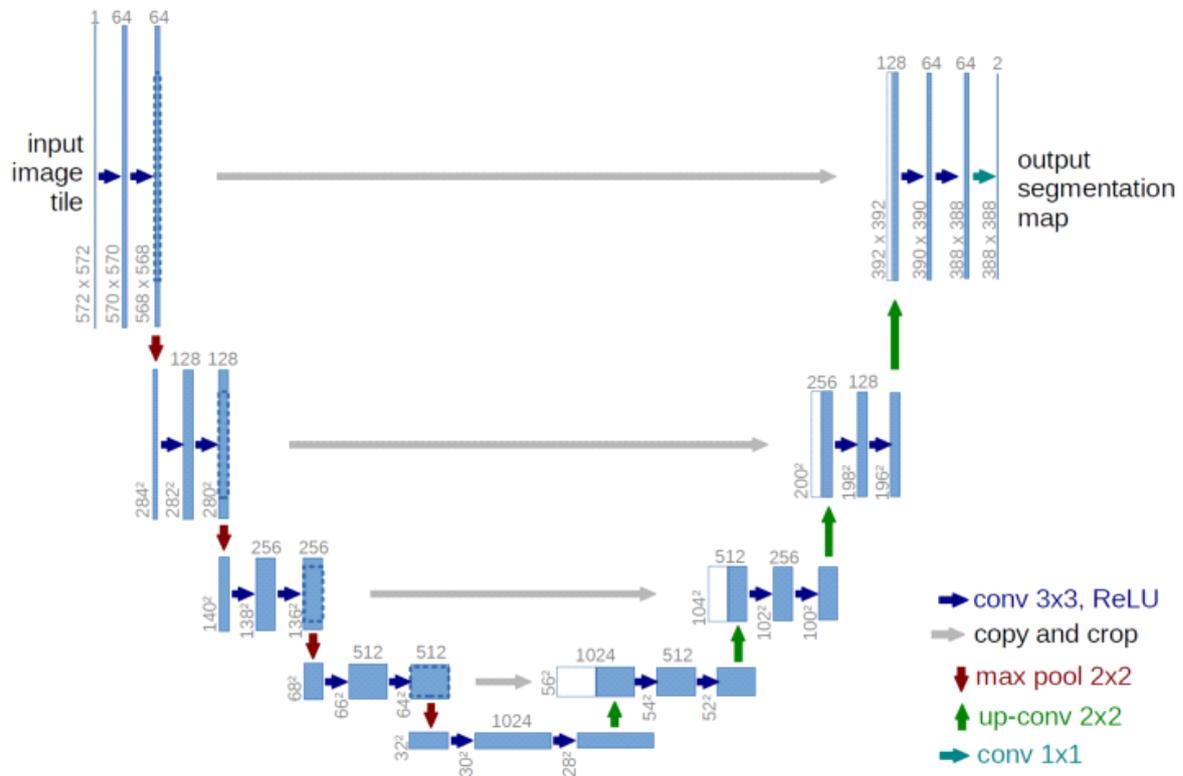


Figura 10: Modelo de CNN U-Net (Krizhevsky et al., 2020).

Creada por Ronneberger, Fischer y Brox en 2015, el modelo de red neuronal convolucional U-Net hace referencia a su morfología, tal y cómo se puede observar en la figura 3.3.2. La idea principal de este tipo de red es incluir en la salida, la información de localización del vóxel, de manera que, si a cada vóxel se le asigna una clase o etiqueta en la salida, ese valor sea coherente con el contexto espacial de ese vóxel.

De esta manera, sus creadores han completado una red de contracción habitual con capas sucesivas de convolución y pooling, con capas de up-sampling que aumentan la resolución en la salida (Krizhevsky et al., 2020). Las capas de contracción en las que se extraen características de alta resolución, se conectan con las capas de muestreo ascendente para la salida. Así, en las capas de expansión se dispone de un mayor número de canales de mapas de características, que le permiten mantener la información de localización (Krizhevsky et al., 2020).

Tal y como se puede observar en el esquema de la red en la figura 3.3.2, la parte de contracción consta de 2 operaciones de convolución consecutivas con un *kernel* de 3x3, seguidas, cada una de ellas, por una ReLU y, posteriormente, una capa de *pooling* de 2x2 con un paso (o *stride*) de factor 2. En cada capa de profundidad se duplica el número de mapas de características, alcanzando cada vez más el detalle (Krizhevsky et al., 2020). De manera contraria, en la ruta de *up-sampling*, cada capa reduce el número de mapas de características a la mitad y se concatenan con los mapas correspondientes de la ruta de contracción. En la última capa se realiza una convolución de 1x1, obteniendo un número de mapas igual al número de clases posibles. En cada uno de esos mapas,

los vóxeles presentan una probabilidad de pertenencia a esa clase. La fusión de todos ellos genera la imagen de segmentación final.

2.3.3 Capas de la CNN

2.3.3.1 Capas de dropout

En el entrenamiento de redes neuronales es típica la aparición de un fenómeno llamado sobreajuste. Cuando una red es entrenada por un número pequeño de casos, puede llegar a tal nivel de detalle, que relacione distintas características de los píxeles de las imágenes de entrada que no existen en la realidad, y que han sido producidas por un sobreentrenamiento de la red, hasta el punto de memorizar patrones que son particulares de un único caso (Hinton et al., 1993). Para solventar este problema, si se cuenta con un conjunto de entrenamiento reducido, se suele utilizar la técnica de *dropout*.

El término hace referencia a la multiplicación entre el mapa de características obtenido en la capa anterior, con un mapa, de sus mismas dimensiones, de 1s y 0s que están distribuidos de manera aleatoria. Así, algunos valores de los mapas de características tendrán un valor de cero. De esta manera, se está eliminando información para que la red no memorice a tanto nivel de detalle y se quede con las características más relevantes y robustas de las estructuras de la imagen. Según el valor de dropout que se asigne, entre 0 y 1, se eliminará tanto porcentaje de datos del mapa de características. En cada capa de dropout, y en cada iteración, la distribución de 1s y 0s será distinta y aleatoria.

En resumen, la adición de estas capas de dropout supone un aprendizaje de características más robusto de la imagen y reducirá el sobreajuste indeseado, dado que se está introduciendo variabilidad, una diferencia entre los mapas de características obtenidos en las capas convolucionales.

2.3.3.2 Capas de activación

Tras las capas de convolución, existen capas de activación que utilizan unas funciones llamadas, concretamente, funciones de activación, que transforman la suma ponderada de salida en la convolución en una salida de valor distinto, que será la entrada de la siguiente capa. El objetivo de estas capas es introducir cambios no lineales en los datos principalmente. Existen diversas funciones de activación, pero las más populares son: la sigmoide, *softmax* y ReLU. En particular, la función ReLU (3) es la usada en la red neuronal de este TFG.

$$Z = b + \sum_i w_i * x_i \quad (2)$$

$$R(z) = \max(0, z) \quad (3)$$

Z representa la suma ponderada de pesos que se obtiene tras la convolución del *kernel* con la imagen al completo. R(z) es la función de activación a la que se somete la suma ponderada. La

función permite el paso de valores positivos, pero no de valores negativos. Aquellos valores que estén por debajo de cero se sustituirán por dicho valor, lo que equivale a tomar el máximo entre el valor de la función z y 0.

La finalidad de las capas de activación es perder esa linealidad que existe entre las entradas y la salida en la suma ponderada. Las funciones de activación no lineales le permiten a la red establecer relaciones no lineales entre las capas, lo que la convierte en un modelo muy expresivo, capaz de aprender relaciones muy complejas entre las entradas y salidas (Hinton et al., 1993).

2.3.3.3 Capas de pooling

Posteriormente a cada capa de convolución, es frecuente aplicar una capa de *pooling*. Una cualidad de las RNC es que registran la posición precisa de las características de la entrada. Este hecho en ocasiones es desfavorable, ya que crea en la red una sensibilidad y dependencia enorme de los desplazamientos espaciales que se pueden producir de una imagen a otra, tales como una leve rotación. La solución para dicha sensibilidad reside en el método de *down-sampling* o muestreo descendente, que resulta en una capa de agrupación (*pooling*) que aporta robustez a la red y además aumenta el campo de visión efectivo de la red.

Estas capas de agrupación funcionan actuando sobre cada mapa de características, creando uno nuevo, de tamaño reducido, pero con la información relevante. Información que se selecciona en base a una operación de agrupación (Brownlee, 2019). Las dimensiones de este filtro de agrupación que recorre la imagen son menores que el tamaño de la imagen de entrada, concretamente, se suele utilizar un *kernel* de $2 \times 2 \times 2$ con un *stride* de 2 vóxeles (Brownlee, 2019). El filtro tomará la información que se le indique. Entre las funciones más utilizadas para la operación de *pooling* encontramos *Average Pooling*, que calcula el valor medio de cada ventana que el *kernel* toma del mapa de características y *Max Pooling* que se queda con el valor máximo de cada ventana de muestra, y pasa este valor como resumen de características sobre ese volumen.

La ventaja principal de usar una capa de pooling es que se toman versiones reducidas de las características de la imagen de entrada, que ignoran los posibles pequeños cambios espaciales de la imagen, salvaguardando las características resumidas por parches y sus localizaciones correspondientes, invariantes a mínimas traslaciones locales.

3. OBJETIVOS

El objetivo principal de este TFG es la creación y validación de una Red Neuronal Convolutiva (RNC), entrenada con imágenes etiquetadas de casos de una base de datos pública, que segmente las 3 estructuras comentadas: tronco encefálico (*brainstem*), núcleo talámico, hipocampo y amígdala. La salida o 'output' de la red será la segmentación de las estructuras para cada uno de los sujetos de test que se le proporcione. El aprendizaje adquirido por la red para la discriminación de las estructuras entre todos los vóxeles de la imagen, llevará a conseguir una herramienta capaz de segmentar las ROIs de cualquier imagen de RM tipo T1/T2 de cualquier paciente de forma precisa y eficiente.

El uso de una red neuronal se justifica por su capacidad de extraer características de las imágenes para estudiar sus estructuras y patrones.

Para lograr alcanzar el objetivo final, se han ido estableciendo objetivos menores a lo largo del proceso. La superación de cada uno de ellos, ha llevado a alcanzar la meta.

- Análisis y estudio de las funciones cognitivas y motoras realizadas por las estructuras de interés: *brainstem*, núcleo talámico, hipocampo y amígdala.
- Estudio de la adquisición de imágenes de resonancia magnética en las modalidades T1 y T2.
- Aprendizaje del funcionamiento del software ‘Freesurfer’, para la segmentación de las ROIs.
- Aprendizaje del lenguaje de programación Python, sobre el que se han desarrollado los *scripts* para la automatización de la etapa de registro de las imágenes al espacio MNI, y los *scripts* necesarios para el funcionamiento de la red neuronal.
- Actualización sobre el estado del arte en registro y segmentación. Aplicación del software ANTs para registro de imágenes.
- Aprendizaje sobre la aplicación Anaconda, que genera el entorno de trabajo para Python y las librerías de Tensor Flow y Keras.
- Estudio del funcionamiento de las CNNs, en concreto de la que se ha hecho uso, el modelo U-Net.
- Entrenamiento y evaluación de la red neuronal, a través de las imágenes etiquetadas y ya registradas obtenidas con Freesurfer. Optimización de los errores obtenidos mediante la actualización de parámetros en la red, hasta obtener el mejor resultado de segmentación.
- Finalmente, creación de un método robusto de segmentación de imágenes de resonancia magnética basado en *Deep Learning* que mejora las posibilidades de las que ya se disponen actualmente.

4. MATERIALES Y MÉTODOS

4.1 Database

La base de datos que se ha utilizado en el trabajo, forma parte del Proyecto Conectoma Humano (*Human Connectome Project*), que es una iniciativa para construir un mapa de la red que supone el cerebro, para aportar luz al conocimiento de la conectividad anatómica y funcional del cerebro humano sano, así como obtener un conjunto de información que sirva para la investigación de las enfermedades mentales como el Alzheimer, demencia o esquizofrenia. Dentro de este proyecto, hay numerosas bases de datos de imágenes de sujetos que son públicas, y, por tanto, descargables. En este TFG se ha tomado el conjunto de imágenes de tipo T1 y T2 de RM de 76 pacientes, dentro de la base de datos de 1200 sujetos sanos que ofrece el Proyecto Conectoma Humano.

Las imágenes escogidas son imágenes de calidad, adquiridas en equipos de RM de 3 Tesla (T), durante el periodo 2012-2015 (*Human Connectome Project*, 2017). Cabe resaltar que, a mayor magnitud de intensidad de campo generado por el imán, mejor calidad de imagen, por lo que dichas imágenes son, en comparación, mejores, que las obtenidas con imanes de 1,5T.

La base de datos S1200 (*1200 Subjects Data Release*) cuenta, además de con los escaneos de RM estructurales 3T, con datos de exploraciones de RM 7T multimodal de 184 sujetos y datos de MEG (Magnetoencefalografía) de estado de reposo (rMEG) y/o MEG de tarea (tMEG) de 95 sujetos (*Human Connectome Project, 2017*).

Estas imágenes T1 Y T2, se registraron al espacio MNI, como paso inicial y primordial del trabajo, para su posterior uso.

El espacio MNI es un sistema de coordenadas establecido por el Instituto Neurológico de Montreal, basado en un conjunto de imágenes de RM de un cerebro sano. El espacio en el que se encuentran las imágenes originalmente es el espacio nativo, al que no se le ha aplicado ninguna transformación. Es por ello que, las dimensiones y el tamaño del vóxel, entre otros, son diferentes en este espacio. Así, registrando las imágenes al espacio MNI, se hacen comparables y posible trabajar con ellas en la parte posterior de aprendizaje profundo.

