

2017

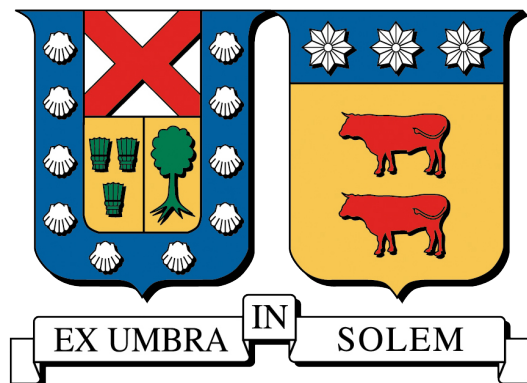
REDES NEURONALES DE CONVOLUCIÓN APLICADAS A LA SEGMENTACIÓN DE LESIONES DE ESCLEROSIS MÚLTIPLE EN IMÁGENES DE RESONANCIA MAGNÉTICA 3D

NARANJO COLLAO, RODRIGO ESTEBAN

<http://hdl.handle.net/11673/23165>

Repositorio Digital USM, UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



**Redes Neuronales de Convolución aplicadas a la segmentación de lesiones de
Esclerosis Múltiple en Imágenes de Resonancia Magnética 3D**

Rodrigo Esteban Naranjo Collao

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

PROFESOR GUÍA : Héctor Allende Olivares
PROFESOR CORREFERENTE : Gustavo Ulloa Poblete

RESUMEN EJECUTIVO

La Esclerosis Múltiple es una enfermedad crónica que afecta al sistema nervioso central. El diagnóstico de esta enfermedad se basa en el uso de Imágenes de Resonancia Magnética para la detección las lesiones propias de esta. Este proceso de detección suele ser realizado de manera manual por expertos, sin embargo, existe el interés en automatizar esta tarea con el fin de agilizar el proceso de diagnóstico y seguimiento de esta enfermedad. Con este fin, se han propuesto múltiples técnicas automáticas de segmentación de imágenes que permitan detectar de manera efectiva estas lesiones. En esta memoria, se estudiará el desempeño de dos técnicas del Estado del Arte aplicadas al problema de detección de lesiones de Esclerosis Múltiple en Imágenes de Resonancia Magnética.

La primera técnica se basa construcción de un Modelo de Mezcla de Gaussianas, con el fin de detectar *outliers* en cada imagen. Luego, se aplica una serie de reglas basadas en el conocimiento experto de las lesiones de Esclerosis Múltiple para descartar aquellos *outliers* del modelo que no presenten características propias de las lesiones. Esta técnica permite obtener una segmentación de las lesiones de manera muy eficiente en términos de tiempo, sin embargo sufre del problema de subestimar de gran manera el tamaño de las lesiones. Además de esto, las lesiones segmentadas suelen corresponder a regiones de materia gris del cerebro. En algunos casos es posible detectar estos falsos positivos, sin embargo el conjunto de reglas aplicado no logra eliminar todos estos errores sin comprometer la calidad de la segmentación.

La segunda técnica consiste en el entrenamiento de una Red Neuronal Convolutiva que logre segmentar las lesiones en base a un conjunto de imágenes de entrenamiento. Esta técnica es capaz de aprender una serie de filtros que, aplicados a pequeñas secciones de la imagen denominadas "parches", permite obtener una serie de características que permite clasificar cada vóxel de la imagen como lesión o tejido saludable. Este tipo de redes requiere de una gran cantidad de tiempo para procesar un gran número de imágenes, por lo cual se aplicaron técnicas de submuestreo de parches para disminuir el tiempo de entrenamiento. Los resultados obtenidos muestran que las redes son capaces de obtener resultados similares en calidad a los algoritmos del Estado del Arte en la base de datos trabajada, sin embargo su capacidad predictiva está limitada por el muestreo inicial de parches.

Palabras Claves: Imagen de Resonancia Magnética, Esclerosis Múltiple, Modelo de Mezcla de Gaussianas, Redes Neuronales Convolutivas .

ABSTRACT

Multiple Sclerosis is a chronic disease which affects the central nervous system. The diagnosis of this disease is based on the use of Magnetic Resonance Imaging for the detection of lesions caused by it. The process of detection is usually performed manually by experts, however, there exists interest in the automation of this task in order to speed up the process of diagnosis and monitoring of this disease. To this end, multiple automated segmentation techniques have been proposed to effectively detect these lesions. In this undergraduate thesis, the performance of two State of the Art techniques applied to the problem of Multiple Sclerosis lesion detection in Magnetic Resonance Imaging will be studied.

The first technique is based on the construction of a Gaussian Mixture Model, with the goal of detecting outliers on each image. Then, a series of rules based on expert knowledge of Multiple Sclerosis lesions is applied to discard the outliers which don't present characteristics said lesions. This technique allows to obtain a segmentation of the lesions in a very efficient way in regards of time, however it suffers from the problem of greatly underestimating the size of the lesions. In addition to this, segmented lesions usually correspond to gray matter regions of the brain. In some cases it's possible to detect these false positives, however the applied ruleset is incapable of eliminating all these errors without compromising the quality of the segmentation.

The second technique consists in training a Convolutional Neural Network which learns to segment the MS lesions from an image training set. This technique is capable of automatically learning a series of filters which, when applied to small sections of the image called "patches", allows to obtain a series of characteristics of the image which can be used to classify each voxel of the image as lesion or healthy tissue. This type of networks requires a large amount of time to process large amounts of training patches, which is why a subsampling technique was applied to decrease training time. The results obtained show that CNNs are capable of obtaining performance similar to that of State of Art algorithms on the dataset utilized, however its predictive capabilities are limited by the patch subsampling method.

Keywords: Magnetic Resonance Imaging, Multiple Sclerosis, Gaussian Mixture Model, Convolutional Neural Networks.

GLOSARIO

- **MRI:** Magnetic Resonance Image.
- **FLAIR:** Fluid Attenuated Inversion Recovery.
- **WM:** White Matter.
- **GM:** Gray Matter.
- **CSF:** Cerebrospinal Fluid.
- **GMM:** Gaussian Mixture Model.
- **EM:** Expectation Maximization.
- **CNN:** Convolutional Neural Network.
- **DNN:** Deep Neural Network.
- **UNC:** University of North Carolina.
- **CHB:** Boston Children's Hospital.
- **TPR:** True Positive Rate.
- **PPV:** Positive Predictive Value.
- **DSC:** Dice Similarity Coefficient.
- **MP:** Max Pooling

Índice de Contenidos

1. Contextualización y Definición del Problema	1
1.1. Objetivos	2
1.1.1. Objetivo General	2
1.1.2. Objetivos Específicos	3
2. Estado del Arte	4
2.1. Imágenes de Resonancia Magnética	4
2.2. Uso de MRI en el diagnóstico de Esclerosis Múltiple	5
2.3. Umbralización	6
2.3.1. Distribución Conocida	7
2.3.2. Picos y Valles	7
2.3.3. Clustering (Otsu)	7
2.3.4. Umbralización en Segmentación de Lesiones de Esclerosis Múltiple	8
2.4. Métodos basados en Atlas Probabilísticos	8
2.5. Gaussian Mixture Models	11
2.5.1. Detección de Lesiones como Outliers en un GMM	12
2.5.1.1. Estimación de Parámetros	12
2.5.1.2. Inicialización Jerárquica	14
2.5.1.3. Verosimilitud Truncada	14
2.5.1.4. Selección de Lesiones a partir de Outliers	15
2.6. Redes Neuronales Convolucionales	16
2.6.1. Muestreo de Parches no Uniformes	21
2.6.2. Redes Longitudinales Multicanal	23
3. Propuesta de Solución	25
4. Construcción e Implementación de la solución	27
4.1. Preprocesamiento	27
4.1.1. Bias Field Correction	27
4.1.2. Skull Stripping	28
4.2. Gaussian Mixture Models	29
4.2.1. Algoritmo K-Means	31
4.3. Redes Neuronales Convolucionales	32

5. Resultados	35
5.1. Postprocesamiento	44
6. Conclusiones	47
7. Trabajo Futuro	49
Bibliografía	50

Índice de Tablas

4.1. Parámetros usados en el BSE	29
4.2. Parámetros de medias obtenidos para la Materia Blanca	31
5.1. Comparación de PPV antes y después de aplicación de Erosión a la máscara de Materia Blanca	36
5.2. Resultados de técnicas del Estado del Arte	38
5.3. Resultados de técnicas implementadas	38
5.4. Número de lesiones segmentadas	40
5.5. Número de regiones segmentadas	41
5.6. Tiempos de ejecución para las redes convolucionales en la CPU	43
5.7. Tiempos de ejecución para las redes convolucionales en la GPU	44
5.8. Resultados de postprocesamiento.	45
5.9. Cambios en las métricas tras la aplicación del postprocesamiento.	45

Índice de Figuras

2.1. Ejemplo de Imagen T1	5
2.2. Ejemplo de Imagen T2	5
2.3. Ejemplo de Imagen T2-FLAIR	5
2.4. Gráficas de las funciones de activación comúnmente utilizadas.	19
2.5. Derivadas de las funciones de activación.	19
2.6. Conexiones en una Feedforward Neural Network.	20
2.7. Conexiones en una Convolutional Neural Network (kernel 3×3 , vista 1D).	20
2.8. Comparación de las conexiones entre neuronas de una FNN y CNN	20
2.9. Muestreo de pixeles centrales propuesto [12]	22
4.1. Imagen FLAIR Original	29
4.2. Imagen tras aplicar <i>Bias Field Correction</i>	29
4.3. Imagen tras aplicado el <i>Skull Stripping</i>	29
4.4. Arquitectura para 1 entrada implementada de la CNN. Elaboración Propia.	33
4.5. Arquitectura para 1 entrada implementada de la CNN sin capa de <i>Max Pooling</i> . Elaboración Propia.	33
5.1. Parámetro h	35
5.2. Parámetro d_{min}	35
5.3. Parámetro k_{T2}	35
5.4. Parámetro k_{FLAIR}	35
5.5. Segmentación obtenida mediante GMM en la imagen CHB02	37
5.6. Segmentación tras eliminación de candidatos fronterizos	37
5.7. Segmentación tras aplicación de reglas de selección de lesiones	37
5.8. Segmentación del experto realizada por miembros del CHB	37
5.9. Segmentación obtenida por la CNN sin capa de <i>Max Pooling</i>	40
5.10. Segmentación obtenida por la CNN con capa de <i>Max Pooling</i>	40
5.11. Segmentación del experto realizada por miembros del CHB	40
5.12. Imagen T1 Preprocesada	42
5.13. Imagen T2 Preprocesada	42
5.14. Imagen FLAIR Preprocesada	42
5.15. WM obtenida de <i>K-Means</i>	42
5.16. Segmentación entregada por el experto del CHB	42
5.17. Segmentación obtenida a partir del GMM	42
5.18. Segmentación obtenida por CNN (sin <i>Max Pooling</i>)	42

5.19. Segmentación obtenida por CNN (con <i>Max Pooling</i>)	42
5.20. Segmentación de CNN Multivista (sin <i>Max Pooling</i>)	42
5.21. Segmentación de CNN Multivista (con <i>Max Pooling</i>)	42
5.22. Ejemplo de imagen en la cual falla la segmentación automática de WM	42
5.23. Segmentación previa al postprocesamiento	46
5.24. Segmentación tras realizado postprocesamiento	46
5.25. Segmentación del experto	46
5.26. Ejemplo de las ventajas del filtro de cierre	46

1 | Contextualización y Definición del Problema

Las imágenes de resonancia magnética (MRI) son una técnica de extracción de imágenes médicas que permite obtener imágenes de distintas partes del cuerpo, tales como el cerebro, corazón o la columna vertebral, con el fin de ayudar en el diagnóstico de diversas patologías que puedan estar afectando a un paciente. Esta técnica presenta un gran número de ventajas con respecto a otras técnicas de escaneo cerebral, entre ellas el uso de radiación no ionizante y la generación de imágenes con alto grado de contraste entre los distintos tipos de tejidos presentes en el cerebro (líquido cefalorraquídeo, materia blanca y materia gris), la cual la convierte en un excelente apoyo para diagnosticar enfermedades que afectan al sistema nervioso central [8], tales como la epilepsia y demencia.

Entre estas enfermedades se encuentra la Esclerosis Múltiple, enfermedad crónica que causa daño al sistema nervioso central, caracterizada por la aparición de regiones de inflamación y desmielinización de las neuronas, los cuales resultan en una pérdida progresiva de las capacidades motoras, sensoriales y cognitivas del paciente afectado. Mediante el uso de las MRI, es posible detectar las regiones afectadas por esta enfermedad y emitir un diagnóstico basado en el número y tamaño de las lesiones detectadas.

Esta labor es por lo general realizada por un profesional médico experto o de manera semiautomática, es decir, haciendo uso de técnicas computacionales de segmentación de imágenes, las cuales requieren de un ajuste de parámetros realizados por el experto. Este enfoque suele ser presentar dificultades y desventajas, tales como la existencia de una alta variabilidad en los resultados emitidos por distintos expertos. Debido a esto, surge el interés de la comunidad médica en desarrollar técnicas de segmentación automáticas, con el fin de poder disminuir el tiempo requerido para

poder identificar las regiones dañadas en el cerebro. Existen muchos desafíos en los problemas de segmentación automática, y su uso en la práctica es bastante limitado debido a la dificultad de validar correctamente las predicciones realizadas por estas herramientas. En esta memoria, se estudiará la efectividad de distintos métodos de segmentación para este tipo de imágenes, proponiendo el uso de Redes Neuronales Convolucionales para la segmentación de lesiones y comparándolas con métodos tradicionales de segmentación, analizando las ventajas y desventajas que cada uno de estos métodos traen en la práctica.

El objetivo de este documento consiste en presentar distintas técnicas del Estado del Arte en el área de segmentación de imágenes. Luego, se procederá a realizar la implementación de estas y su comparación en términos de desempeño y efectividad en el problema de segmentación de imágenes de resonancia magnéticas. Para realizar la validación de resultados, se hará uso del conjunto de imágenes de entrenamiento del CHB del *MS Lesion Segmentation Challenge* MICCAI 2008, el cual consiste en 10 imágenes en modalidades T1, T2 y FLAIR de distintos pacientes, acompañados de la segmentación de un experto. Debido a que las segmentaciones del conjunto de prueba no son de acceso público, para los algoritmos de aprendizaje supervisado, en este caso las Redes Neuronales Convolucionales, se dividirá el conjunto de imágenes en un subconjunto de entrenamiento, a partir del cual se entrenará el modelo de redes convolucionales, y un conjunto de prueba, con el cual se pondrá a prueba la capacidad del modelo de segmentar correctamente las lesiones en una imagen nueva. Se realizará este proceso 5 veces por técnica y se obtendrá un promedio de las medidas de rendimiento.

1.1. Objetivos

1.1.1. Objetivo General

Estudiar y evaluar de manera experimental el desempeño de las redes neuronales convolucionales aplicadas al problema de segmentación automática de Lesiones de Esclerosis Múltiple en Imágenes de Resonancia Magnética multimodales, comparándola con otras técnicas del Estado del Arte.

1.1.2. Objetivos Específicos

- Investigar el Estado del Arte de las distintos tipos de técnicas de segmentación de imágenes aplicadas a la problemática.
- Implementar dos técnicas de segmentación del Estado del Arte aplicadas al problema de detección de lesiones de Esclerosis Múltiple.
- Evaluar la efectividad de las redes neuronales comparadas con un método tradicional en el conjunto de Imágenes de Resonancia Magnética del *Multiple Sclerosis Lesion Segmentation Challenge* MICCAI 2008.

2 | Estado del Arte

2.1. Imágenes de Resonancia Magnética

Las imágenes de resonancia magnética son generadas aplicando un campo magnético de entre 1.5 y 3 Tesla (T), causando la alineación de los protones de los átomos de hidrógeno, abundantes en los átomos de agua del cuerpo humano, con la dirección del campo aplicado. Este proceso de alineación es luego interrumpido mediante el uso de ondas de baja frecuencia, produciendo que los protones vuelvan a su estado original. Este proceso se repite con determinadas frecuencias, las cuales son controladas por el operador del resonador magnético. Tras este proceso de excitación y relajación de los momentos magnéticos dentro de los átomos, se mide la energía emitida por los protones al volver a su estado original, la cual es traspasada a niveles de intensidad. Estas mediciones de energía son luego utilizadas para reconstruir una imagen de los tejidos de interés.

El operador del resonador magnético es capaz de calibrar los parámetros TR y TE . TR corresponde al *Repetition Time*, es decir, el tiempo entre aplicaciones del campo magnético de excitación, causante de la alineación de los protones con la dirección del campo. TE corresponde al *Echo Time*, es decir, el tiempo entre la aplicación del pulso de radiación de baja frecuencia que provoca que los protones vuelvan a su estado original. Mediante el manejo de estos parámetros, es posible obtener distintos tipos de imágenes de la zona afectada, entre las cuales se encuentran las *T1-Weighted*, *T2-Weighted* y *Fluid-Attenuated Inversion Recovery (FLAIR)*. Cada una de estas imágenes se caracteriza por el tipo de contraste de intensidades existente entre los distintos tipos de tejido.

2.2. Uso de MRI en el diagnóstico de Esclerosis Múltiple

La Esclerosis Múltiple se caracteriza por la aparición de lesiones principalmente ubicadas en la materia blanca del cerebro. Dependiendo del tipo de imagen, las lesiones pueden identificarse por su intensidad con respecto a los demás tejidos del cerebro [17]. En las imágenes *T1-Weighted*, el líquido cefalorraquídeo (CSF) se caracteriza por tener una baja intensidad, mientras que la materia blanca (WM) corresponde a los vóxeles de mayor intensidad, con la materia gris (GM) en un nivel intermedio entre estos. En este tipo de imágenes, las lesiones de Esclerosis Múltiple pueden ser identificadas como regiones de menor intensidad que aparecen dentro del área de la materia blanca. En las imágenes *T2-Weighted*, el líquido cefalorraquídeo adopta niveles intermedios de intensidad, mientras que la materia blanca sigue siendo identificable por su alto nivel de intensidad, siendo las lesiones identificables como regiones de hiperintensidad en la materia blanca. Por último, las imágenes FLAIR poseen características similares a las T2, con la excepción del líquido cefalorraquídeo, el cual puede ser visualizado como regiones de baja intensidad.