4.2 *Software* Freesurfer

4.2.1 Manejo del *software*

Previamente a la fase de *Deep Learning*, se tiene todo el proceso previo de preparación de las imágenes para poder introducirlas en la red. Partiendo de la base de datos de la que se dispone, se hace uso de Freesurfer, un paquete de *software* de libre acceso, desarrollado por los investigadores del centro de Imagen Biomédica Athinoula A. Martinos, que se encarga de la segmentación de las ROI's definidas en las imágenes de la base de datos de pacientes de la que se dispone. El programa funciona a través de comandos de consola tanto en sistemas operativos de Linux como IOS (Mac). Al disponer de un ordenador con sistema operativo Windows, se tuvo que realizar una partición para dar posibilidad al ordenador de tener un arranque con Linux. Freesurfer consume mucho tiempo de procesador y recursos de memoria en el disco.

Toda la información acerca del funcionamiento del *software* se encuentra en la página web del mismo. El comando principal de uso que ejecuta es 'recon-all'. Este comando realiza un flujo basado en superficie para la segmentación cortical, a partir de una imagen de entrada de resonancia magnética potenciada en T1, con un buen contraste entre materia gris y blanca.

Una reconstrucción cortical es una representación computarizada de la capa cortical cerebral, que resulta de especial utilidad para el análisis cuantitativo de la estructura del cerebro, como es estudiar sus patrones de plegamiento, morfometría o su espesor. Dado que estos modelos se utilizan como base para estudios posteriores, deben ser realmente precisos en cuanto a geometría y topología del cerebro humano.

Recon-all primero extrae el cráneo de la imagen anatómica para crear un archivo 'brainmask.mgz'. Posteriormente, delinea la sustancia blanca y genera dos archivos: lh.white y rh.white, correspondiente a los dos hemisferios. A partir de la información de la materia blanca, calcula la superficie pial, que es una envoltura externa de la sustancia gris y lo guarda en otros dos archivos: lh.pial y rh.pial. Con esta información, obtiene el espesor de la sustancia gris, que resulta la distancia entre la sustancia blanca y la superficie pial. Todos estos archivos tienen extensión mgz, que es característica de archivos comprimidos en Freesurfer (*FreeSurfer Tutorial*). El programa permite extender estas superficies a volúmenes tridimensionales, que se almacenan en un directorio 'mri', y que se pueden visualizar gracias al comando 'freeview' seguido del archivo que se quiere observar.

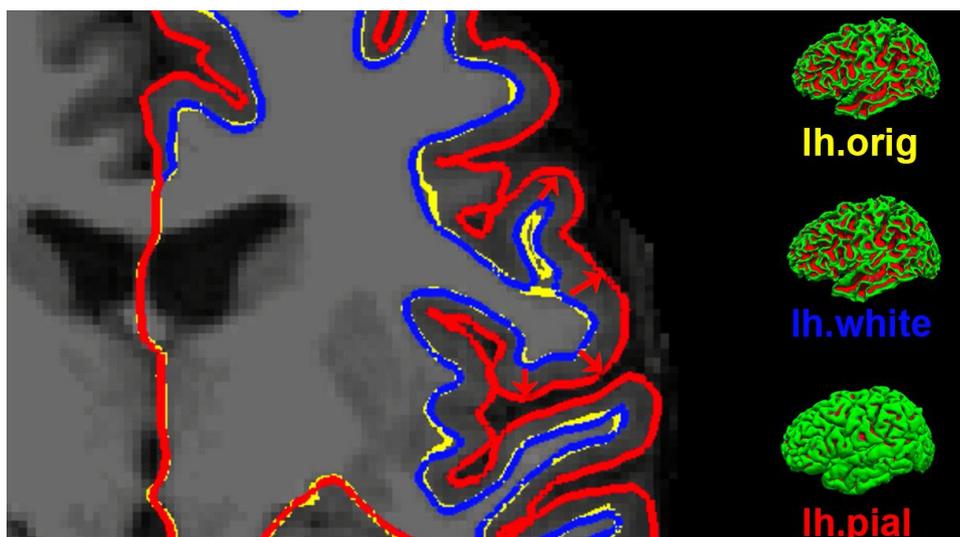


Figura 11: Segmentaciones de la materia blanca, materia gris y superficie pial del cerebro a través del comando ‘recon-all’ (*FreeSurfer Tutorial*).

La línea de ejecución de la figura 4.1.1.1, crea un directorio con el nombre indicado en el apartado ‘subjectname’ para el almacenado de todos los archivos que la función recon-all va a generar del volumen T1 que se le introduce como *input* en el campo ‘invol1’. La inclusión de un volumen invol2 es opcional. El término ‘invol’ hace referencia a *input volume*. Recon-all únicamente funciona con volúmenes de tipo T1. En este caso, se decidió darle el nombre identificativo de cada sujeto de la base de datos al directorio o carpeta donde se iban a almacenar los volúmenes correspondientes.

```
recon-all -subject subjectname -i invol1 <-i invol2> -all
```

Figura 12: Ejecución del comando ‘recon-all’.

Freesurfer, además, puede generar otro flujo basado en volumen, para la segmentación de las estructuras subcorticales. Para ello, Freesurfer utiliza un atlas probabilístico, que se obtiene de un conjunto de entrenamiento que consta de múltiples imágenes de atlas etiquetadas a mano. Tras una alineación volumétrica no lineal de la imagen a estudiar y el atlas, las etiquetas de las ROIs se transfieren directamente a cada vóxel de la imagen de estudio, maximizando la probabilidad de la imagen de entrada dados los datos del mapa, que contienen probabilidades a priori. Como se puede ver, Freesurfer usa una técnica de segmentación basada en *single-atlas*. Cabe destacar que el coste computacional de este comando es de unas 6 o 7 horas.

Tras la segmentación general del cerebro, la puesta en marcha de las segmentaciones de las ROIs se produce mediante la utilización de comandos específicos de segmentación. En primer lugar, el tronco encefálico (Iglesias):

```
segmentBS.sh bert [SUBJECTS_DIR]
```

Figura 13: Ejecución del comando de segmentación del tronco encefálico.

El software realizará la segmentación del tronco encefálico del sujeto que se le indique en el apartado ‘bert’. El directorio del sujeto es opcional indicarlo, por ello lo encontramos entre corchetes. Automáticamente, si se deja este campo en blanco, Freesurfer accederá a buscar ese sujeto en lo que se haya definido como SUBJECTS_DIR en la instalación de Freesurfer. La ayuda para la instalación del software indica una serie de pasos a seguir que estructuran las distintas carpetas que Freesurfer va a generar.

```

$ cd $HOME
$ pwd
/home/tester
$ sudo tar -C /usr/local -zxvpf freesurfer-linux-centos7_x86_64-7.0.0.tar.gz
x freesurfer/
x freesurfer/WMParcStatsLUT.txt
x freesurfer/sessions/
x freesurfer/sessions/README
...
..
<rest of output deleted>

$ export FREESURFER_HOME=/usr/local/freesurfer
$ export SUBJECTS_DIR=$FREESURFER_HOME/subjects
$ source $FREESURFER_HOME/SetUpFreeSurfer.sh
----- freesurfer-linux-centos7_x86_64-7.0.0-20200427-5b17a0d -----
Setting up environment for FreeSurfer/FS-FAST (and FSL)
FREESURFER_HOME   /usr/local/freesurfer
FSFAST_HOME       /usr/local/freesurfer/fsfast
FSF_OUTPUT_FORMAT nii.gz
SUBJECTS_DIR      /usr/local/freesurfer/subjects
MNI_DIR           /usr/local/freesurfer/mni

$ which freeview
/usr/local/freesurfer/bin/freeview

```

Figura 14: Tutorial de instalación de Freesurfer para Linux con opciones de sudo (administrador) (*Linux Install & Setup*).

La distribución general de directorios que hace Freesurfer es generar una carpeta llamada SUBJECTS_DIR donde guarda las imágenes de los sujetos que se van a estudiar, suponiendo cada uno de ellos una carpeta distinta. Dentro de las carpetas de cada uno de ellos se encuentra otra carpeta con el nombre ‘mri’ donde se almacenan todos los archivos que se generan relacionados con el mismo paciente. Ahí es donde se encontrarán los ficheros con los volúmenes que se irán segmentando.

4.2.2 Segmentación de las estructuras

La segmentación del tronco encefálico va a generar 3 archivos distintos (Iglesias):

- brainstemSsLabels.v1x.mgz (el volumen con la segmentación a 0.5 mm de resolución).
- brainstemSsLabels.v1x.FSvoxelSpace.mgz (segmentación discreta en el tamaño del vóxel, 1mm).
- brainstemSsVolumes.v1x.txt (un archivo de texto con los volúmenes de la estructura del tronco encefálico en mm).

El archivo de interés es el que presenta el volumen con las etiquetas del tronco encefálico con resolución a 0.5mm, dado que, presenta mayor resolución, por una disminución en el tamaño del vóxel. Este fichero presenta 4 etiquetas en las que separa las regiones del núcleo talámico:

Etiquetas del tronco encefálico	
173	Mesencéfalo
174	Puente troncoencefálico (Pons)
175	Médula
178	Pedúnculo cerebeloso superior (SCP)

Tabla 1: Etiquetas del tronco encefálico.



Figura 15: Imagen T1 anatómica segmentada con las etiquetas del tronco encefálico.

Los números asignados a las etiquetas no siguen una continuidad, por lo que se encuentran grandes saltos entre los valores de una etiqueta a la siguiente. Esto no es relevante en el proceso, es meramente el nombre de las etiquetas.

Es posible visualizar simultáneamente la segmentación resultante y la imagen anatómica T1 de entrada. Esta función se puede conseguir con la aplicación del comando 'freeview'; siendo el fichero nu.mgz la versión corregida de orig.mgz en cuanto a heterogeneidades de intensidad.

```
freeview -v un.mgz -v brainstemSsLabels.v1x.mgz:colormap=lut
```

Figura 16: Ejecución del comando de visualización de la segmentación del tronco encefálico.

La segmentación del hipocampo y la amígdala, por su parte, permite añadir una imagen adicional, cuya información se va a utilizar para complementar y mejorar la segmentación de las estructuras. En este caso, como se disponía de imágenes de los sujetos potenciadas en T2, se decidió añadir este escáner adicional a la imagen que ya se tenía de partida, T1. Los archivos generados por Freesurfer tras la ejecución del comando de segmentación son:

- [lr]h.hippoSfVolumes-T1-T1+T2.v21.txt (dos archivos de texto, uno para la estructura del hipocampo y la amígdala en el hemisferio derecho y otro para el hemisferio izquierdo, con el valor estimado de los volúmenes de las subestructuras del hipocampo y del hipocampo al completo) [38].
- [lr]h.hippoAmygLabels-T1-T1+T2.v21.CA.mgz (dos volúmenes, de nuevo separando el hemisferio izquierdo del derecho, con las segmentaciones correspondientes a una resolución de 0.333 mm) [38].
- [lr]h.hippoAmygLabels-T1.v21.CA.FSvoxelSpace.mgz (volúmenes etiquetados con una resolución de 1 mm, siendo ésta la resolución del vóxel en el espacio nativo de Freesurfer) (Iglesias).

Los archivos que resultan útiles son los que contienen los volúmenes del hipocampo y la amígdala segmentados en los hemisferios derecho e izquierdo a la resolución de 0.333 mm. Distinguiendo las áreas en las que se divide el hipocampo y la amígdala se encuentran las siguientes etiquetas.

Etiquetas del hipocampo y la amígdala

203	Parasubículo
204	Presubículo
205	Subículo
206	Cornu Ammonis región 1 (CA1)
208	Cornu Ammonis región 2+3 (CA3)
209	Cornu Ammonis región 4 (CA4)
211	Área de transición del hipocampo a la amígdala (HATA)
212	Fimbria
215	Fisura hipocampal
226	Cola del hipocampo
7001	Núcleo lateral
7003	Núcleo basal
7005	Núcleo central
7006	Núcleo medio

7007	Núcleo cortical
7008	Núcleo basal accesorio
7009	Cortico-amigdalino transitio
7010	Área anterior de la amígdala
7015	Núcleo paralaminar

Tabla 2: Etiquetas del hipocampo y la amígdala.