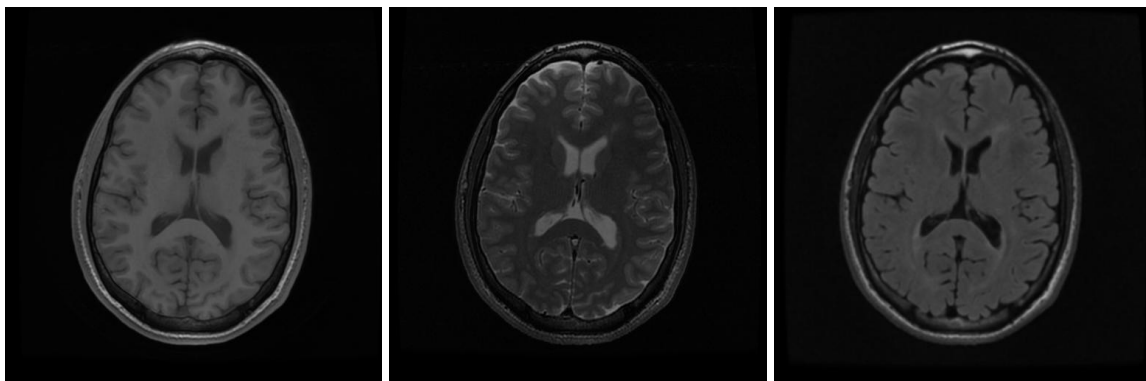


Figura 2.1: Ejemplo de Imagen T1 **Figura 2.2:** Ejemplo de Imagen T2 **Figura 2.3:** Ejemplo de Imagen T2-FLAIR

Haciendo uso de estos conocimientos, una estrategia general utilizada para el reconocimiento de lesiones en las imágenes consiste en realizar los siguientes pasos:

1. Realizar una segmentación inicial de la materia cerebral en los 3 distintos tipos principales de tejido: WM, GM y CSF. Esta tarea de segmentación preliminar suele realizarse con el apoyo de Atlas Probabilísticos, sin embargo es posible aplicar algoritmos de clustering para agrupar los vóxeles de la imagen en los 3 distintos tipos. En este caso, suele preferirse el uso de la

imagen T1, debido a que en esta los tejidos son relativamente fáciles de distinguir debido al contraste de intensidades entre cada tipo de tejido.

2. Aislar un conjunto de vóxeles, basado en la segmentación previa, como un conjunto de candidatos a lesión.
3. Aplicar algún algoritmo o conjunto de reglas para separar los falsos positivos obtenidos en el paso anterior de los vóxeles que efectivamente corresponden a lesiones de Esclerosis Múltiple. Existen diversas técnicas propuestas en la literatura para este último paso, las cuales van desde el uso de clasificadores o definición de un conjunto de reglas, propuestas generalmente con ayuda de un experto, para elegir los vóxeles que corresponden a lesiones.

2.3. Umbralización

La umbralización es el método más simple de segmentación de imágenes [14]. Este método consiste en crear una nueva imagen con un número discreto de niveles de intensidad en escala de grises, por lo general 2 en su forma más básica (identificar el *background* y *foreground* de una imagen). Sea $g(x, y)$ una imagen 2D generada al aplicar un umbral T a una imagen original $f(x, y)$, los niveles de intensidad de cada pixel de esta imagen están dados por:

$$g(x, y) = \begin{cases} 1, & f(x, y) \geq T \\ 0, & f(x, y) < T \end{cases}$$

Este tipo de umbralización tiene el problema de que considera las relaciones espaciales de entre pixeles vecinos, solo la intensidad de estos. Para solucionar este problema, pueden tomarse umbrales localmente sobre distintas regiones de la imagen.

Para determinar el o los distintos niveles de umbrales en una imagen, se toma el histograma de la imagen. El histograma de una imagen corresponde a la distribución de las intensidades de cada pixel. Esto puede verse como una función de distribución de probabilidad:

$$P(g) = \frac{n_g}{n}$$

Donde g un nivel de intensidad en escala de grises, n_g es el número de píxeles de la imagen que posee intensidad g y n es el total de píxeles de la imagen. Existen distintos métodos para determinar el nivel de umbral apropiado para una imagen.

2.3.1. Distribución Conocida

Si se conoce aproximadamente la proporción entre el número de píxeles que corresponden al elemento que se desea segmentar (*foreground*) y el tamaño de la imagen, basta con elegir un umbral tal que la suma acumulada de frecuencias en el histograma sea cercana a esta proporción.

2.3.2. Picos y Valles

Dado el histograma de la imagen, se deben hallar los máximos de este. Una vez hallados los máximos, se buscan los mínimos locales entre estos y se eligen como umbrales las sumas acumuladas de frecuencias en el histograma para cada uno de estos mínimos. Esta técnica tiene la desventaja de ser muy sensible al ruido, por lo cual se debe aplicar alguna técnica que permita suavizar el histograma previamente.

2.3.3. Clustering (Otsu)

Otra manera de ver el problema de segmentación es asumir que la distribución de elementos del fondo de la imagen y del objeto a segmentar se traslapan en el histograma de intensidades, habiendo valores de intensidad que aparecen en ambos elementos. Debido a esto, el objetivo de la segmentación debe consistir en hallar el nivel de umbral que minimice el traslape de la distribución de intensidades del fondo y del objeto.

Se define la varianza intra-clases $\sigma_{\text{intra}}^2(T)$ como la suma ponderada de las varianzas de cada clase, en este caso el *foreground* y *background*, dado un umbral T como:

$$\sigma_{\text{intra}}^2(T) = \frac{n_B(T)\sigma_B^2(T) + n_O(T)\sigma_O^2(T)}{n_B(T) + n_O(T)}$$

Donde $n_B(T)$ es el número de píxeles del fondo (*background*), cuya varianza de intensidades es $\sigma_B^2(T)$, y $n_O(T)$ es el número de píxeles del objeto a segmentar, cuya varianza de intensidades es

$\sigma_o^2(T)$.

El método de Otsu consiste en hallar el valor de intensidad T para el cual la varianza intra-clases es mínima. Alternativamente, se puede buscar maximizar la varianza inter-clases, la cual se calcula de la forma:

$$\sigma_{\text{inter}}^2(T) = n_B(T)n_O(T)\left(\frac{\mu_B(T) - \mu_O(T)}{n_B(T) + n_O(T)}\right)^2$$

Esta forma depende de las medias del fondo $\mu_B(T)$ y del objeto a segmentar $\mu_O(T)$, las cuales son mucho más fáciles de calcular computacionalmente.

2.3.4. Umbralización en Segmentación de Lesiones de Esclerosis Múltiple

Las lesiones de esclerosis múltiple se presentan como regiones de hiperintensidad en las imágenes T2 y FLAIR. Debido a esto, una multitud de métodos en el Estado del Arte hace uso de técnicas de umbralización para seleccionar un subconjunto de vóxeles de la imagen como candidatos a lesión. En las siguientes secciones se discutirá en mayor profundidad aquellos métodos que hacen uso de umbralización durante el proceso de segmentación.

2.4. Métodos basados en Atlas Probabilísticos

Un Atlas Cerebral consiste en una recolección de las distintas estructuras que componen el cerebro hecha por un experto, la cual describe la anatomía de un cierto grupo de individuos. Esta puede ser realizada mediante la segmentación de los distintos tejidos que componen el cerebro, por ejemplo la materia blanca. Estas herramientas permiten tener información a priori que permite facilitar el proceso de segmentación de imágenes. Existen 2 tipos de atlas utilizados: los Atlas Topológicos y los Atlas Probabilísticos. Un Atlas Topológico corresponde a una imagen volumétrica seleccionada para representar a un grupo de imágenes como una especie de retrato promedio. A diferencia de estos, los Atlas Probabilísticos son construcciones que permiten describir la distribución de tejidos que componen el cerebro, los cuales son construidos en base a múltiples imágenes, lo cual permite representar la variabilidad entre distintos individuos [13].

Se define un conjunto de imágenes 3D $I(x)$ como un mapa $I : x \in \mathbb{R}^3 \rightarrow I(x) \in \mathbb{R}^N$, donde x representa las coordenadas de los vóxeles en una imagen 3D y N representa el número de

modalidades de imagen obtenidas de una MRI multi-espectral (por ejemplo, en el caso que se trabajará en esta memoria se tiene trabajó con las modalidades T1, T2 y T2-FLAIR, por lo tanto $N = 3$). Las coordenadas espaciales para estas imágenes se encuentran inicialmente en el espacio \mathcal{X}_I . Además, se definen los volúmenes $\pi^I(x)$, $\pi^L(x)$ y $\pi_c^P(x)$, donde:

- $\pi^I(x) : x \in \mathbb{R}^3 \rightarrow \pi^I(x) \in \mathbb{R}$, la cual corresponde a la imagen en escala de grises a la cual se le aplicará el Atlas.
- $\pi^L(x) : x \in \mathbb{R}^3 \rightarrow \pi^L(x) \in \mathcal{L}$, la cual corresponde a la imagen del Atlas que contiene las etiquetas pertenecientes al conjunto $\mathcal{L} = \{1, 2, \dots, C\}$, donde C es el número de etiquetas que se aplicará a la imagen. Por ejemplo, si interesa obtener una segmentación para los 3 tipos de tejido cerebral, cada una de las etiquetas de este volumen corresponderán a WM, GM y CSF respectivamente.
- $\pi_c^P(x) : x \in \mathbb{R}^3 \rightarrow \pi_c^P(x) \in \mathbb{R}$, los cuales corresponden a los volúmenes probabilísticos para cada etiqueta c con $c \in \mathcal{L}$, donde cada vóxel del volumen contiene la probabilidad de que este pertenezca a la clase c .

Las coordenadas espaciales para estos volúmenes se encuentran en el espacio del Atlas \mathcal{X}_π , el cual no necesariamente coincide con el espacio de la imagen \mathcal{X}_I . Esto se debe a que las imágenes obtenidas a través del proceso de resonancia magnética pueden encontrarse en resoluciones distintas a la imagen de Atlas con el que se trabajará. Debido a esto, es necesario aplicar una transformación que permita adecuar el tamaño de los volúmenes del Atlas para que coincidan con las imágenes de resonancia magnética, de manera que cada uno de los vóxeles de las imágenes originales correspondan espacialmente a aquellos del Atlas. Se define $\tau : x_\pi \in \mathcal{X}_\pi \rightarrow x_I \in \mathcal{X}_I$ como la transformación que permite llevar los volúmenes del Atlas al mismo espacio que las imágenes de resonancia magnética. El proceso de hallar esta transformación se conoce como *Image Registration* y consiste en el problema de optimización:

$$\hat{\tau} = \arg \max_{t \in \mathbb{T}} \delta(I(x), \pi^I(\hat{\tau}(x)))$$

Donde \mathbb{T} es el espacio de todas las transformaciones, δ es una medida de similaridad que permita comparar las imágenes y $\hat{\tau}$ es la transformación óptima de los volúmenes del Atlas al espacio de las

imágenes. Un ejemplo de medidas de similaridad que suelen usarse en la práctica son las medidas derivadas de la entropía conjunta. Se define la entropía de un conjunto de valores $X = \{x_i\}$, con $i \in 1, \dots, N$ como:

$$H(X) = \sum_{i=1}^N -p(x_i) \log(p(x_i)) \quad (2.1)$$

Donde $p(x_i)$ es la probabilidad de observar el valor x_i en el conjunto de observaciones. En el caso de las imágenes, estos valores se obtienen a partir del histograma de intensidades normalizado. La entropía conjunta para 2 muestras $A = \{a_i\}$ y $B = \{b_i\}$ se define como:

$$H(A, B) = \sum_{i=1}^N -p(a_i, b_i) \log(p(a_i, b_i)) \quad (2.2)$$

Para este caso $p(a_i, b_i)$ se obtiene para cada una de las combinaciones de valores de intensidad en el histograma. Otras métricas derivadas a partir de esta son la información mutua dada por: $MI(A, B) = H(A, B) - H(A) - H(B)$ y la información mutua normalizada $NMI(A, B) = \frac{H(A) + H(B)}{H(A, B)}$.

Finalmente, la segmentación $S(x)$ corresponde a asignar a cada voxel de la imagen una etiqueta. Esto es $S : x \in \mathbb{R}^3 \rightarrow S(x) \in \mathcal{L}$.

Una vez hallada la transformación, se procede a segmentar la imagen transformada en base a las etiquetas del Atlas realizando:

$$S(x) = \pi^L(\hat{\tau}(x))$$

Esta estrategia se conoce como *Label Propagation*. El problema de esta segmentación consiste en que su eficacia depende de la similitud entre las anatomías del paciente y de la imagen del Atlas. Si existe mucha diferencia, la segmentación puede ser errónea.

Otra estrategia que permite tomar en cuenta la variabilidad en la anatomía de cada individuo consiste en el *Multi-Atlas Label Propagation*. En este método, se seleccionan P Atlas y se realiza la segmentación mediante una combinación ponderada de sus etiquetas de la forma:

$$S(x) = \arg \max_c \sum_{i=1}^P w_i(x) \cdot f(\pi_i^L(\hat{\tau}_i(x)), c)$$

Donde $f(\pi_i^L(\hat{\tau}_i(x)), c)$ es una función que es 1 si el voxel posee la etiqueta c en el i -ésimo Atlas y 0 en el caso contrario. Los pesos $w_i(x)$ pueden seleccionarse de acuerdo a una variedad de criterios.

Por último, la segmentación en Atlas Probabilísticos se realiza mediante las reglas Bayesianas de probabilidad, donde se busca maximizar:

$$S(x) = \arg \max_c P(I(x) | c) \cdot P(c)$$

Donde $P(I(x) | c)$ es la probabilidad condicional de la intensidad dada una etiqueta c y $P(c)$ es la probabilidad a priori de la clase c .

2.5. Gaussian Mixture Models

Un Modelo de Mezcla de Gaussianas, conocido también por sus siglas en inglés GMM (Gaussian Mixture Model), es un modelo probabilístico que se crea bajo el supuesto de que todos los patrones, o vectores de características, de un conjunto de datos provienen de una mezcla de un número finito de distribuciones Gaussianas con parámetros desconocidos. Un GMM está dado por la ecuación:

$$f(x) = \sum_{i=1}^K \alpha_i f_i(x | \mu_i, \Sigma_i) \quad (2.3)$$

Donde:

f = función de densidad de la mezcla de gaussianas

x = vector de características

K = número de gaussianas que conforman el modelo

f_i = función de densidad de la i -ésima gaussiana

α_i = coeficiente de peso de la i -ésima gaussiana

μ_i = vector de medias de la i -ésima gaussiana

Σ_i = matriz de varianzas y covarianzas de la i -ésima gaussiana

Esta técnica posee aplicaciones como algoritmo de aprendizaje no supervisado, permitiendo agrupar datos basándose en la probabilidad a posteriori de que un vector de características pertenezca a cada gaussiana dentro del modelo. Dada la ecuación (2.3) para un GMM, la etiqueta asignada al vector de características x corresponde a la gaussiana que maximiza la probabilidad a posteriori de

que dicho punto pertenezca esta, es decir:

$$\text{etiqueta}(x) = \arg \max_i \alpha_i f_i(x | \mu_i, \Sigma_i) , i \in \{1, \dots, K\} \quad (2.4)$$

Un GMM aplicado a las MRI cerebrales consiste en modelar la intensidad de los vóxeles de la imagen como una variable aleatoria que proviene de una mezcla de gaussianas, donde cada gaussiana representa un distinto tipo de tejido cerebral, caracterizado por sus respectiva media y varianza de intensidad. Dentro de la literatura, se encontraron 2 enfoques de este algoritmo aplicados al problema de segmentación de lesiones de Esclerosis Múltiple. El primero modela los distintos tipos de tejido cerebral y se enfoca en la búsqueda de *outliers* dentro del modelo para hallar las lesiones, mientras que el segundo hace uso de la información espacial de la imagen mediante el uso de múltiples gaussianas y luego realiza la búsqueda de lesiones a nivel de las gaussianas obtenidas en el modelo.

2.5.1. Detección de Lesiones como Outliers en un GMM

El segundo enfoque consiste en el modelamiento del tejido cerebral mediante el uso de un GMM para las intensidades de los vóxeles a lo largo de 3 distintos tipos de modalidades de imagen (T1, T2 y T2-FLAIR). Sean $x_{T1}(i)$, $x_{T2}(i)$ y $x_{T2-F}(i)$ las intensidades del vóxel en la posición i en las imágenes T1, T2 y FLAIR registradas para un mismo paciente respectivamente, el modelo propuesto en [10] toma los vectores $(x_{T1}(i), x_{T2}(i), x_{T2-F}(i))$ como entrada y modela su distribución mediante un GMM de 3 componentes, correspondientes a la materia blanca, materia gris y líquido cefalorraquídeo. El modelo asume independencia entre las intensidades de las 3 distintas imágenes, por lo cual las covarianzas entre intensidades de distintas modalidades de imagen son nulas y el modelo está compuesto por 21 parámetros: $\alpha^{(T)}$, $\mu_I^{(T)}$ y $\sigma_I^{2(T)}$, donde I es un índice que representa el tipo de imagen (T1, T2 y FLAIR) y T el índice que representa el tipo de tejido que modela la gaussiana (WM, GM o CSF).

2.5.1.1. Estimación de Parámetros

La estimación de parámetros del GMM se realiza aplicando el algoritmo de *Expectation Maximization* (EM). El algoritmo EM es de carácter iterativo, por lo cual se toman los parámetros de la

iteración anterior y se recalculan estos hasta llegar a una condición de término. Sean x_j el vector de intensidades para cada vóxel de las imágenes, μ_i el vector de medias de intensidad y Σ_i la matriz diagonal de varianzas de intensidad, la función de verosimilitud del GMM para un conjunto de imágenes de resonancia magnética de N vóxeles con K modalidades está dada por:

$$\prod_{j=1}^N \sum_{i=1}^K \alpha_i f_i(x_j | \mu_i, \Sigma_i) \quad (2.5)$$

Cada iteración del algoritmo se divide en 2 pasos:

- *E-Step*: durante este paso, se calcula la probabilidad a posteriori de observar las características de cada vóxel para cada una de las gaussianas dados los parámetros actuales. Para cada vóxel de la imagen se calcula:

$$w_{ij} = \frac{\alpha_i f_i(x_j | \mu_i, \Sigma_i)}{\sum_{i=1}^K \alpha_i f_i(x_j | \mu_i, \Sigma_i)}, \quad \forall i \in \{1, \dots, K\}, j \in \{1, \dots, N\} \quad (2.6)$$

Donde w_{ij} es la probabilidad de que el vóxel j pertenezca al tejido i .