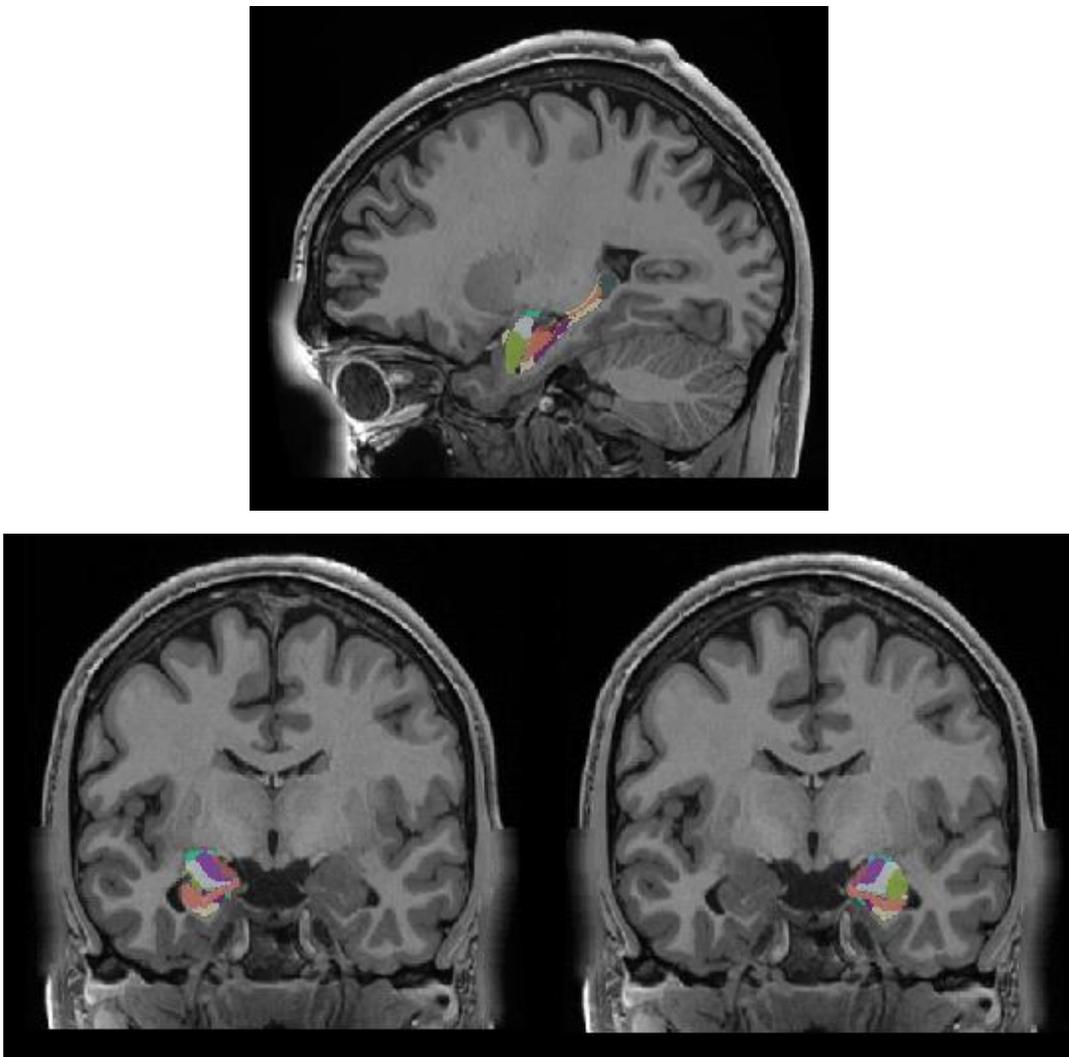


Figura 17: Imágenes sagital y coronal T1 anatómicas segmentadas con las etiquetas del hipocampo y la amígdala.

Tal y cómo se puede observar, en el caso de la segmentación del hipocampo y la amígdala, Freesurfer la lleva a cabo por separado, distinguiendo entre el hemisferio derecho e izquierdo. A pesar de ello, la asignación de los valores de etiquetas es el mismo en los dos hemisferios. Por

tanto, el valor de etiqueta que se encuentra para una misma estructura en el lado derecho e izquierdo es idéntico.

La visualización de la imagen anatómica junto a la segmentación se puede visualizar de la misma manera que en el caso anterior con el comando 'freeview'. La ejecución de la segmentación para dichas estructuras se lleva a cabo a través del comando siguiente.

```
recon-all -s <subject_name> -hippocampal-subfields-T1T2 <file name  
of additional scan> <analysisID>
```

Figura 18: Ejecución del comando de segmentación del hipocampo y la amígdala con escáner extra.

Es estrictamente necesario haber llevado a cabo la ejecución del comando *recon-all* previamente a realizar cualquier segmentación.

Finalmente, para la segmentación del Núcleo Talámico se dispone también de la opción de complementar esta función con una imagen adicional. Sin embargo, en este caso, se probó, siguiendo la misma metodología que en la estructura anterior, aportar la imagen potenciada en T2 del sujeto, pero el módulo de registro que incluye el comando de Freesurfer para la segmentación, no conseguía alinear correctamente las imágenes, por lo que la segmentación salía errónea. De esta manera, se decidió segmentar el Núcleo Talámico únicamente con la información aportada con la imagen T1, que resultó ser suficiente.

Los archivos que se obtienen de la segmentación son(Iglesias) :

- ThalamicNuclei.v12.T1.volumes.txt.
- ThalamicNuclei.v12.T1.mgz.
- ThalamicNuclei.v12.T1.FSvoxelSpace.mgz.

Análogamente, se selecciona el archivo que contiene el volumen de la estructura etiquetada, a una resolución de 0.5 mm. Las resoluciones de las segmentaciones varían dependiendo de la estructura. Para los volúmenes del tronco encefálico y del núcleo talámico se ha obtenido una resolución del vóxel de 0.5 mm, contrariamente a la resolución de 0.33 mm del hipocampo y la amígdala. Esto no supone un problema, dado que, posteriormente, se ha realizado un registro de todas las estructuras segmentadas con la imagen T1 anatómica del paciente, a la resolución de la misma.

Las etiquetas correspondientes al área del núcleo talámico que nos proporciona la segmentación de Freesurfer son las siguientes.

Etiquetas del Núcleo Talámico

8103	Núcleo anteroventral izquierdo
8104	Núcleo centromedial izquierdo
8105	Núcleo centrolateral izquierdo
8106	Núcleo centromediano izquierdo
8108	

	Núcleo laterodorsal izquierdo
8109	Núcleo geniculado lateral izquierdo
8110	Núcleo lateral posterior izquierdo
8111	Núcleo supragenículado izquierdo
8112	Núcleo parvocelular lateral mediodorsal izquierdo
8113	Núcleo magnocelular medial mediodorsal izquierdo
8115	Núcleo geniculado medial izquierdo
8116	Núcleo ventral medial izquierdo
8118	Núcleo parafascicular izquierdo
8119	Núcleo paratenial izquierdo
8120	Núcleo pulvinar anterior izquierdo
8121	Núcleo pulvinar inferior izquierdo
8122	Núcleo pulvinar lateral izquierdo
8123	Núcleo pulvinar medial izquierdo
8126	Núcleo ventral anterior izquierdo
8127	Núcleo magnocelular ventral anterior izquierdo
8128	Núcleo lateral ventral anterior izquierdo
8129	Núcleo lateral ventral posterior izquierdo
8130	Núcleo ventromedial izquierdo
8133	Núcleo posterolateral ventral izquierdo
8203	Núcleo anteroventral derecho
8204	Núcleo centromedial derecho
8205	Núcleo centrolateral derecho
8206	Núcleo centromediano derecho
8208	Núcleo laterodorsal derecho

8209	Núcleo geniculado lateral derecho
8210	Núcleo lateral posterior derecho
8211	Núcleo suprageniculado derecho
8212	Núcleo parvocelular lateral mediodorsal derecho
8213	Núcleo magnocelular medial mediodorsal derecho
8215	Núcleo geniculado medial derecho
8216	Núcleo ventral medial derecho
8218	Núcleo parafascicular derecho
8219	Núcleo paratenial derecho
8220	Núcleo pulvinar anterior derecho
8221	Núcleo pulvinar inferior derecho
8222	Núcleo pulvinar lateral derecho
8223	Núcleo pulvinar medial derecho
8226	Núcleo ventral anterior derecho
8227	Núcleo magnocelular ventral anterior derecho
8228	Núcleo lateral ventral anterior derecho
8229	Núcleo lateral ventral posterior derecho
8230	Núcleo ventromedial derecho
8233	Núcleo posterolateral ventral derecho

Tabla 3: Etiquetas del núcleo talámico.

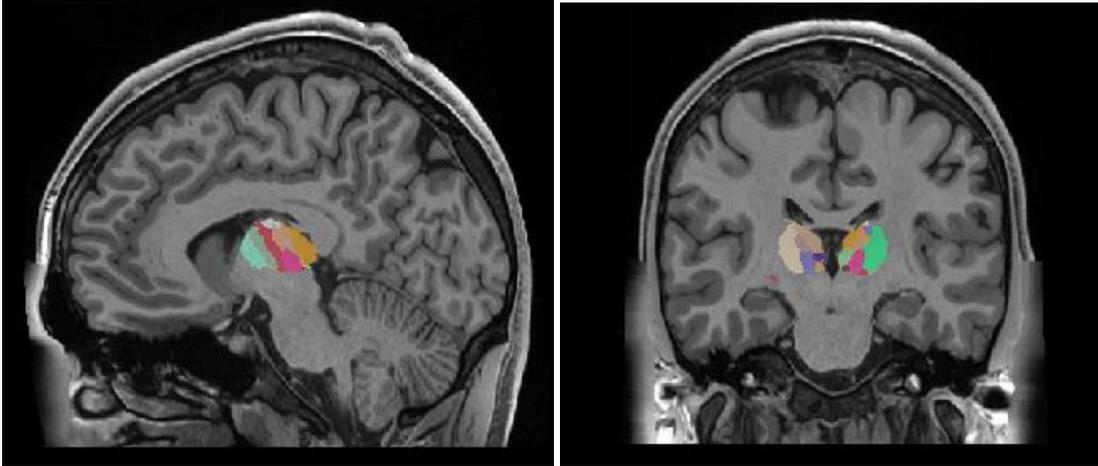


Figura 19: Imagen sagital y coronal de la segmentación del núcleo talámico con las distintas etiquetas.

A continuación, se visualiza el comando para la ejecución de dicha segmentación.

```
segmentThalamicNuclei.sh bert [SUBJECTS_DIR]
```

Figura 20. Ejecución del comando de segmentación del núcleo talámico.

Finalmente, la primera etapa de este TFG concluye con la obtención de todos los volúmenes de segmentación que ha generado la herramienta Freesurfer. A partir de ellos, se va a trabajar para mejorar dichas segmentaciones y construir una red neuronal que lo haga automáticamente y con una eficacia y eficiencia mayor. Cabe destacar que, los tiempos de espera que hay que tomar para la conclusión de cada uno de los procesos de segmentación que Freesurfer genera son considerables, especialmente el comando *recon-all*.

4.3 Postprocesado de las segmentaciones

Para, posteriormente, poder trabajar con las imágenes de segmentación proporcionadas por Freesurfer, fue necesaria la conversión de los archivos de extensión *mgz* a la extensión *nii.gz*, ya que la herramienta usada para el registro de imágenes, ANTs, admite imágenes de tipo 2D y 3D en formato *nifti* y no *mgz*.

El registro de las estructuras segmentadas para cada paciente de la base de datos es necesario, pues tal y como se ha visto con anterioridad, la resolución de las estructuras es dispar y se busca tener todas a la misma resolución que la imagen original anatómica T1. Por ello, se ha procedido a registrar los distintos volúmenes de las ROI's obtenidos con Freesurfer a la imagen T1. Se ha hecho uso del software ANTs de registro a través del lenguaje de programación Python.

Se define una función *ToMNI*, con 2 entradas, la imagen a registrar y la plantilla. Esta función aplica la línea de comando del software ANTs, en la que se indica la imagen a registrar, la imagen de referencia, el nombre del fichero de salida, y el tipo de registro a realizar. Dado que se trata de

un registro de etiquetas, se aplicó la técnica *NearestNeighbor*, para evitar la introducción de artefactos de interpolación.

Para cada sujeto, se toman los 4 archivos de interés obtenidos con Freesurfer: `brainstemBsLabels.v12.nii.gz`, `lh.hippoAmygLabels-T1-T1+T2.v21.CA.nii.gz`, `rh.hippoAmygLabels-T1-T1+T2.v21.CA.nii.gz` y `ThalamicNuclei_v12_T1.nii.gz`, y se registran consecutivamente con su imagen de referencia correspondiente, con la aplicación de la función `ToMNI` para cada una de las estructuras. Dado que el registro se ha realizado por separado, cada estructura con la plantilla, interesa juntar estas imágenes en una, en un proceso de *merging*. Se obtienen 5 archivos, los correspondientes a las estructuras segmentadas registradas con la imagen T1, y un archivo de etiquetas global que incluye los 4 anteriores.

4.4 Entrenamiento de la Red Neuronal Convolutiva (CNN)

La arquitectura seguida para la segmentación de las imágenes de RM, es la creación de 3 redes neuronales convoluciones, una para cada ROI. La metodología es que estas tres redes van a seguir un modelo de red convolutiva definido en un *script* llamado 'modelos', y van a tomar los datos de entrada o *inputs* de otro *script* denominado 'Data'. También se dispone de los *scripts* de *training* y de *test* de cada una de las estructuras.

4.4.1 Data Augmentation

La técnica de *Data augmentation*, tal y cómo indica su nombre, tiene como finalidad incrementar la cantidad de datos de partida, especialmente en conjuntos de datos limitados, para evitar ese sesgo en los resultados, al trabajar, la red, con muy pocos datos de entrada, que le ofrecen poca variabilidad en su aprendizaje. El sesgo desaparece conforme aumenta la cantidad de información y la diversidad de casos. Es por ello, que se decidió aplicar la técnica de *flip* al conjunto de imágenes que se había destinado al entrenamiento. Con ello, se consigue la inversión espacial en el eje x de las imágenes T1, T2, y las imágenes etiquetadas con la segmentación correspondiente de las estructuras, aprovechando al pseudosimetría del cerebro.

Se debe tener en consideración que, las etiquetas en el tronco encefálico tienen el mismo valor para el lado derecho que para el izquierdo, al igual que ocurre en el caso del hipocampo y la amígdala. Por el contrario, en el caso del Núcleo Talámico, las etiquetas del lado derecho de la estructura, no se corresponden con las del lado izquierdo, presentando valores distintos.