- *M-Step*: durante este paso, utilizando las probabilidades calculadas en el paso anterior, se calculan nuevamente los parámetros del modelo para cada gaussiana. Sean:

$$n_i = \sum_{j=1}^N w_{ij}, \quad \forall i \in \{1, \dots, K\} \quad (2.7)$$

Donde n_i es el número esperado de vóxeles que pertenecen a la gaussiana i , los parámetros del modelo se actualizan de la siguiente manera:

$$\mu_i = \frac{1}{n_i} \sum_{j=1}^N w_{ij} x_j \quad (2.8)$$

$$\Sigma_i = \frac{1}{n_i} \sum_{j=1}^N w_{ij} (x_j - \mu_i)(x_j - \mu_i)^T \quad (2.9)$$

$$\alpha_i = \frac{n_i}{N} \quad (2.10)$$

La condición de término del algoritmo varía dependiendo de la implementación, sin embargo

generalmente se hace uso de un parámetro de tolerancia, el cual se aplica a la diferencia entre la función de log-verosimilitud de la iteración actual y la anterior. Si la diferencia es menor a la tolerancia, se concluye que el modelo ha convergido.

2.5.1.2. Inicialización Jerárquica

Para que el modelo sea robusto ante selección inicial de parámetros, se propone el uso de GMM con inicialización jerárquica con 100 componentes para segmentar la imagen T1 en los 3 tipos de tejido principales. Los parámetros se inicializan como valores aleatorios en el rango de intensidades de la imagen, obteniendo finalmente los 6 parámetros iniciales correspondientes a la imagen T1. De esta segmentación además se pueden obtener las probabilidades a priori de pertenencia a cada tipo de tejido $\alpha^{(T)}$.

Una vez obtenida la segmentación inicial, se obtiene una máscara para los 3 distintos tipos de tejido cerebral y se aplica esta a las imágenes T2 y FLAIR. Para obtener los parámetros iniciales en estas imágenes, se crea un histograma de intensidades para cada tipo de tejido en cada imagen. A este histograma se le aplica un suavizado gaussiano utilizando como parámetros la media y varianza del histograma original. Finalmente, la media para cada par tejido/imagen se estima como la moda de su histograma correspondiente y la varianza se estima como:

$$\sigma_I^{2(T)} = \left(1,4918 \text{ med}(|x_I - \mu_I^{(T)}|)\right)^2$$

Donde *med* es la mediana, x_I son las intensidades de la imagen I (T2 o FLAIR).

2.5.1.3. Verosimilitud Truncada

Debido a que interesa obtener un modelo robusto ante la presencia de *outliers*, se hizo uso de la función de verosimilitud truncada durante el proceso de *Expectation Maximization* (EM) para estimar los parámetros del modelo. La verosimilitud truncada se define como:

$$TL(\theta) = \prod_{i=1}^{n-\lfloor n/h \rfloor} f(x_{v(i)}|\theta)$$

Donde θ son los parámetros del modelo, h es un factor que indica que porcentaje de los datos de entrenamiento se descartarán durante el cálculo de la verosimilitud y $v(i)$ es una función que ordena los datos x_i , de manera que:

$$f(x_{v(1)}|\theta) > f(x_{v(2)}|\theta) > f(x_{v(3)}|\theta) > \dots > f(x_{v(n)}|\theta)$$

En otras palabras, la verosimilitud se calcula solamente utilizando aquellos puntos que tengan un mayor aporte a esta función. Los valores que no se utilizan se consideran como el conjunto de *outliers* del modelo. Los parámetros del GMM se calculan en cada iteración descartando este conjunto. El algoritmo EM se detiene una vez que el conjunto de *outliers* no cambia o que se llega a un límite de iteraciones.

2.5.1.4. Selección de Lesiones a partir de Outliers

Una vez obtenido el GMM y el último conjunto de *outliers* en el paso anterior se debe diferenciar entre los falsos positivos y aquellos vóxeles que efectivamente corresponden a lesiones. Para esto, se aplican las siguientes reglas para obtener el conjunto final de lesiones de esclerosis múltiple:

- Sea d_i la distancia de Mahalanobis mínima entre las intensidades de cada vóxel x_i y las gaussianas correspondientes a cada tipo de tejido dada por:

$$d_i = \min_{\forall j} \left\{ \sqrt{(x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j)} \right\}$$

Donde μ_j el vector de medias para el tejido j y Σ_j es la matriz de covarianzas para el tejido j . Cabe recordar que debido al supuesto de independencia, Σ_j es una matriz diagonal, cuyos elementos son las varianzas de intensidades para cada tipo de imagen, por lo tanto su inversa es simplemente una matriz diagonal cuyos elementos corresponden a la inversa de cada varianza. Los vóxeles que superen un cierto umbral para d_i son considerados candidatos a lesión.

- Debido a que las lesiones se presentan como regiones hipertensas en la materia blanca en las imágenes T2 y FLAIR, se aplica umbralización a los vóxeles candidatos para ambos tipos de imagen. El umbral para cada imagen es un parámetro que debe fijarse.

- Se aplica un algoritmo de componentes conexas a los vóxeles candidatos y se descartan aquellas regiones estén compuestas por menos de 6 vóxeles. Estas regiones se consideran como ruido en el modelo.
- Se eliminan aquellos vóxeles que no estén conectados a vóxeles de materia blanca en la segmentación obtenida a partir del GMM. Las etiquetas de cada vóxel se obtienen mediante la probabilidad a posteriori de pertenencia a cada tipo de tejido.
- Se descartan los vóxeles pertenecientes a las regiones conexas que no contengan al menos 2 vóxeles a lo largo de cada eje. Esto se realiza debido a que las lesiones suelen presentarse como regiones tridimensionales, por lo tanto deben aparecer en más de una *slice* a lo largo de cada eje en la imagen.
- Se descartan los vóxeles pertenecientes a las regiones que no tengan una forma circular en cada *slice*. Para determinar el nivel de circularidad, se calcula la varianza a lo largo de los 2 ejes de las posiciones de los vóxeles en cada *slice* y se calcula el cociente entre la mayor y la menor. Si este cociente supera un umbral, la región se descarta.

2.6. Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales (abreviadas como CNN por su nombre en inglés *Convolutional Neural Networks*) corresponden a una familia de redes neuronales artificiales multicapa o profundas (*Deep Neural Network*).

Una red neuronal consiste en una serie de capas formadas por múltiples nodos denominados neuronas. Las capas intermedias de una red se denominan como capas ocultas. Sea K el número de neuronas de la primera capa oculta de una red feedforward, cada neurona de la primera capa oculta recibe como entrada un vector de características $\vec{x} = (x_1, x_2, \dots, x_n)$ y genera como salida un valor calculado como:

$$q_k^{(1)} = w_{0k}^{(1)} + \sum_{i=1}^n w_{ik}^{(1)} x_i \quad \forall k \in \{1, 2, \dots, K\} \quad (2.11)$$

Donde $q_k^{(1)}$ es la salida de la k -ésima neurona de la primera capa, $w_{ik}^{(1)}$ es el peso que conecta la i -ésima componente del vector de características con la k -ésima neurona en la primera capa y w_{0k} es

el peso asociado a lo que se conoce como *bias* de la red. A esta salida suele aplicarse lo que se conoce como una función de activación, la cual se encarga de transformar la salida $q_k^{(1)}$ a un valor acotado acorde a la aplicación. Ejemplos de funciones de activación comúnmente utilizadas son la función sigmoideal y tangente hiperbólica, las cuales permiten transformar las salidas al intervalo $[0,1]$ y $[-1,1]$ respectivamente, y la función softmax, la cual se utiliza para escalar las salidas de la red a valores de probabilidad cuya suma total sea 1. Este proceso puede repetirse para una siguiente capa oculta, utilizando las salidas de las neuronas de la capa anterior como entradas de las neuronas de la siguiente capa, en la cual se repite el proceso. Una red que posee múltiples capas ocultas se denomina red neuronal profunda (DNN).

La capa final de una red neuronal, conocida como capa de salida, varía dependiendo del tipo de tarea que se quiera realizar con estas. Para una tarea de regresión, la capa de salida consiste en una neurona, a cuya salida no suele aplicarse una función de activación ya que se interpreta este valor como la predicción de la neurona para el vector de entrada. Para una tarea de clasificación con N clases, la capa de salida consiste en N neuronas, a las cuales se le aplica la función softmax, obteniéndose las probabilidades condicionales de pertenencia a la clase k de la forma:

$$p(k|\vec{x}; W) = \frac{e^{f(q_k^{(o)})}}{\sum_{d=0}^N e^{f(q_d^{(o)})}} \quad (2.12)$$

Donde W representa los pesos de la red y $q_k^{(o)}$ corresponde a la salida de la neurona k -ésima de la capa de salida. La clase que la red predice se interpreta como la neurona de la capa de salida que maximiza la función softmax entre las N clases.

Los pesos de la red se ajustan mediante el uso de diversos algoritmos de aprendizaje, de los cuales el algoritmo más común es el algoritmo de *backpropagation*, mediante el cual se calculan las derivadas de la función de pérdida con respecto a los pesos de la red, para finalmente actualizar estos mediante el uso de gradiente descendente.

En los problemas de procesamiento de imágenes, las redes feedforward descritas anteriormente poseen el problema de que no logran captar información de las vecindades de los píxeles, debido a que la vectorización de la imagen elimina la estructura y relación espacial entre los píxeles. Otro problema es la cantidad de parámetros que son necesarios para entrenar una red utilizando la imagen vectorizada como entrada. Debido a que las conexiones entre neuronas en estas redes son completas

entre una capa y la siguiente, si se tiene una imagen de $A \times B$ dimensiones, el número de parámetros a entrenar para una primera capa oculta compuesta de N neuronas sería de $A \times B \times N$ pesos. Las redes neuronales convolucionales resuelven estos problemas al hacer uso de la información local de cada pixel al recibir como entrada "parches" de la imagen centrados alrededor de cada pixel, los cuales son procesados dentro de la red por una serie de capas, las cuales se encargan de extraer y muestrear de manera automática características de la imagen.