Al aplicar el *flip*, las imágenes T1 y T2 rotarán, generando nuevos casos de entrenamiento, introduciendo una pequeña variabilidad correspondiente a las mínimas diferencias que se pueden encontrar entre los lados derecho e izquierdo de las estructuras. En este punto, el fichero de etiquetas que, también ha sido modificado, debe ser corregido, para restaurar los valores de las *labels* correspondientes al lado derecho e izquierdo respectivamente, sólo en el caso del núcleo talámico, que es el que presenta distintas etiquetas dependiendo del hemisferio. Así, los índices de la matriz tridimensional rotada, que tienen el valor de la etiqueta de uno de los lados, bien derecho o izquierdo, se reasignan a los valores de etiqueta de la matriz original. Para las otras dos estructuras subcorticales, la técnica de *flip* no significa una modificación en las etiquetas, ya que, sus valores son los mismos en ambos lados.

A través del uso del software Matlab, se aplicó esta técnica de *Data augmentation* para generar una especie de “nuevos casos” para la red, que aumentarían el conjunto de entrenamiento. De esta

manera, se consigue aumentar el conjunto de datos de entrenamiento, de 55 casos (del total de 76) a 110 casos (105 de entrenamiento y 5 de validación).

4.4.2 Data preparation

El script de generación de datos consta de 6 funciones, dos dedicadas a cada una de las estructuras. Una primera función 'read_val_', lista las imágenes en el directorio indicado, tanto las de tipo T1 como tipo T2 de cada sujeto, que se encuentran en formato nifti, y lee los valores de dichas matrices tridimensionales que describen las imágenes. Para evitar trabajar con toda la imagen al completo, que resulta de grandes dimensiones, se ha realizado un *crop* con las coordenadas correspondientes en los ejes x, y, z que contienen a la estructura de interés. Estas coordenadas varían en la función, dependiendo de cuál de las tres estructuras se esté contemplando, ya que se encuentran espacialmente distribuidas en distintas zonas. Este recorte en la imagen se ha podido realizar gracias a la extracción de las coordenadas de interés con el *software* Matlab. El archivo cargado para la extracción de coordenadas es el que contiene la estructura del tronco encefálico etiquetada una vez registrada con la imagen. Dado que estos archivos solo contienen la estructura segmentada en las coordenadas espaciales correspondientes sobre un fondo negro, se consideró una buena opción calcular los límites de las estructuras en estos ficheros, para después aplicarlos a las imágenes anatómicas T1 y T2 y al propio fichero de etiquetas global, que contiene las *labels* de las 3 ROIs.

A continuación, se normalizan los datos con el cálculo de la media y la desviación típica para cada una de las matrices y aplicando una transformación Z-scoring, separando T1 y T2. Es necesaria la normalización previamente a la preparación del *input*. La entrada de la red constituye una concatenación de las matrices T1 y T2 normalizadas y transformadas a formato tensor. Un tensor es una estructura de 5 dimensiones, donde la primera y la última dimensión resultan el número de volúmenes de imágenes/número de canales y las 3 dimensiones centrales son los tamaños de las imágenes en los tres ejes. Las dos entradas, imagen T1 y T2, en principio, independientes, se concatenan para que resulten un único input. Así, el tensor de entrada a la red posee un valor de 1 en la primera y 2 en la última dimensión y el tamaño de la imagen en los 3 ejes, en las dimensiones centrales. Es necesaria la conversión dimensional al formato tensor, dado que la API de redes con la que se trabaja, Keras, en la que, posteriormente, se profundizará, requiere del uso de tensores.

Tras la preparación del *input* de la red se requiere de la definición del *output* o salida. En este caso, se utilizó un tensor de 1 x ancho x alto x profundo x número de clases que representaban los mapas de probabilidad de cada etiqueta. Esta es la solución al problema de segmentación que se le plantea en la entrada a la red, y que, con la fase de *training* intentará explicar a través de la actualización de los pesos hasta alcanzar el modelo óptimo.

Esta función proporciona como salida los valores de las variables que se requerirán posteriormente para la aplicación del modelo de red.

La segunda función a destacar es 'generator_'. Esta función genera los datos para la red. Aplica *Data augmentation online* basado en la utilización de una serie de filtros en la imagen con un tamaño de kernel determinado y caracterizado por una operación de convolución que puede producir *sharpening* o afinado de la imagen, o *blurring* o emborronamiento de la imagen, según convenga. Este procedimiento es totalmente aleatorio, de manera que, imágenes de entrada al azar serán modificadas con estas técnicas de imagen, para la generación de datos nuevos, que no dejan de ser pequeñas alteraciones en las imágenes para generar mayor variabilidad.

Si aplicamos esta segunda función, las salidas que nos proporciona son un *input* para la entrada de la red, con las modificaciones de *data augmentation* correspondientes, y una salida, tal y como se ha indicado en la anterior función.

4.4.3 Training

La fase de entrenamiento o aprendizaje de una red puede ser de tipo supervisado o no supervisado. En este caso, el entrenamiento es supervisado por entrenar a la red por un conjunto de patrones de entrada y de salida. La finalidad del algoritmo de aprendizaje es ajustar los pesos de la red de tal forma que la salida proporcionada sea lo más cercana posible a la verdadera salida, ofrecida a la red como un dato. La red neuronal va a tratar de encontrar un modelo que lleve a la solución proporcionada. Este aprendizaje se llama supervisado pues se conoce el patrón de salida el cual hace el papel de supervisor de la red.

En la fase de entrenamiento se importa el fichero de Data. Python permite la integración de distintas librerías, para poder hacer uso de ellas, pero es necesaria la importación de las mismas. Al igual ocurre con otros ficheros con extensión py. La integración del *script* 'Data' permite aplicar la función 'read_val', que, tal y como se comentó en el apartado 4.4.2.1, proporciona las variables requeridas para la aplicación del modelo de red. En la fase de entrenamiento se debe distinguir entre los datos puramente para el aprendizaje y los datos de validación. El conjunto de validación es útil para evitar el *overfitting* o sobreentrenamiento de la red. Si se evalúa la habilidad de un modelo con datos pertenecientes al conjunto de datos de entrenamiento, se obtendría un resultado sesgado. El sesgo muestral conduce a errores y preferencias en los resultados. Por ello, el conjunto de evaluación se mantiene fuera del de entrenamiento. Gracias a él se puede obtener una estimación no sesgada de la habilidad del modelo (Brownlee, 2017) mientras éste está entrenando, de forma que, el modelo que está generando la red se utiliza para predecir el resultado del conjunto de datos de validación.

La tasa de error obtenida se utiliza para optimizar el modelo. Se debe diferenciar específicamente el conjunto de validación del de 'test', que permite “guardar” de forma automática la red con mejor desempeño. La función del conjunto de validación es optimizar la fase de aprendizaje de la red evitando el *overfitting*.

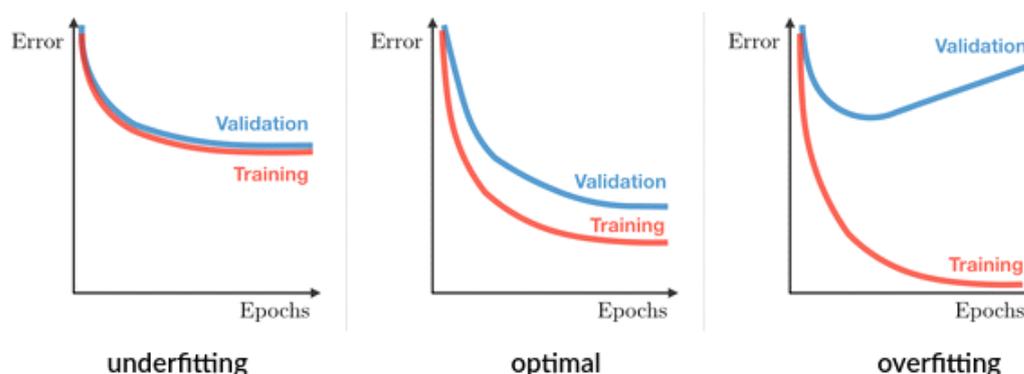


Figura 21: Gráficos de la evolución del error del modelo en relación a las *epochs*. Describen fenómenos de *overfitting*, *underfitting* y aprendizaje óptimo (*GitHub-Predicciones Con Redes Neuronales*).

Es posible entrenar la red en base a distintos tipos de *loss* o error, y optimizadores; siendo posible, con una elección acertada de los mismos, mejorar los resultados posteriores en el testaje de la red. Uno de los optimizadores más utilizados y, que ha sido el seleccionado en este TFG, es el optimizador Adam. Primeramente, un optimizador es el responsable de generar pesos que sean cada vez mejores y más explicativos de la imagen. El funcionamiento básico que lleva a cabo es el cálculo del gradiente de la función de coste por cada peso de la red. Dado que el gradiente supone la máxima variación, y lo que se busca es lo contrario, la minimización del error, se toma la dirección negativa del gradiente (Velasco, 2020).

Por otro lado, en cuanto a las funciones de error, en este TFG se han contemplado 3 tipos de errores diferentes:

- *Dice loss*. Este tipo de error está especialmente diseñado para la segmentación de objetos basada en el coeficiente de similitud Dice, y es el error más usado en labores de segmentación de imagen médica [43]. La similitud Dice realiza un cociente en el que el numerador es la suma de los píxeles que coinciden entre la imagen 1 y la imagen 2, es decir, la imagen de entrenamiento y la de validación, lo que supone la intersección entre ellas, multiplicada por dos; y el denominador es la suma total de los píxeles de ambas imágenes. El índice Dice tiene un valor entre 0 y 1. Por tanto, para el cálculo del *loss*, se realiza la operación 1-Dice. Se usó como *loss* el *dice* promedio de cada clase (*mdice loss*)

$$d = 1 - \frac{2|X \cap Y|}{|X| + |Y|} \quad (4)$$

- *Jaccard loss*. Función de error basada en el índice Jaccard. Calcula el cociente entre la intersección de la imagen de entrenamiento y la de validación (numerador), y la unión de ambas (denominador). El error, por tanto, se obtiene con la operación 1-Jaccard. De nuevo, su valor está en el rango entre 0 y 1. En el TFG se usó la versión generalizada de este *loss*, que da más peso a aquellas clases con volúmenes más pequeños.

$$\mathcal{J}(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (5)$$

- *Smooth mdice loss*. Se calcula de la misma manera que la función de error basada en el coeficiente Dice. La única diferencia que se encuentra es que, al añadir la suavidad, se evita que el resultado del cociente tienda a cero, cuando los píxeles de la imagen de entrenamiento y la de validación no coinciden. En la práctica, se suma 1 al valor del numerador en el cociente de Dice.

Los conjuntos de entrenamiento y validación se separaron en 105 y 5 casos respectivamente, al disponerse de un conjunto total de 110 casos en el *database*. Se entrenó cada una de las redes correspondientes a las distintas ROIs de manera separada, con variación en el tipo de función de error escogida para el entrenamiento y los valores de *drop* que se introducen como variable de entrada en la aplicación del modelo de red. Estos dos factores han supuesto la principal diferencia en los resultados obtenidos para cada una de las combinaciones.

Para el proceso de entrenamiento se han definido un total de 2000 ciclos o *epochs* como máximo, que, en el caso de que la red las alcanzara, por haber extendido el tiempo de aprendizaje a un gran número de horas, el entrenamiento cesaría. Por cada ciclo realizado, todas las imágenes de entrenamiento van a pasar por la red, para que ésta aprenda (Wikipedia, 2019). Dentro de cada ciclo existen las iteraciones o *steps* por *epoch*, que suponen ejecuciones internas. Para las redes de estudio se han definido 20 iteraciones. La finalidad de suplementar los ciclos con iteraciones es que se actualizan más veces los parámetros variables de la red y se afina más en la salida.

Al ejecutar el primer ciclo, los pesos generados se guardarán como mejor representación del modelo. Tras cada ciclo que tiene lugar posteriormente, se realiza una comprobación sobre si ha habido, o no, una mejora en el valor de *loss* del modelo con los pesos actualizados. Si ha sido así, el modelo generado en última instancia sustituirá al anterior, con los pesos óptimos. Se puede considerar que el entrenamiento de una red ha finalizado cuando no se produce actualización del modelo al cabo de unas horas, ya que éste se ha estabilizado y no es probable que la red mejore significativamente su aprendizaje actual. En el caso de las tres redes de las ROIs, este tiempo de entrenamiento se solía traducir en unas 400 o 500 *epochs*.

En esta fase de entrenamiento, se ha tratado de manera distinta a las 3 estructuras de interés. Por un lado, el tronco encefálico ha sido entrenado en su totalidad por la red, pero, por otro lado, las estructuras del hipocampo y amígdala y el núcleo talámico han sido entrenadas únicamente en su parte derecha. Se consideró que, entrenar al completo estas estructuras, cuando presentan simetría en los dos hemisferios, izquierdo y derecho, conllevaba un coste computacional mayor. Se consideró como opción, entrenar únicamente la estructura del hemisferio derecho del HA y el NT y aplicar a las imágenes T1 y T2, posteriormente en fase de test, una función de *flip* que, invierta la posición espacial de las estructuras, de izquierda a derecha y viceversa, en la primera dimensión espacial, eje x. La alternativa propuesta, descarta un entrenamiento extra de la red, gracias a dicha simetría presentada por las estructuras, de manera que el modelo generado a partir de la información del área derecha de ambas estructuras subcorticales se aplicará para la formación de la imagen de segmentación de la estructura al completo en el test (HA Y NT derecho e izquierdo).

4.4.4 Test

Tras la etapa de entrenamiento, viene la aplicación del test, para comprobar cómo de satisfactorio ha sido el aprendizaje y la capacidad de la red para segmentar las imágenes. Los scripts de test, uno para cada red entrenada, contienen distintas funciones que se pueden aplicar para la obtención de la imagen final de segmentación. Se ha trabajado con 3 funciones distintas: TTDO (test time dropout), TTDO2 (test time dropout) y TTDA (test time *data augmentation*). Las dos primeras funciones utilizan capas de dropout en el test. La diferencia entre las dos, reside en que, la función TTDO2, recibe como entrada un determinado número de iteraciones, que ofrecen estimaciones parecidas, con una cierta variabilidad, y al promediarlas, se obtiene un resultado de segmentación más fiable y con menos variabilidad que si no se hubiera usado dropout. TTDA utiliza *data augmentation* a través de *kernels* que modifican las imágenes, y después promedia para ofrecer el resultado final, que se corresponde con la imagen de segmentación.

Para conocer el acierto en la segmentación ofrecida como resultado final, se calcula el índice Dice, como una medida de coincidencia en las etiquetas entre la imagen de segmentación ofrecida como entrada, y la imagen de salida ofrecida por la red, a nivel de vóxel.

Dado que, en la parte de *test* de la red neuronal no se deben utilizar datos que hayan sido ya vistos por la misma, para ver cómo resuelve un nuevo problema, el conjunto de imágenes que han sido preservadas para el *test* en exclusivo suponen 20 de los 75 casos de los que se partían inicialmente. Cabe resaltar que, la técnica de *data augmentation* con los casos *flip* empleada para el *training*, no se usó en el caso del *test*, por ello sólo ha sido aplicada para el conjunto de entrenamiento.

En el script de *test* se cargan las imágenes, para las que se ha dedicado un directorio especial en una carpeta llamada 'test'. Al cargar las imágenes, se les aplica el mismo *crop* que a las imágenes de entrenamiento y se normalizan los datos. Tal y cómo se ha visto en la descripción del script *Data*, las matrices normalizadas se transforman en tensores (5 dimensiones) y se concatenan para proporcionar la entrada a la red. Se aplica entonces, la función de test, que recibe como entrada la concatenación de las imágenes, y el modelo de red usado para generar la imagen de segmentación.

un número de mapas de características o número de filtros de la imagen, que, en este caso, se declararon 32 como número óptimo.

A partir de esta primera convolución, la red ya no trabaja con la imagen inicial, si no con esos mapas que ha obtenido de ella. Al volver, de nuevo, a convolucionar, se está extrayendo una información que, antes, podía haber sido pasada por alto en la primera operación de convolución. Por otro lado, cada vez que se desciende un nivel en la arquitectura de la red, actúa una capa de *pooling*, que, extraen la información más importante y representativa de cada ventana de observación. El *kernel* de *pooling* se ha establecido de 2×2 . Adicionalmente, se introdujeron capas de *dropout*.

La banda izquierda de la figura 21 constituye la parte de *encoder* de la red. Esta es, la fracción de extracción de la información, y compresión de la misma, dado que, tal y como se puede observar en la figura, las dimensiones de los mapas de características de la imagen son cada vez más pequeñas, pero de un volumen mayor, porque el número de mapas que se extraen por capa es incremental respecto a la profundidad de la red. Contando que la primera capa, la de entrada, recibe el índice 0, se tienen 3 capas de *pooling*, que comienzan en la capa 1, y una capa final latente, que supone la transición entre la fase de *encoder* y la fase de *decoder*.

Tras la primera capa de *pooling*, se duplican los filtros que se le van a aplicar a la imagen, de manera que, el volumen de mapas de características empieza a aumentar. En la capa consecutiva, se multiplicará el número de filtros inicial por 4, hasta alcanzar la última capa, donde, en la convolución, el número de filtros se multiplica por 8.

Cabe resaltar que, a mayor extracción de información de la imagen, mayor trabajo debe realizar el algoritmo, pero también es interesante saber que, las capas más profundas de la red son las que menos coste computacional conllevan, dado que trabajan con datos de dimensiones reducidas.

Una vez alcanzado el espacio latente, comienza la parte de *decoder* en la que se deben ir reduciendo el número de características para llegar a las dimensiones iniciales, de manera inversa, aunque simétrica a como se hace en *encoder*. Se concatenan las salidas de las capas de *down-sampling* con las capas de *up-sampling*, de manera que, la información extraída de la imagen en la primera etapa de entrenamiento, se transfiere a la etapa de *decoder*. Para que esto sea posible, las neuronas correspondientes a estas *layers* están conectadas. Así, en la última capa antes de la presentación del resultado, se ha recuperado el número de filtros iniciales, pero que, ahora, contienen muchos más datos acerca de la imagen que inicialmente.

El resultado generado por la red es una imagen por cada clase posible. Es decir, por cada etiqueta que un vóxel puede adoptar, se va a generar una imagen, que contendrá los pesos o probabilidades de adoptar esa etiqueta en particular. Así, en el caso del tronco encefálico, se tienen 5 clases posibles a las que un vóxel puede pertenecer, incluyendo el fondo. En el caso del hipocampo y la amígdala, el número de clases es 20, y para el núcleo talámico, 26. A través de la concatenación de las imágenes por cada una de las etiquetas, se obtiene el resultado final, la imagen de segmentación.

En la última capa, la generación de las imágenes por clases se produce gracias a la función de activación *softmax*. En contraposición al resto de neuronas en las que se hacía uso de la función de activación ReLU, en la capa final se suele utilizar una capa de convolución de $1 \times 1 \times 1$ con *softmax*, que trabaja con las probabilidades o pesos de cada uno de los vóxeles, asignándoles un valor de etiqueta en función de la distribución de probabilidades.

5. RESULTADOS

Una vez se han preparado los datos, gracias al registro de las imágenes T1 y T2, ha tenido lugar su posterior segmentación con el *software* Freesurfer, y su, de nuevo, etapa de registro para la alineación de las estructuras segmentadas con las imágenes T1 y T2, y, se han definido las arquitecturas de las tres redes, se procede a la puesta en práctica de las capacidades de las mismas.

5.1 Librerías Keras y Tensor Flow

El desarrollo de las redes neuronales que se han utilizado en este TFG, se han implementado gracias a dos librerías principales, Tensor Flow y Keras en el entorno de programación Python. Actualmente, casi la totalidad de lo relacionado con implementación y experimentación de redes y *Deep Learning* se lleva a cabo a través de este lenguaje de programación y estas librerías, por ser una herramienta potente a la vez que interactiva y de uso sencillo para el usuario.

Keras constituye una API desarrollada en Python para el uso, por debajo, de Tensor Flow, cuya complejidad es muy superior. A través de los atributos definidos de Keras, se hace manejo de las funciones de Tensor Flow.

5.2 Resultados del *training*

Los resultados obtenidos provienen de la combinación de parámetros mutables de las redes, para averiguar cuál era la secuencia que proporcionaba mejor resultado o rendimiento de la red. En base a este criterio, la mejora de resultados se ha centrado en variar los valores de las capas de *drop* en el entrenamiento y la función de error que rige la mejora de los pesos para aproximar el modelo a la solución del conjunto de casos de validación. Adicionalmente, en la etapa de *test*, que posteriormente se comentará, se ha probado también con distintos valores de *drop*.

En los primeros entrenamientos de prueba que se realizaron, se observó la existencia de un error en las segmentaciones que la red generaba. Al ser, este error, sistemático en todas las segmentaciones, se consideró que el error no provenía del modelo de la red, sino de los casos que se le estaban proporcionando como conjunto de *test*. Así, se corroboró que las segmentaciones generadas por Freesurfer eran insuficientes en una de las etiquetas del área del tronco encefálico para todos los sujetos, concretamente la correspondiente a la médula (etiqueta 175).

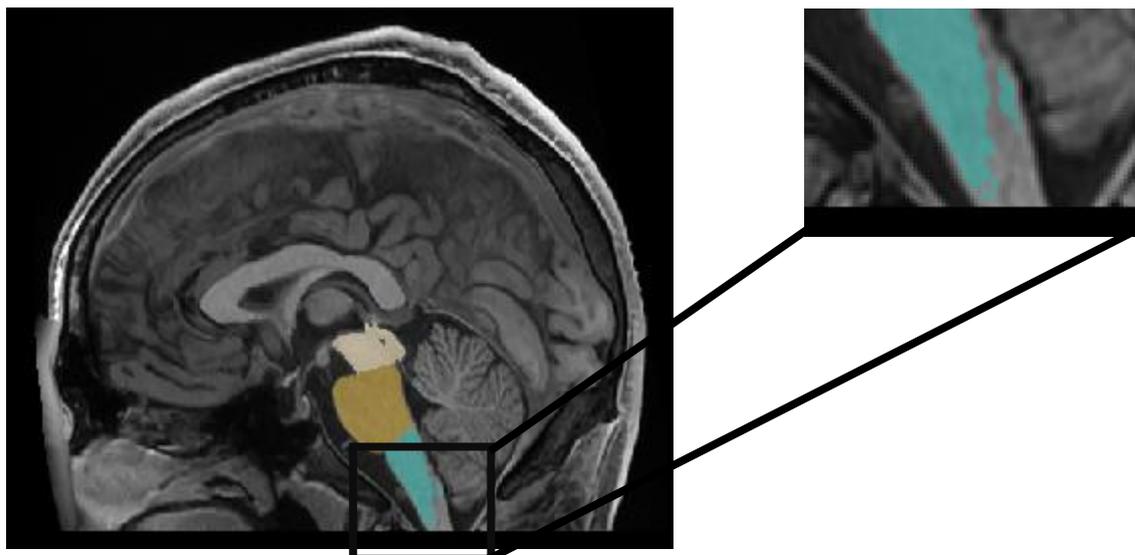


Figura 23: Segmentación deficiente en el área medular del tronco encefálico.

Proporcionar estas imágenes de entrenamiento a la red neuronal del tronco encefálico supondría que la red aprendería de manera errónea la segmentación de este volumen. Por ello, se creó una segunda red con el nombre de 'trainBrainstem2', con 10 imágenes de entrenamiento correspondientes a 10 segmentaciones realizadas a mano con el programa ITK-snap y se usó esa red para corregir las etiquetas erróneas que fueron validadas visualmente.

Una vez solventado el problema con la segmentación del tronco encefálico, se procedió al entrenamiento de esta red, y de las otras dos restantes. La metodología empleada ha sido en los tres casos la misma. En cualquier proceso de experimentación es importante realizar las pruebas variando en cada ocasión solo un parámetro respecto al anterior experimento, dado que, si se varían más de dos variables al mismo tiempo, para la posible modificación en el resultado obtenido, no se conocería la razón de su procedencia, es decir, cuál es la variable que ha influido.

Se comenzó con la versión más simplificada de la red, y es, asignar el valor de 0 al factor de *drop*, de manera que, lo que se está haciendo es ocultar las capas de *drop*. Así, se entrenó la red para las 3 funciones de error. Posteriormente, el valor de *drop* se ascendió a 0.25, y, por último, a 0.5. De nuevo, se volvió a aplicar para cada función de error, generando un total de 23 modelos distintos.

Redes entrenadas		
Tronco encefálico	Drop 0	Mdice loss
		Smooth mdice loss
		GJL smooth (Jaccard)
	Drop 0.25	Mdice loss
		Smooth mdice loss
		GJL smooth (Jaccard)
Drop 0.5	Mdice loss	
Hipocampo y amígdala	Drop 0	Mdice loss
		GJL smooth (Jaccard)
		Mdice loss
	Drop 0.25	GJL smooth (Jaccard)
		Mdice loss
		GJL smooth (Jaccard)
Drop 0.5	Mdice loss	
	GJL smooth (Jaccard)	

Núcleo talámico	Drop 0	Smooth mdice loss
		Mdice loss
		Smooth mdice loss
		GJL smooth (Jaccard)
		Mdice loss
		Smooth mdice loss
	Drop 0.25	GJL smooth (Jaccard)
		Mdice loss
		Smooth mdice loss
		GJL smooth (Jaccard)
		Mdice loss
		Smooth mdice loss
Drop 0.5	GJL smooth (Jaccard)	
	Mdice loss	
	Smooth mdice loss	
	GJL smooth (Jaccard)	
	Mdice loss	
	Smooth mdice loss	

Tabla 4: Totalidad de las redes entrenadas.

Se descartó la idea de seguir experimentando con valores de *drop* superiores a 0.5. Estas capas, explicadas en el apartado 3.3.3.1, suponen asignar valores aleatorios de los mapas de características a cero, para así generar un mayor esfuerzo en la red a la hora del aprendizaje, para evitar que memorice patrones y su comportamiento derive en un indeseado *overfitting*. A pesar de que, su uso es ventajoso, provocar la conversión de una gran cantidad de valores a cero en los mapas, puede suponer un sobreesfuerzo para la red y la incapacidad de llegar a aprender las características más robustas.

Los conjuntos de entrenamiento suponen 105 casos de los 130 totales, incluidos los casos de *data augmentation* generados con la función *flip*. Para el conjunto de validación se han reservado 5 casos. Los parámetros que no han variado para ninguno de los experimentos son: el optimizador, que se ha usado el de tipo *Adam*, y las *epochs*, que se ha establecido un total de 2000, con 20 iteraciones internas por *epoch*.

5.3 Resultados del *test*

El proceso de evaluación de las redes se ha llevado a cabo a través de los scripts de *test* de cada una de ellas. Tal y como se vio en el apartado de metodología, los archivos de *test* contienen 3 funciones de evaluación a las que se puede someter el modelo de pesos generado en el *training*.