La arquitectura de una CNN está compuesta por múltiples tipos de capas que cumplen distintos tipos de funciones. Algunos de los tipos comunes de capas son:

1. Capa Convolutiva: la capa convolutiva es la encargada de recibir los parches de entrada de la imagen y realizar la operación de convolución a estos. Para esto, se aplica un filtro de convolución, también conocido como kernel, al parche de entrada. Considere una imagen I de 32×32 y una capa de convolución que aplica un kernel de convolución de 5×5 , la salida de la capa de convolución, conocida como mapa de características, será un arreglo C de 28×28 , cuyos valores son calculados como:

$$C_{xy} = \sum_{i=1}^5 \sum_{j=1}^5 w_{ij} I(x+i-1, y+j-1), \quad \forall x, y \in \{1, \dots, 28\}$$

Donde, w_{ij} es la componente (i, j) del filtro de convolución e $I(x, y)$ es el valor de intensidad del pixel (x, y) en la imagen I .

En general, en cada capa de convolución es posible aplicar cualquier número de filtros, obteniendo un mapa de características para cada uno. Además, es posible recibir múltiples imágenes de entrada, las cuales deben ser procesadas por filtros de convolución tridimensionales, donde la profundidad de estos filtros debe coincidir con el número de imágenes de entrada.

Las funciones de activación que suelen utilizarse en estas capas consisten en funciones lineales, tales como la función rectificadora lineal uniforme (ReLU) definida como:

$$f(x) = \max(0, x)$$

La razón del uso de estas funciones es que evitan el problema de desvanecimiento del gradiente.

Debido a que la arquitectura de las redes convolucionales suele estar compuesta por múltiples capas de convolución, durante el algoritmo de *backpropagation*, la derivada de la función de pérdida con respecto a los pesos de las capas más cercanas a la entrada de la red depende del producto de las derivadas de las siguientes capas. Esto se debe a que las derivadas se calculan mediante la regla de la cadena, por lo cual la capa de salida depende únicamente de su derivada, la capa oculta que precede a esta depende de su derivada y la de la capa de salida, y así sucesivamente. Al usar funciones tales como la sigmoideal y tangente hiperbólica, la derivada se encuentra acotada en el rango $] - 1, 1[$, lo cual se puede observar en la figura 2.5, lo cual resulta en que las derivadas se van haciendo cada vez más pequeñas a medida que uno se acerca a las capas más superficiales de la red. El uso de una función lineal permite mitigar este problema en las redes profundas, ya que su derivada es constante por intervalos.

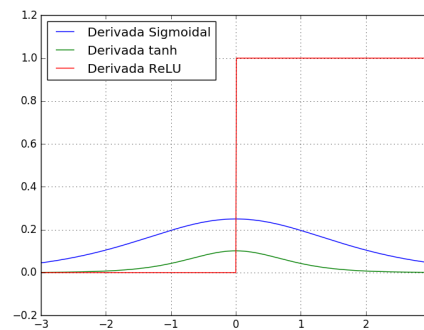
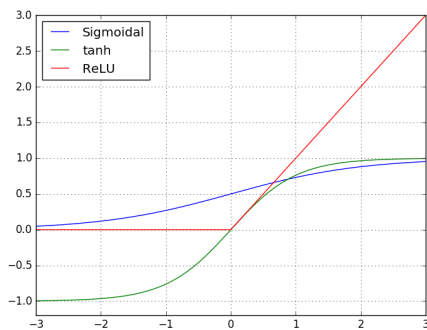


Figura 2.4: Gráficas de las funciones de activación comúnmente utilizadas. **Figura 2.5:** Derivadas de las funciones de activación.

2. Capa de Pooling: la capa de *pooling*, también conocida como capa de submuestreo, es una capa opcional que se aplica a los mapas de características obtenidos a partir de la capa convolucional. Esta capa aplica, mediante una ventana deslizante, una función que resume la información más importante del mapa de características. Ejemplos de funciones que suelen ser aplicadas son el *Max Pooling*, el cual consiste en tomar el mayor elemento de la ventana, y el promedio de los elementos de la ventana.

La aplicación de la capa de *pooling* trae múltiples ventajas, dentro de las cuales se encuentran:

- Reducción de la dimensionalidad de los mapas de características, lo cual resulta en una disminución de los parámetros a entrenar en la red, además de ayudar a controlar el

- sobreajuste.
- Aumenta la invarianza ante pequeñas transformaciones, tales como translaciones y rotaciones, al representar múltiples parches locales bajo un valor promedio o máximo.
3. Capa de Dropout: la capa de *dropout* es una capa opcional que puede aplicarse posterior a la mayoría de las capas de la red. Dado un parámetro p , la capa de *dropout* se encarga de "desactivar", con probabilidad p de éxito, cada una de las neuronas de la capa anterior. Esto resulta en que, en promedio, solo una fracción p de las neuronas sea utilizada durante cada etapa de entrenamiento. A pesar de que a veces se refiere a esta como una capa adicional en la literatura, en estricto rigor no lo es, ya que la regularización de parámetros ocurre a nivel de las conexiones entre capas.
- El objetivo principal de esta capa, consiste en reducir el sobreajuste del modelo al utilizar una menor cantidad de parámetros que el total de la red durante la etapa de entrenamiento, lo cual además resulta en menores tiempos de entrenamiento.
4. Capa Densa: dado que la salida entregada por las capas de convolución y *pooling* consiste en un mapa de características, el cual puede interpretarse como la aplicación de uno o múltiples filtros a la imagen de entrada, en la práctica, estos mapas suelen utilizarse como entrada para otras redes neuronales. Una capa densa consiste en una red *feedforward*, tal y como las descritas en la sección anterior, cuyo vector de entradas corresponde a la vectorización del mapa de características obtenido en las capas previas.

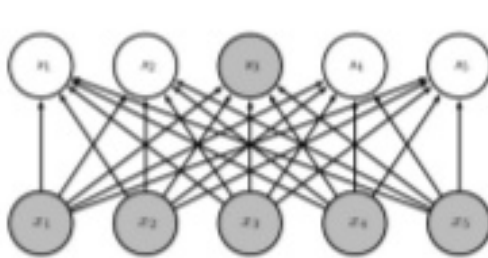


Figura 2.6: Conexiones en una Feedforward Neural Network.

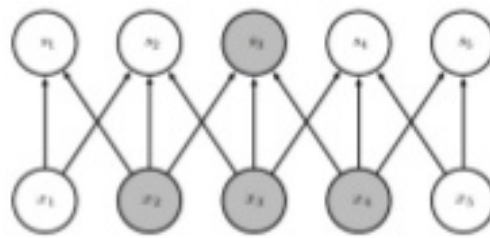


Figura 2.7: Conexiones en una Convolutional Neural Network (kernel 3×3 , vista 1D).

Figura 2.8: Comparación de las conexiones entre neuronas de una FNN y CNN

Estas redes se entrenan a lo largo de múltiples *epochs*. Durante cada *epoch*, la red recibe los datos de entrada y aplica el algoritmo de entrenamiento para ajustar sus pesos. Al final de cada *epoch*, se evalúa el desempeño de la red y se detiene el proceso de aprendizaje si es que se llega a la condición de convergencia. Por lo general, las condiciones de término del entrenamiento consisten en la observación de cambios insignificantes en el error de predicción de la red para los datos de entrenamiento o en la fijación de un número máximo de iteraciones.

Las redes neuronales convolucionales han sido aplicadas en la literatura al problema de segmentación de lesiones de esclerosis múltiple. Debido al problema de desbalance de clases en el caso de segmentación de esclerosis, correspondiendo las lesiones a un porcentaje muy pequeño de los vóxeles de la imagen, generalmente en la literatura se trabaja utilizando como entradas de la red un subconjunto de vóxeles de la imagen, alrededor de los cuales se obtiene un parche centrado en dicho vóxel.

2.6.1. Muestreo de Parches no Uniformes

El método propuesto en el paper de Ghafoorian [12] para realizar segmentación de regiones de hiperintensidad en imágenes FLAIR, realiza un muestreo no uniforme de los vóxeles de la imagen. Sea $P_{i,j,k}$ un parche de $n \times n$ que se desea añadir a la muestra no uniforme para representar la vecindad un vóxel (i,j,k) , se tiene:

$$P_{i,j,k}\left(a + \left\lfloor \frac{n}{2} \right\rfloor, b + \left\lfloor \frac{n}{2} \right\rfloor\right) = I(i + l, j + m, k)$$

Donde a y b son enteros en el intervalo $\left\{ -\left\lfloor \frac{n}{2} \right\rfloor, \left\lfloor \frac{n}{2} \right\rfloor \right\}$, correspondientes a un desplazamiento con respecto al centro del parche, y l y m son enteros correspondientes a un desplazamiento con respecto al centro de la imagen con respecto a la cual se esta muestreando, dados por:

$$l = \left\lfloor a \cdot e^{\alpha \sqrt{a+b}} + \frac{1}{2} \right\rfloor$$

$$m = \left\lfloor b \cdot e^{\alpha \sqrt{a+b}} + \frac{1}{2} \right\rfloor$$

Donde α es un parámetro que controla la distribución de los parches, donde $\alpha = 0$ representa una distribución uniforme de parches. Este método de muestreo resulta en una distribución de vóxeles de mayor densidad cerca del centro del parche, la cual disminuye a medida que uno se aleja del centro del parche. Las imágenes resultantes de este proceso de muestreo se utilizan como entrada para la red. El tamaño utilizado para los parches es de $n = 32$, y el valor de α fue variado entre 0.01, 0.02 y 0.04.