En esta etapa de *test*, se ha hecho uso de las distintas funciones combinadas con distintos valores de *drop*. Las capas de *dropout* en la fase de *test* presentan sus ventajas, distintas a las de su aplicación en la fase de entrenamiento. Las funciones son: TTDO, TTDO2 y TTDA. Dado que, las funciones TTDO y TTDO2, se basan en el principio básico del uso de capas de *dropout*, y la función TTDA se basa en la aplicación de modificaciones en la imagen de entrada, con técnicas de *sharpening* y *blurring*, que, seguidamente a la obtención de las imágenes de segmentación para todos los casos, incluidos en los que se ha introducido variabilidad gracias a los filtros, se promedian para ofrecer un resultado más robusto, se consideró relevante comparar las técnicas.

La función TTDA presentó unos datos de índice *Dice* bajos, considerablemente inferiores a los obtenidos con las funciones de *dropout*, por lo que se descartó el uso de esta función. A continuación, la estrategia a seguir fue probar, para cada red entrenada, que combinación de función de *test* con un valor de *dropout* asociado, ofrecía mejores resultados.

Evaluación de la red del tronco encefálico					
	<i>Función de loss (train)</i>	<i>Dropout (train)</i>	<i>Función de test</i>	<i>Dropout (test)</i>	<i>Mean dice</i>
Tronco encefálico	<i>Mdice loss</i>	0	TTDO	0	0.954196
		0.25	TTDO2 it(3)	0.25	0.954208
		0.5	TTDO2 it(3)	0.1	0.955662
				0.25	0.955446
				0.5	0.954884
				TTDO2 it(5)	0.1
	<i>Smooth mdice loss</i>	0	TTDO	0	0.952873
		0.25	TTDO2 it(3)	0.1	0.956015
				0.25	0.955310
				0.1	0.955791
	<i>GJL smooth</i>	0	TTDO	0	0.954917
		0.25	TTDO2 it(3)	0.1	0.956194
				0.2	0.955987

Tabla 5: Conjunto de las pruebas de test realizadas a cada red entrenada del tronco encefálico. El mejor resultado se muestra en negrita.

En el caso del tronco encefálico, el mejor resultado se ha obtenido para una red entrenada con la función de error de Jaccard y un valor de *dropout* de 0.25. De las combinaciones realizadas en el *test*, la aplicación de la función TTDO2 con un número de iteraciones igual a 3 y un valor de *dropout* de 0.1, ha sido la mejor. En la etapa de *test* se busca afinar en el resultado, los decimales, a nivel centesimal o milesimal, van a ofrecer la diferencia entre un método y otro.

Al tratarse, el tronco encefálico, de una estructura con muy pocas etiquetas y sencilla en cuanto a forma, el índice de *mean Dice* que se obtiene sobre 1 ha sido muy alto. El resultado *mean Dice* se calcula como un promedio del índice Dice obtenido para cada uno de los casos del conjunto de *test*. El conjunto de test está formado por 20 casos, por ello, el mdice, es un promedio del *Dice* de 20 casos.

La red del tronco encefálico consta de 5,954,693 parámetros, de los cuales, se han entrenado 5,951,237.

Evaluación de la red del hipocampo y la amígdala						
	<i>Función de loss (train)</i>	<i>Dropout (train)</i>	<i>Función de test</i>	<i>Dropout (test)</i>	<i>Mean dice</i>	
Hipocampo y amígdala	<i>Mdice loss</i>	0	TTDO	0	0.728467	
		0.25	TTDO2 it(3)	0.1	0.760707	
				0.25	0.760539	
				0.5	0.756404	
				0.1	0.759519	
		0.5	TTDO2 it(3)	0.25	0.759039	
				0.5	0.760288	
				0.5	0.761838	
	<i>Smooth mdice loss</i>	0.5	TTDO2 it(3)	TTDO	0	0.761838
				0.1	0.762076	
				0.25	0.762313	
				0.5	0.762739	
				0.75	0.743463	
				0.75	0.743463	

	<i>GJL smooth</i>	0	TTDO	0	0.757355	
		0.25	TTDO2 it(3)	0.1	0.759369	
				0.25	0.759426	
		0.5	TTDO2 it(2)	0.1	0.758989	
				TTDO2 it(3)	0.1	0.759659
					0.25	0.760397
		0.5	0.759634			

Tabla 6: Conjunto de las pruebas de test realizadas a cada red entrenada del hipocampo y la amígdala. El mejor resultado se muestra en negrita.

Al ser, hipocampo y amígdala, estructuras muy diferentes al tronco encefálico, los resultados óptimos para esta área no son los mismos que para la anterior. En este caso, la combinación de la función de *loss smooth mdice* con un valor de *drop* de 0.5 en el *training* y el empleo de la función TTDO2 en el *test* con 3 iteraciones y un valor de *drop* de 0.5, ha proporcionado el mejor resultado.

El índice mean Dice obtenido, es considerablemente menor que en el caso del tronco encefálico, dado que, el número de etiquetas o clases posibles a las que el vóxel puede pertenecer es mayor y la complejidad de la estructura también incrementa.

La red del hipocampo y la amígdala consta de 5,955,188 parámetros, de los cuales, se han entrenado 5,951,732.

Evaluación de la red del Núcleo Talámico					
	<i>Función de loss (train)</i>	<i>Dropout (train)</i>	<i>Función de test</i>	<i>Dropout (test)</i>	<i>Mean dice</i>
Núcleo Talámico	<i>Mdice loss</i>	0	TTDO	0	0.668816
		0.25	TTDO2 it(3)	0.1	0.670118
				0.25	0.669960
		0.5	TTDO2 it(3)	0.1	0.672982
				0.25	0.672397
	<i>Smooth mdice loss</i>	0	TTDO	0	0.671609
		0.25	TTDO2 it(3)	0.1	0.671658
				0.25	0.670253
	0.5	TTDO2 it(3)	0.1	0.670253	
	<i>GJL smooth</i>	0	TTDO	0	0.667352
		0.25	TTDO2 it(3)	0.1	0.668719
				0.25	0.669068
0.5				0.665345	
0.5		TTDO2 it(3)	0.1	0.671789	
		0.25	0.672260		

Tabla 7: Conjunto de las pruebas de test realizadas a cada red entrenada del núcleo talámico. El mejor resultado se muestra en negrita.

Finalmente, la tercera y última de las tres redes evaluadas, la del núcleo talámico, discrepa con las anteriores al presentar el resultado óptimo para la función de *loss mdice* en el *training*, con un valor de *dropout* de 0.5 y, en la fase de *test*, la función TTDO2 con 3 iteraciones y un valor de *drop* de 0.1.

Tal y cómo se veía en el hipocampo y la amígdala, el índice *mean Dice* obtenido es considerablemente más bajo que para el caso del tronco encefálico, y es que, a mayor complejidad de la estructura, en cuanto a anatomía y etiquetado, más le cuesta a la red acertar con exactitud en el segmentado.

La red del núcleo talámico consta de 5,955,386 parámetros, de los cuales, se han entrenado 5,951,930.

Se ha notificado que, el uso de capas de *dropout*, tanto en el *training* como en el *test*, mejora significativamente los resultados. Si se aumentan los valores de *drop* por encima de 0.5, los resultados empeoran. Al igual que, se ha comprobado que el número de iteraciones óptimo para la función TTDO2 de test es 3. Aumentar o disminuir las iteraciones respecto a este número supone una imperfección en el resultado.

5.4 Unión de las redes

Puesto que, uno de los objetivos principales de este TFG es la creación de una red neuronal para la segmentación automática de imágenes, se consideró juntar las 3 redes, una para cada estructura, en una única. Con esta idea, se creó un nuevo *script* en Python con la fusión de los tres *scripts* de test de las estructuras. Este fichero recibe el nombre de 'test_FULL' y carga cada uno de los modelos de los pesos óptimos para cada estructura y ejecuta el *test* de cada una de ellas, para, posteriormente, ofrecer una imagen de segmentación final como la integración o unión de las segmentaciones de las estructuras por separado.

Para este script se decidió separar los lados derecho e izquierdo de las estructuras del hipocampo y amígdala y núcleo talámico, para que se generase una segmentación para cada uno de los lados, y tuvieran etiquetas diferenciadas para lados derecho e izquierdo, con distintos valores. El hipocampo y la amígdala presentaban el mismo valor de etiquetas para ambos lados. Para el lado izquierdo se asignó, en consecuencia, los valores de las etiquetas más 100, para generar la diferenciación. Así, se obtienen 5 segmentaciones: tronco encefálico, hipocampo y amígdala izquierdo, hipocampo y amígdala derecho, núcleo talámico izquierdo y núcleo talámico derecho.

Por último, se integran estas 5 segmentaciones en una única imagen de segmentación que ofrece la segmentación en su totalidad de las 3 áreas de interés para cada sujeto. El coste computacional del método propuesto es de unos 10 segundos.

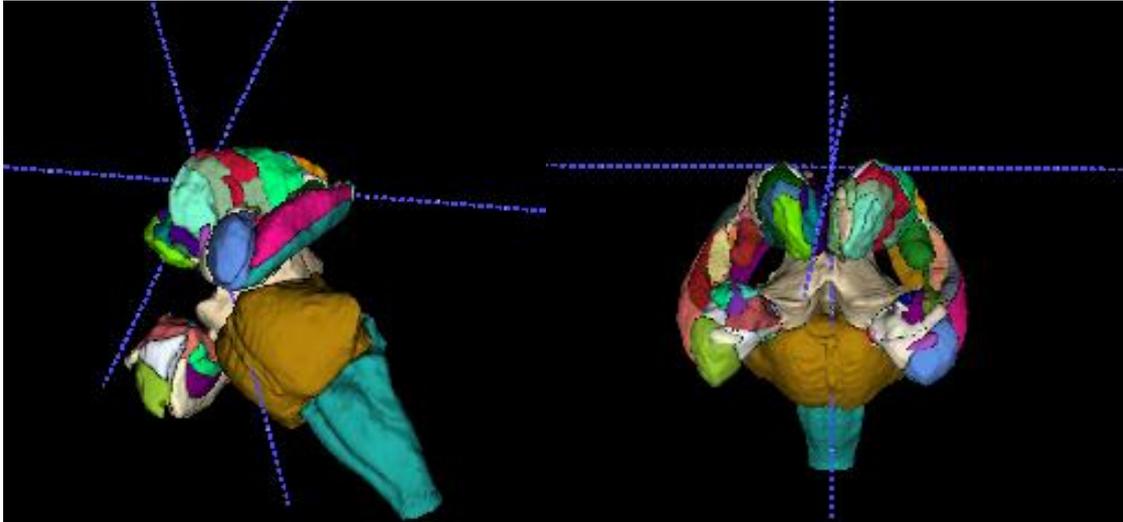


Figura 24: Segmentación del volumen completo de las 3 estructuras.

Adicionalmente, en la fase de *test* se ha aplicado una función de post procesado. Esta función tiene como objetivo eliminar pequeñas segmentaciones que se generan en zonas incoherentes, es decir, hay regiones que aparecen con una etiqueta asignada en lugares que no se corresponden y que, por tanto, son incorrectos. Estos pequeños fallos en la segmentación se pueden corregir aplicando un post procesado que, se quede con la estructura de mayor área de las que se presentan en la imagen. Así, se está seleccionando el volumen de interés, correspondiente a la estructura subcortical y se está dejando atrás esas pequeñas segmentaciones ruidosas. La mencionada función de post procesado se aplica a todas las estructuras, las 5.

A la segmentación completa final, que ya las integra a todas, se le aplica otro tipo de post procesado, una segunda función que impone un filtro para los bordes de la segmentación. La finalidad es eliminar los vóxeles sueltos que aparecen en los bordes de la estructura de manera saliente, que impiden que la región quede bien definida y uniforme, otorgando así un aspecto más preciso en la segmentación.

La modificación que se ha aplicado a este *script* de *test* conjunto, en relación a los anteriores, es que se ha eliminado completamente la parte correspondiente a las etiquetas, ya que no interesa calcular ningún tipo de índice *Dice* de éxito, pues se están cargando los modelos con los pesos que han proporcionado ya, los resultados óptimos.

Ha sido necesaria la creación de una pequeña función adicional, que reestablece el número asignado a las etiquetas generadas por la red, que se ordenan con números consecutivos empezando desde el 0, a los números originales de las etiquetas, que, eran ciertamente aleatorios.

El *script* 'test_FULL' se ha aplicado con un conjunto de datos de *test* igual al *database* completo de los 75 sujetos (se eliminó un caso de los 76 iniciales por errores en las imágenes de partida), dejando fuera los casos *flip*. Así se ha generado una segmentación completa para cada sujeto.

5.5 Implementación para la mejora de resultados de la red

Finalmente, el último paso de este TFG consiste en la búsqueda de mejora de la calidad de las segmentaciones generadas por la red. Por ello, se planteó la idea de que, si se realizaba un cambio en el conjunto de datos de entrenamiento que se le estaban proporcionando a la red, siendo estos datos de entrenamiento de mejor calidad que las imágenes utilizadas inicialmente para su *training*, correspondientes a las segmentaciones de Freesurfer, los resultados de la CNN serían óptimos. Así, se ha entrenado una nueva red desde 0, para cada ROI, con un conjunto de datos de entrenamiento correspondientes a las segmentaciones generadas para los 75 casos que componen el *database*, a través del script 'test_FULL'. Se han tenido en cuenta para este *training* las consideraciones de cuales habían resultado las combinaciones óptimas en la anterior etapa, así, se ha entrenado directamente con ellas y no ha sido necesaria una nueva fase de experimentación.

Se debe tener en cuenta que, a pesar de que los datos de *training* que se le están proporcionando a la red, en esta ocasión, son de mejor calidad, no dejan de ser imágenes que la propia red ha generado, por lo que el aprendizaje, sobre su propio aprendizaje, alcanzará a mejorar los resultados en la aplicación posterior del *test*, ofreciendo segmentaciones más robustas y sólidas.

En el entrenamiento de la red del tronco encefálico (con el nuevo conjunto de *training*), se ha obtenido la gráfica de la figura 25, correspondiente al error del modelo y el error de validación en el *training*. Dado que, el problema de segmentación del tronco encefálico es muy sencillo, pues solo se divide en 5 etiquetas, el error en el conjunto de *training* es muy parecido al del conjunto de validación. Se presentan errores muy bajos en ambos casos, lo que indica que la red es muy capaz de afrontar este problema de segmentación y llevarlo a cabo con éxito. Por esta razón, precisamente, se puede afirmar que en ningún momento se ha producido overffiting en los casos de entrenamiento.

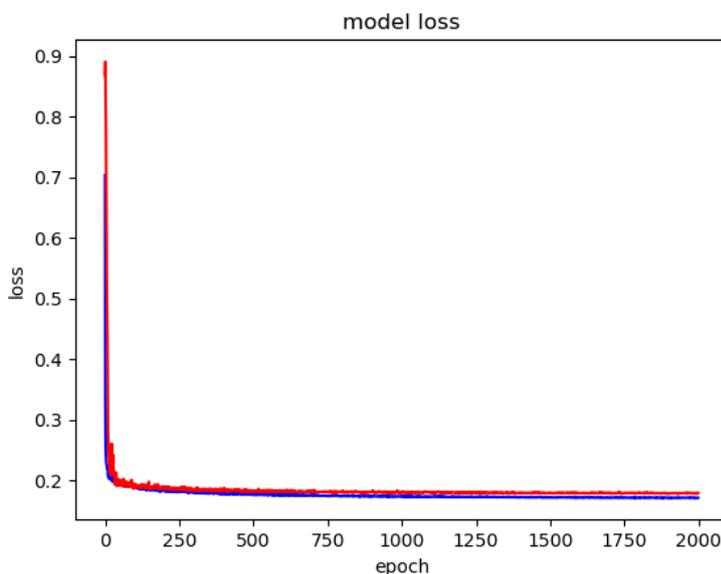


Figura 25: Gráfica de la evolución del error en el conjunto de entrenamiento (azul) y el conjunto de validación (rojo) en relación a las *epochs* para el tronco encefálico.

En el caso del *training* para el hipocampo y la amígdala, se ha obtenido la gráfica de la figura 26, correspondiente al error del modelo y el error de validación en el *training*. En comparación al

problema de segmentación del tronco encefálico, el hipocampo y la amígdala resulta ser un dilema de mayor magnitud, ya que el número de regiones en las que se divide la estructura es mayor, y el error se ve más dispar entre los distintos conjuntos (*training* y validación) Aún así, el aprendizaje de la red sigue siendo óptimo, pues no se observa fenómeno de *overfitting*.

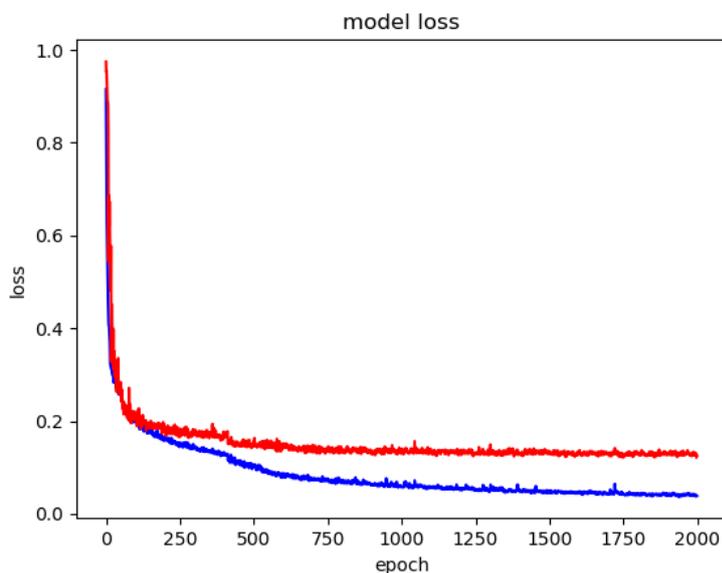


Figura 26: Gráfica de la evolución del error en el conjunto de entrenamiento (azul) y el conjunto de validación (rojo) en relación a las *epochs* para el hipocampo y la amígdala.

Por último, en el núcleo talámico, vemos un error de *training* y test similares al caso anterior, donde, el problema de segmentación es más complejo, por un mayor número de etiquetas, pero el aprendizaje continúa siendo óptimo. La ausencia de *overfitting* en el entrenamiento viene justificada, en su mayoría, por las capas de *dropout* que se han aplicado en la red.

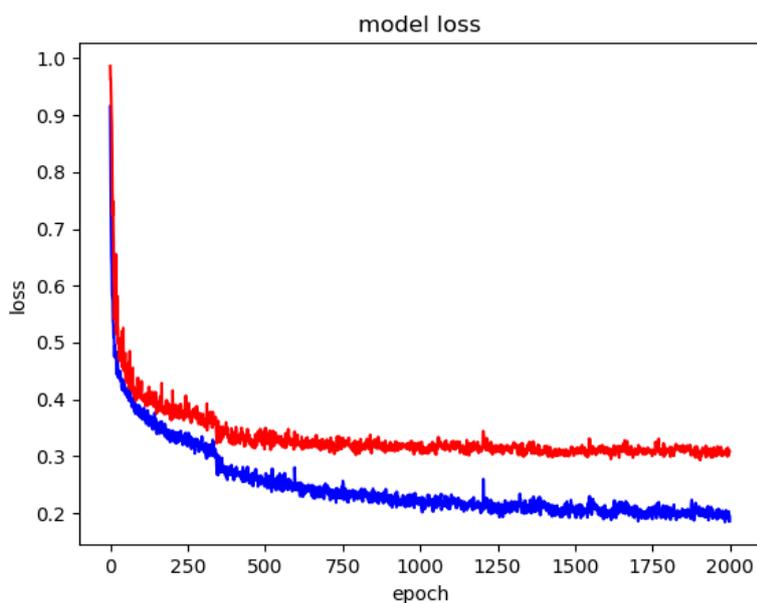


Figura 27: Gráfica de la evolución del error en el conjunto de entrenamiento (azul) y el conjunto de validación (rojo) en relación a las *epochs* para el núcleo talámico.

Finalmente, los resultados de segmentación generados por la red definitiva, describen un volumen de tronco encefálico, hipocampo y amígdala y núcleo talámico bien definido y sin ruido, siendo segmentaciones robustas y muy ajustadas a las verdaderas áreas de estas estructuras en el cerebro humano.

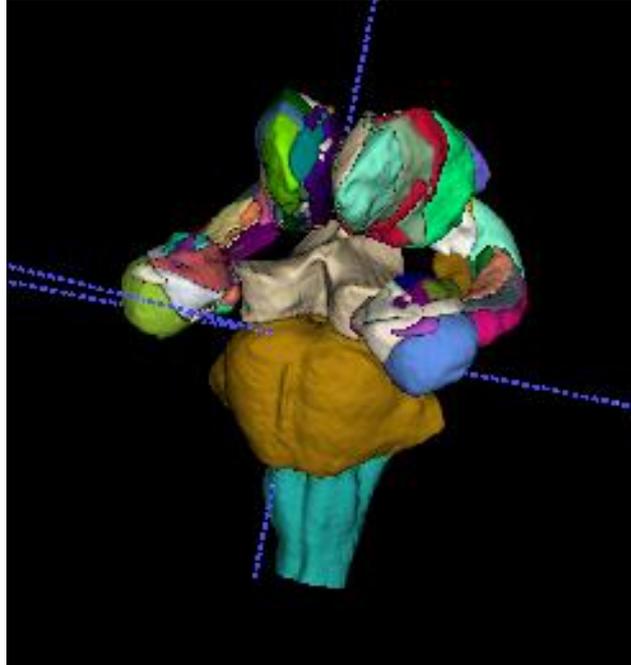


Figura 25: Segmentación del volumen completo de las 3 estructuras mejorada.

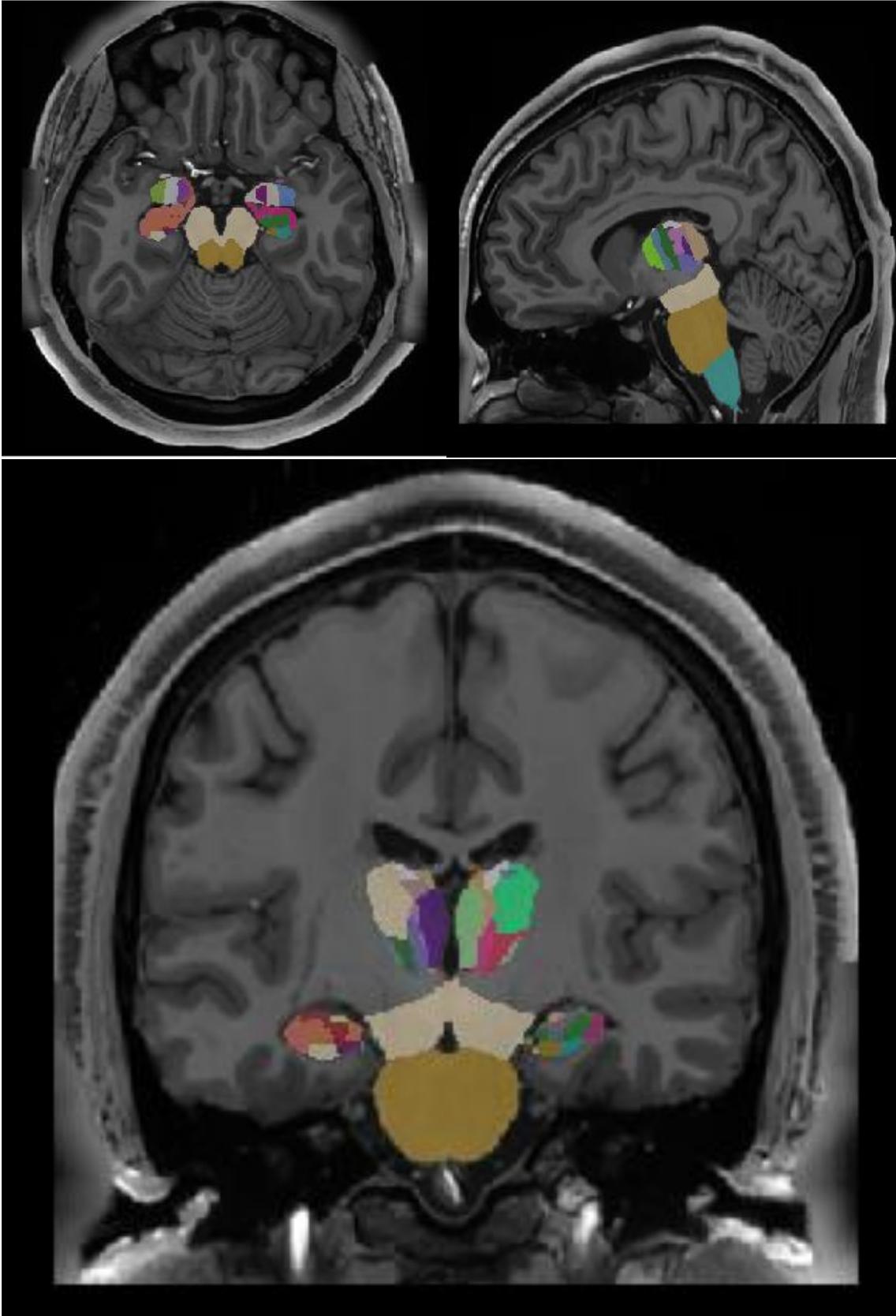


Figura 26: Visualización de la segmentación en los 3 ejes ortogonales.

6. DISCUSIÓN Y CONCLUSIONES

Este Trabajo Fin de Grado ha propuesto la segmentación de estructuras subcorticales mediante *Deep Learning* a través del uso de imágenes cerebrales de tipo T1 y T2 de resonancia magnética. Aún partiendo de las imágenes de segmentación de dichas estructuras, generadas por el *software* Freesurfer, en las que se ha comprobado que la segmentación es deficiente en muchos de los casos, el uso de redes neuronales para la mejora de la tarea de segmentado de las estructuras cerebrales, que son en su mayoría, complejas, ha sido fructífero.

La CNN utilizada, resultado de una modificación del modelo U-Net ha sido capaz de generar segmentaciones más robustas y completas que las obtenidas en primera estancia con Freesurfer. A pesar de partir de unos datos de entrenamiento insuficientes, el aprendizaje de la red de las características de las estructuras en la imagen, ha dado como resultado una mejora en la eficiencia de la técnica de segmentación de *single-atlas* de la que hace uso el *software* Freesurfer.

Tal y como se ha visto en el apartado de resultados, el problema de segmentación del tronco encefálico es el más sencillo de todos, y en el que se ha conseguido un mejor índice de acierto (*mean Dice*). La segmentación de las estructuras del hipocampo y amígdala y el núcleo talámico resulta más costosa, por su mayor complejidad en cuanto a número de etiquetas en las que se dividen. Así, los valores de *mean Dice* obtenidos, son significativamente menores que en el caso del tronco encefálico, aún así, se ha conseguido alcanzar un índice del 0.76 en el hipocampo y la amígdala y un índice de 0.67 en el núcleo talámico.