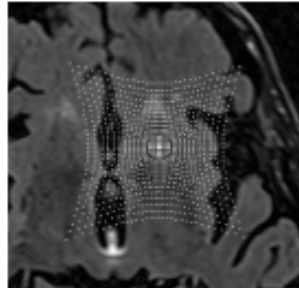


Figura 2.9: Muestreo de píxeles centrales propuesto [12]

La arquitectura de la red convolucional que aplican consiste de 4 capas convolucionales de 20 filtros de 7×7 , 40 filtros de 5×5 , 80 filtros de 3×3 y 110 filtros de 3×3 ; 3 capas densas de 300, 200 y 2 neuronas respectivamente; y finalmente una capa de activación *softmax* para cada una de las 2 neuronas de salida de la capa densa. La función de activación para cada capa fue la función ReLU y no se hizo uso de capas de *Pooling* intermedias, debido a que el problema de segmentación de lesiones es sensible a pequeñas traslaciones. Se utilizó el algoritmo RMSPROP para el entrenamiento de las redes y se añadió una capa de dropout con parámetro 0.3 para regularizar los pesos de la red. Las redes fueron entrenadas por 10 *epochs*.

Esta arquitectura fue entrenada para 3 distintos tipos de entrada:

- *Single-scale*: se utilizan solo parches de 32×32 como entrada.
- *Multi-scale Early Fusion*: se utilizan como entrada para cada vóxel un parche de 32×32 , parches de 32×32 muestreados a partir de un parche de 64×64 centrado alrededor de dicho vóxel y parches de 32×32 muestreados a partir de un parche de 128×128 centrado en dicho vóxel. Finalmente, se obtiene una entrada con múltiples canales.

- *Multi-scale Late Fusion*: se entrenan independiente 3 copias de la misma serie de capas convolucionales descritas en la arquitectura de la red, utilizando los 3 tipos de parches descritos en la entrada anterior para cada una. Los mapas de salida de cada una de estas se pasan a una capa densa de 300 neuronas independientemente y, finalmente, estas se combinan en una capa de 200 neuronas, seguida por la capa de salida de 2 neuronas con la aplicación de la capa *softmax*.

2.6.2. Redes Longitudinales Multicanal

El método propuesto en Birenbaum [4] hace uso de una etapa de selección de candidatos a lesión para determinar aquellos vóxeles alrededor de los cuales se obtendrán los parches de la imagen que serán utilizados como entrada de la red. Los criterios de selección de candidatos son los siguientes:

- Realizar umbralización de la imagen FLAIR con parámetro θ_{FLAIR} , obteniendo una máscara de los vóxeles de mayor intensidad de la imagen original.
- Debido a que las lesiones suelen presentarse en la frontera entre materia blanca y gris, se registra un Atlas probabilístico de la materia blanca a la imagen para conseguir una segmentación de esta. Luego, se realiza un filtro de dilatación con parámetro R a la máscara de la materia blanca obtenida para obtener una segmentación más robusta de la materia blanca. Finalmente, se aplica umbralización a la imagen obtenida con parámetro θ_{WM} para obtener la máscara de candidatos en la materia blanca.

La máscara final corresponde a la intersección de ambas máscaras obtenidas. Los parámetros θ_{FLAIR} , θ_{WM} y R se eligen mediante Cross Validation. Según los resultados reportados en el paper, la máscara final obtenida con la mejor combinación de parámetros corresponde aproximadamente a 2,5 % del tejido cerebral, lo cual disminuye significativamente el tiempo de entrenamiento de la red si se compara con el uso de todos los vóxeles de la imagen para entrenar el modelo.

Se experimentó con 4 distintas arquitecturas de red. Los parches de entrada consisten en parches de 32×32 centrados en cada uno de los vóxeles candidatos a lesión, tomando un parche con vista axial, coronal y sagital. Las 4 arquitecturas son las siguientes:

- *Single Image, Single Time Point (SISTP)*: se extraen parches de la imagen FLAIR en un instante de tiempo.

- *Multiple Images, Single Time Point (MISTP)*: se extraen parches para 4 modalidades de la imagen (T1, T2, PD-Weighted y FLAIR) en un instante de tiempo.
- *Single Image, Multiple Time Points (SIMTP)*: se extraen parches de la imagen FLAIR para el tiempo actual y el tiempo anterior.
- *Multiple Images, Multiple Time Points (MIMTP)*: se extraen parches para 4 modalidades de la imagen (T1, T2, PD-Weighted y FLAIR) para el tiempo actual y el tiempo anterior.

Cada una de estas arquitecturas está compuesta por 2 tipos de bloques, denominados *V-Net* (*Single View CNN*) y *L-Net* (*Longitudinal Network*). El bloque *V-Net* está compuesto por 3 capas convolucionales de 24 filtros de 5×5 , 32 filtros de 3×3 y 48 filtros de 3×3 . Las primeras 2 capas convolucionales son seguidas de una capa de *Max Pooling* con un kernel de 2×2 que se aplica cada 2 unidades, efectivamente dividiendo a la mitad el número de mapas extraído en las primeras 2 capas. El bloque *L-Net* procesa las imágenes en el tiempo t y $t - 1$ en un bloque *V-Net* cada una, obteniendo un bloque de características de $48 \times 4 \times 4$ para cada imagen. Estas se concatenan y son usadas como entrada de una capa convolucional de 48 filtros de 1×1 , pasándose finalmente a una capa densa de 16 neuronas. El procesamiento de las distintas vistas de las imágenes se hace cada una en su respectivo bloque, finalmente concatenándose los mapas de características obtenidos.

La capa densa está compuesta por una capa de 16 neuronas seguida por la capa de salida de 2 neuronas con activación *Softmax*. La función de activación utilizada en cada capa (con excepción de la capa de salida) es la función *Leaky ReLU*. La arquitectura no hace uso de *Pooling* y aplica una capa de *Dropout* después de cada capa con parámetro 0,25.

3 | Propuesta de Solución

La propuesta de solución al problema de segmentación de lesiones de esclerosis múltiple en imágenes de resonancia magnética consiste en la implementación y evaluación de dos distintos métodos aplicados en la literatura al problema. El conjunto de datos con el cual se validará la propuesta, corresponde a la base de datos de imágenes de entrenamiento utilizadas en el desafío de segmentación de lesiones de esclerosis múltiple organizado por la *Medical Image Computing and Computer Assisted Intervention Society* (MICCAI) del año 2008.

Esta base de datos se encuentra disponible al público y está compuesta por imágenes de resonancia magnética de veinte distintos pacientes en 3 distintas modalidades: *T1-Weighted*, *T2-Weighted* y *T2-Fluid Attenuated Inversion Recovery* (FLAIR), con sus respectivas máscaras de lesiones segmentadas. Diez de estas imágenes en sus 3 modalidades, con sus respectivas segmentaciones, fueron proveídas por la Universidad de Carolina del Norte (UNC) y las otras diez fueron proveídas por el Hospital de Niños de Boston (CHB).

Las modalidades de cada imagen del conjunto de datos se encuentran corregistradas, es decir, los vóxeles de cada imagen corresponden a una misma coordenada espacial del cerebro, sin embargo carecen de un preprocesamiento mayor, por lo cual es necesaria una etapa de preprocesamiento previa a la aplicación de los algoritmos de segmentación para obtener buenos resultados.

Una vez preprocesadas las imágenes, se procederá a implementar dos algoritmos de segmentación de la literatura. Una vez evaluado el estado del arte, se optó por implementar una técnica basada en el Modelo de Mezcla de Gaussianas (*Gaussian Mixture Models*) basada en la búsqueda de *outliers* en el modelo y el uso de reglas destinadas a eliminar falsos positivos de la segmentación. Por otra parte, se optó por la implementación de una Red Neuronal Convolutiva para la segmentación de lesiones, método que se ha vuelto muy popular en la literatura en los últimos años.

Para comparar la efectividad de ambos métodos, se hará uso de las métricas de *Sensitivity* (*True Positive Rate*) y *Precision* (*Positive Predictive Value*), las cuales están dadas por:

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

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

Donde TP es el número de verdaderos positivos de la segmentación (lesiones etiquetadas correctamente), FP es el número de falsos positivos de la segmentación (regiones sanas que fueron etiquetadas como lesiones por la segmentación) y FN es el número de falsos negativos de la segmentación (lesiones que fueron etiquetadas como tejido saludable). Cada una de estas medidas se cuantifica a nivel de vóxel.

La *Sensitivity* representa el porcentaje de la lesión que fue correctamente segmentada, mientras que la *Precision* mide el porcentaje de aciertos con respecto a la región segmentada. Una alta *Sensitivity* indica que la mayoría de los vóxeles de lesión están siendo segmentados correctamente, mientras que una *Sensitivity* baja indica que las lesiones no están siendo detectadas o que se está detectando solo una pequeña porción de estas. Por otra parte, una alta *Precision* indica que de los vóxeles segmentados como lesión un muy pequeño porcentaje corresponde a tejido saludable, mientras que una baja *Precision* indica que el porcentaje de tejido saludable que está siendo segmentado como lesión es muy alto, producto del gran número de falsos positivos.