Las capas de *dropout* suponen una mejora indiscutible en los resultados, en comparación a la aplicación del modelo de red sin la utilización de las mismas. Se ha podido comprobar en la obtención de resultados de segmentación el índice *mean Dice* mejora significativamente si se hace uso de las capas de *dropout*, especialmente si se emplean valores de entre 0.1 y 0.5. Minimizan la aparición del indeseado fenómeno de *overfitting* en el entrenamiento de las redes.

Por otro lado, los resultados obtenidos con la CNN, son mejorables aportando unos datos de entrenamiento de mayor calidad, entre otros factores. Así, influyen de manera directa la metodología de adquisición de las imágenes, así como el proceso de registro de las mismas, que también juegan un papel muy importante en la calidad de los datos. La herramienta de registro ANTs utilizada en este TFG, y que supone parte del estado del arte de esta disciplina, ha ofrecido unos buenos resultados de registro.

Finalmente, las redes neuronales, suponen así una herramienta muy potente y con una gran capacidad de progreso, realmente buena en el reconocimiento de patrones. Aporta una gran ayuda a la tarea de segmentación de estructuras subcorticales cerebrales que resultan realmente de ayuda en el diagnóstico precoz de enfermedades mentales.

7. LÍNEAS FUTURAS

Las conclusiones obtenidas tras la experimentación, recogen unos resultados fructíferos, aunque posiblemente mejorables con la implementación del método. Una posible acción a realizar es la utilización de las estructuras del hipocampo y amígdala y núcleo talámico en su hemisferio izquierdo para el entrenamiento de las redes. Tal y como se indicó en el apartado de *training* de las redes, únicamente se entrenó el lado derecho de las estructuras tras realizar el *crop* a las imágenes con las coordenadas indicadas para esa región. Posteriormente, se aplicó el conocimiento adquirido de la red para esa zona, al lado izquierdo, debido a la hipótesis de simetría que se asumió. Haber entrenado a la red con el hemisferio izquierdo de estas estructuras subcorticales, hubiera supuesto una mayor extracción de información.

Paralelamente, se debe tener en consideración que las segmentaciones que se han utilizado en este TFG para el entrenamiento de las redes son segmentaciones automáticas, generadas por un software de segmentación, Freesurfer. Es por este motivo, que no se puede asegurar el hecho de que las segmentaciones sean correctas en su totalidad, ya que no han sido segmentadas por un especialista en imagen médica cerebral y por ello no se dispone de un conjunto de *ground truth*.

A modo de línea futura, las últimas imágenes que se han obtenido en este TFG, que se corresponden con las segmentaciones finales conseguidas, podrían ser analizadas por un experto en la materia y complementadas por sus conocimientos, para crear así un nuevo *database* con el que entrenar la red.

BIBLIOGRAFÍA:

- Abderrahim, M., Baâzaoui, A., & Barhoumi, W. (2020). Comparative Study of Relevant Methods for MRI/X Brain Image Registration. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12157 LNCS, 338–347. https://doi.org/10.1007/978-3-030-51517-1_30
- Avants, B. B., Tustison, N., & Johnson, H. (2002). Advanced Normalization Tools (ANTS). *Biochemistry. 5th Edition.*, Section 16.2.
- Bereciartua Pérez, A. (2016). *Desarrollo de algoritmos de procesamiento de imagen avanzado para interpretación de imágenes médicas. Aplicación a segmentación de hígado sobre imágenes de Resonancia Magnética multiseuencia. 2016(cc).*
- Berman, M., Triki, A. R., & Blaschko, M. B. (2018). The Lovasz-Softmax Loss: A Tractable Surrogate for the Optimization of the Intersection-Over-Union Measure in Neural Networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 4413–4421. <https://doi.org/10.1109/CVPR.2018.00464>
- Bonsignore, F. (2004). Información en el registro de Imágenes Médicas. *XIII Seminario de Ingeniería Biomédica, 2004.*, 1–5. <http://www.nib.fmed.edu.uy/bonsignore.pdf>
- Brownlee, J. (2017). *What is the Difference Between Test and Validation Datasets?* <https://machinelearningmastery.com/difference-test-validation-datasets/>
- Brownlee, J. (2019). *A Gentle Introduction to Pooling Layers for Convolutional Neural Networks.* <https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>
- Calvo, D. (2017). *Red Neuronal Convolutional CNN.* <https://www.diegocalvo.es/red-neuronal-convolucional/>
- Enrique, J., & Gómez, R. (2018). *Nuevos métodos para el análisis automático del volumen de estructuras cerebrales a partir de imágenes de resonancia magnética nuclear.*
- Filoteo, J. V., Reed, J. D., Litvan, I., & Harrington, D. L. (2014). Volumetric correlates of cognitive functioning in nondemented patients with Parkinson's disease. *Movement Disorders*, 29(3), 360–367. <https://doi.org/10.1002/mds.25633>
- Fischl, B., Salat, D. H., Busa, E., Albert, M., Dieterich, M., Haselgrove, C., van der Kouwe, A., Killiany, R., Kennedy, D., Klaveness, S., Montillo, A., Makris, N., Rosen, B., & Dale, A. M. (2002). Whole brain segmentation: Automated labeling of neuroanatomical structures in the human brain. *Neuron*, 33(3), 341–355. [https://doi.org/10.1016/S0896-6273\(02\)00569-X](https://doi.org/10.1016/S0896-6273(02)00569-X)
- Fischl, B., Salat, D. H., van der Kouwe, A. J. W., Makris, N., Ségonne, F., Quinn, B. T., & Dale, A. M. (2004). Sequence-independent segmentation of magnetic resonance images. *NeuroImage*, 23(SUPPL. 1), 69–84. <https://doi.org/10.1016/j.neuroimage.2004.07.016>
- FreeSurfer Tutorial.* (n.d.). https://andysbrainbook.readthedocs.io/en/latest/FreeSurfer/FS_ShortCourse/FS_03_ReconAll.html

- Gil, D., & Udiat, B. (n.d.). *Semiología básica de la resonancia magnética*.
- GitHub-Predicciones con redes neuronales. (n.d.). <https://github.com/mariajosemv/Predicciones-con-Redes-Neuronales>
- Hajnal, J. v., Hill, D. L. G., & Hawkes, D. J. (2001). Medical image registration. *Medical Image Registration*, 1–383. <https://doi.org/10.1051/ejn:2000401>
- Hinton, G., Srivastava, N., Sutskever, I., Krizhevsky, A., & Salakhutdinov, R. (1993). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Physics Letters B*, 299(3–4), 345–350. [https://doi.org/10.1016/0370-2693\(93\)90272-J](https://doi.org/10.1016/0370-2693(93)90272-J)
- Human Connectome Project. (2017). <https://humanconnectome.org/study/hcp-young-adult/document/1200-subjects-data-release/>
- Iglesias, J. E. (n.d.-a). *Brainstem Substructures*. <https://surfer.nmr.mgh.harvard.edu/fswiki/BrainstemSubstructures>
- Iglesias, J. E. (n.d.-b). *Segmentation of hippocampal subfields and nuclei of the amygdala (cross-sectional and longitudinal)*. <https://surfer.nmr.mgh.harvard.edu/fswiki/HippocampalSubfieldsAndNucleiOfAmygdala>
- Iglesias, J. E. (n.d.-c). *Segmentation of thalamic nuclei*. <https://freesurfer.net/fswiki/ThalamicNuclei>
- Joynt, R. (n.d.). *Brainstem*. <https://www.britannica.com/science/brainstem>
- Killiany, R. J., Gomez-Isla, T., Moss, M., Kikinis, R., Sandor, T., Jolesz, F., Tanzi, R., Jones, K., Hyman, B. T., & Albert, M. S. (2000). Use of structural magnetic resonance imaging to predict who will get Alzheimers disease. *Annals of Neurology*, 47(4), 430–439. [https://doi.org/10.1002/1531-8249\(200004\)47:4<430::AID-ANA5>3.0.CO;2-I](https://doi.org/10.1002/1531-8249(200004)47:4<430::AID-ANA5>3.0.CO;2-I)
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2020). ImageNet Classification with Deep Convolutional Neural Networks. *ACM International Conference Proceeding Series*, 145–151. <https://doi.org/10.1145/3383972.3383975>
- Lafuente Martínez, J., & Hernández Moreno, L. (2017). Técnica De La Imagen Por Resonancia Magnética. *Resonancia Magnética Del Sistema Músculo-Esquelético.*, 9–13.
- Linux Install & Setup. (n.d.). https://surfer.nmr.mgh.harvard.edu/fswiki/FS7_linux
- Manjón, J. v., & Coupé, P. (2016). Volbrain: An online MRI brain volumetry system. *Frontiers in Neuroinformatics*, 10(JUL), 1–14. <https://doi.org/10.3389/fninf.2016.00030>
- Meléndez Cuauero, R. (2016). *Resonancia Magnética (RM)*. <https://es.slideshare.net/melendezcuauero/resonancia-magntica-rm>.
- Penadés Migallón, J. H. (2020). *Re-identificación de personas a partir de múltiples capturas mediante aprendizaje automático*.
- Ros García, A. (2019). *Escuela Técnica Superior de Ingeniería Industrial Sistema de percepción de elementos viarios usando técnicas de visión por computador para aplicación en conducción autónoma*.
- Ruíz, L. (2021). *El hipocampo: qué es, anatomía y funciones*. <https://www.psyciencia.com/el-hipocampo-que-es-anatomia-y-funciones/>

- Triglia, A. (n.d.). *Amígdala cerebral: qué es, funciones y anatomía*.
<https://azsalud.com/medicina/amigdala-cerebral>
- Uchida, S. (2013). Image processing and recognition for biological images. *Development Growth and Differentiation*, 55(4), 523–549. <https://doi.org/10.1111/dgd.12054>
- Velasco, L. (2020). *Optimizadores en redes neuronales profundas: un enfoque práctico*.
<https://velascoluis.medium.com/optimizadores-en-redes-neuronales-profundas-un-enfoque-practico-819b39a3eb5>
- Wikipedia. (2019). *Coeficiente de Sorensen-Dice*.
https://es.wikipedia.org/wiki/Coeficiente_de_Sorensen-Dice
- Wikipedia. (2021). *Red neuronal convolucional*.
https://es.wikipedia.org/wiki/Red_neuronal_convolutiva
- Zhang, Y. J. (1996). A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8), 1335–1346. [https://doi.org/10.1016/0031-3203\(95\)00169-7](https://doi.org/10.1016/0031-3203(95)00169-7)



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

Parte II

Presupuesto

Sánchez Soto, Paula
Manjón Herrera, José Vicente

ÍNDICE DE PRESUPUESTO

1.1 Cuadro de mano de obra	3
1.2 Cuadro de <i>hardware</i> y <i>software</i>	4
1.3 Presupuesto de ejecución por contrata	5

En este último punto del trabajo, se ha llevado a cabo un cálculo económico del proyecto realizado.

La valoración económica efectuada, se basa en el coste de mano de obra del ingeniero biomédico estudiante, que ha realizado el proyecto en su mayor parte, el coste de mano de obra del ingeniero superior, que se corresponde con el tutor del trabajo y el coste de las herramientas *hardware* y *software* utilizadas en este TFG. Se han tenido en cuenta los precios de venta de estos instrumentos. En esencia, se ha empleado un ordenador y una GPU, respecto a la parte *hardware* y el sistema operativo Windows 10 Enterprise, junto con la licencia del *software* Matlab R2019b y los programas de ITK-snap, ANTs para registro de imágenes, Anaconda y Freesurfer para segmentación, en la parte *software*.

1.1 Cuadro de mano de obra

CUADRO DE MANO DE OBRA			
Denominación	Precio (euros/hora)	Cantidad de horas	Total (euros)
Tutor del Trabajo de Fin de Grado responsable de la guía y comprobación del proyecto	40,00	100,00	4.000,00
Ingeniero Biomédico encargado de realizar las tareas para llevar a cabo el Trabajo de Fin de Grado “SEGMENTACIÓN DE ESTRUCTURAS SUBCORTICALES USANDO DEEP LEARNING”	27,50	300,00	8.250,00
Importe Total de Mano de Obra:			12.250,00

Tabla 8: Tabla de precios de mano de obra.

1.2 Cuadro de *hardware* y *software*

CUADRO DE HARDWARE Y SOFTWARE					
Denominación	Precio (euros)	Vida útil (meses)	Coste amortizado (euros/meses)	Uso (meses)	Total (euros)
Ordenador de sobremesa	1.000,00	60	16,66	8	133,28
Tarjeta Gráfica	780,00	60	13	8	104,00
Microsoft Windows 10 Enterprise	259,00	48	5,39	8	43,12
Matlab R2019b	250,00	12	20,83	3	62,49
ITK-Snap 3.6	0,00				0,00
Advanced Normalization Tools (ANTs)	0,00				0,00
Anaconda 3.7	0,00				0,00
Freesurfer	0,00				0,00
Importe Total Maquinaria y Software: 342,89					

Tabla 9: Tabla de precios de *hardware* y *software*.

1.3 Presupuesto de ejecución por contrata

PRESUPUESTO TOTAL	
COSTE	IMPORTE (euros)
Coste de Mano de Obra	12.250,00
Coste de Hardware y Software	342,89
13% gastos generales	1.637,07
6% beneficio general	755,57
Suma	14.985,53
21% IVA	3.146,96
Presupuesto de ejecución por contrata	18.132,49

Tabla 10: Tabla de presupuesto de ejecución por contrata.

Finalmente, el coste económico del proyecto asciende a la cantidad de DIECIOCHO MIL CIENTO TREINTA Y DOS CON CUARENTA Y NUEVE.