Finalmente, se hará uso de coeficiente de similaridad de Sørensen Dice, métrica estándar utilizada en la literatura que permite determinar el nivel de traslape entre la segmentación obtenida y la segmentación proveida por un experto, el cual está dado por la media armónica entre las 2 métricas anteriores:

$$DSC = \frac{2}{\frac{1}{TPR} + \frac{1}{PPV}}$$

$$= \frac{2TP}{2TP + FP + FN}$$

4 | Construcción e Implementación de la solución

4.1. Preprocesamiento

Para obtener buenos resultados en los algoritmos de segmentación, fue necesario realizar una etapa de preprocesamiento para eliminar el ruido de la imagen y otras irregularidades presentes.

4.1.1. Bias Field Correction

Las imágenes de resonancia magnética son víctima de una señal que corrompe la calidad de la imagen conocido como *Bias Field*. Esta señal es producto de las inhomogeneidades en el campo magnético utilizado para obtener estas imágenes y su presencia resulta en sectores borrosos en la imagen, añadiendo ruido a los valores de intensidad de la imagen, lo cual afecta negativamente la calidad de los resultados obtenidos por algoritmos de procesamiento y segmentación de imágenes, ya que estas perturbaciones corrompen la distribución de intensidades de los distintos objetos y regiones que aparecen en la imagen. Para corregir esta señal, se hizo uso del algoritmo N4ITK.

El algoritmo N4ITK consiste en una modificación del algoritmo N3 (*non-parametric non-uniformity normalization*) que busca resolver algunas falencias del algoritmo [15]. Dada una imagen, el algoritmo modela el *bias field* como una función que multiplica la imagen sin corrupción como:

$$v(x) = u(x)f(x) + n(x)$$

Donde:

$v(x)$ = imagen original

$u(x)$ = imagen sin corrupción

$f(x)$ = *bias field*

$n(x)$ = ruido (se asume que es ruido gaussiano independiente)

Utilizando la notación $\hat{u} = \log u$, la ecuación anterior aplicando logaritmo queda de la forma:

$$\hat{v}(x) = \hat{u}(x) + \hat{f}(x)$$

A partir de este punto, la imagen sin corrupción se estima iterativamente mediante:

$$\begin{aligned}\hat{u}^n &= \hat{u}^{n-1} - \hat{f}_r^n \\ &= \hat{u}^{n-1} - S^*\{\hat{u}^{n-1} - E[\hat{u}|\hat{u}^{n-1}]\}\end{aligned}$$

Donde $S^*\{\cdot\}$ es una aproximación mediante *B-Splines* del residuo del *bias field* entre la n -ésima iteración y la $n - 1$ -ésima iteración. Los detalles de esta implementación se pueden encontrar en las referencias [15].

4.1.2. Skull Stripping

Para evitar involucrar los vóxeles de las imágenes correspondientes al *background* y al cráneo, se realizó la tarea de *Skull Stripping* para obtener una máscara binaria que solo incluyera aquellos voxeles correspondientes a tejidos cerebrales.

Para esta tarea, se hizo uso del software *BrainSuite*. Se realizó la tarea de *Skull Stripping* utilizando el algoritmo BSE (*Brain Surface Extractor*). La selección de parámetros se realizó de manera manual para cada imagen hasta obtener una máscara apropiada que no incluyera voxeles del cráneo. Los parámetros que produjeron resultados satisfactorios para cada una de las imágenes de la base de datos del CHB fueron los siguientes:

Imagen	Edge Constant	Erosion Size
CHB01	0.7	2
CHB02	0.7	2
CHB03	0.7	2
CHB04	0.9	2
CHB05	0.6	3
CHB06	0.7	3
CHB07	0.7	2
CHB08	0.7	3
CHB09	0.7	3
CHB10	0.7	3

Tabla 4.1: Parámetros usados en el BSE

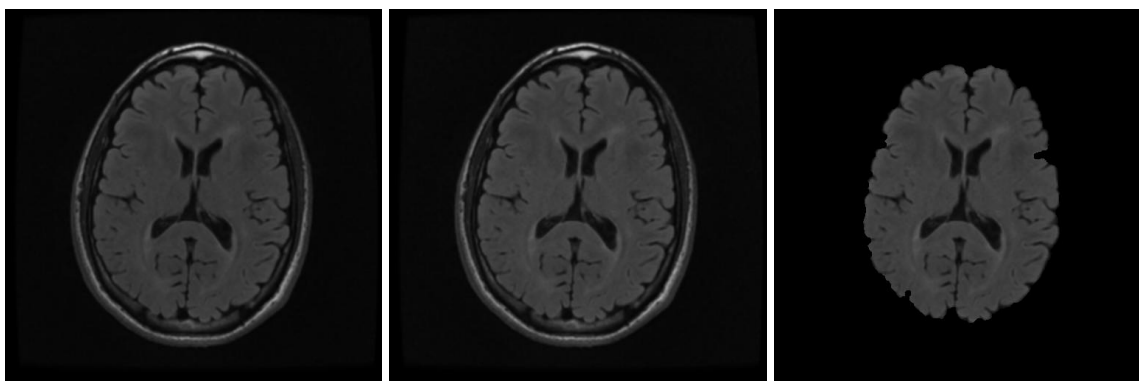


Figura 4.1: Imagen FLAIR Original

Figura 4.2: Imagen tras aplicar *Bias Field Correction*

Figura 4.3: Imagen tras aplicado el *Skull Stripping*

4.2. Gaussian Mixture Models

El primer método elegido para realizar segmentación, consiste en el modelamiento de las intensidades de los 3 tipos de tejido cerebral usando *Gaussian Mixture Models* de 3 componentes, similar al pocedimiento realizado en el paper de Karimian y Jafari [10]. La segmentación se realizará buscando los *outliers* del modelo derivados a partir del uso del estimador truncado de máxima verosimilitud del GMM, seguido de la aplicación de un conjunto de reglas para eliminar falsos positivos del modelo.

El algoritmo propuesto en Karimian y Jafari [10] depende de múltiples parámetros, entre ellos el porcentaje de *outliers* que se obtienen del GMM y los umbrales aplicados a cada modalidad de la imagen con el fin de eliminar outliers. Para elegir dichos parámetros, se ajustaron múltiples

GMM para cada imagen haciendo variar el porcentaje de *outliers* en el modelo h entre los valores $\{0,1; 0,2; 0,3; 0,4; 0,5\}$. Luego, para cada modelo obtenido, se hicieron variar los umbrales definidos en las reglas de selección de lesiones para la distancia de Mahalanobis, intensidad de la imagen T2 e intensidad de la imagen FLAIR. El rango de valores para la distancia de Mahalanobis probado fue de $\{0,4; 0,8; 1,2; 1,6; 2\}$. Los parámetros elegidos finalmente corresponden a la combinación que obtenga el mayor DSC.

Un problema que fue observado al ajustar los GMM para las distintas imágenes de la base de datos, es que las distribuciones de intensidad para la materia blanca en las imágenes T2 y FLAIR variaban mucho entre las distintas imágenes, por lo cual el uso de un umbral fijo único para las imágenes T2 y FLAIR no produjo buenos resultados en la segmentación. Para solucionar este problema, se tomó una idea similar a la propuesta en los papers de Lorenzo [11] y Souplet [9] de utilizar umbrales basados en parámetros del modelo obtenido. De este modo, se seleccionan las regiones de hiperintensidad en la materia blanca de las imágenes T2 y FLAIR mediante los umbrales $\mu_{WM,T2} + k_{T2} \cdot \sigma_{WM,T2}$ y $\mu_{WM,FLAIR} + k_{FLAIR} \cdot \sigma_{WM,FLAIR}$, donde $\mu_{WM,i}$ es la media de las intensidades de la materia blanca en la imagen en modalidad i , $\sigma_{WM,i}$ es la desviación estándar de las intensidades de la materia blanca en la imagen en modalidad i , y k_{T2} y k_{FLAIR} son 2 constantes asociadas al umbral su imagen respectiva. Estos 2 últimos parámetros reemplazan el umbral fijo definido por Karimian y Jafari, y se hicieron variar en el rango de valores $\{0, 1, 2\}$. Por último, se añadió además al conjunto de reglas la definición de hipointensidad en la imagen T1, definida mediante el umbral negativo $\mu_{WM,T1} - k_{T1} \cdot \sigma_{WM,T1}$. El parámetro k_{T1} se hizo variar en el mismo rango que k_{T2} y k_{FLAIR} .

Otro problema observado al aplicar el método descrito en el paper es que una gran mayoría de las lesiones que se predecían se encontraban en el borde externo de la materia blanca, en la frontera con la materia gris cortical. Este problema es descrito en el paper de Datta et. al. [5], y la solución que proponen consiste en aplicar una máscara obtenida a partir de la aplicación de la operación morfológica de erosión a la máscara de materia blanca, permitiendo eliminar los vóxeles cercanos de esta de los candidatos a lesión. Antes de aplicar la máscara de erosión, se rellenaron los agujeros internos de la máscara de materia blanca. El tamaño del filtro de erosión elegido fue de $9 \times 9 \times 9$. El tamaño de este filtro fue definido en base al trabajo de Sajja et. al. [3], quienes informan haber aplicado un filtro de erosión de 3×3 a cada *slice* de su conjunto de imágenes, cuya resolución era la mitad de la resolución de las imágenes con las que se trabajó, por lo cual inicialmente se fijó el

Imagen	$\mu_{WM,T2}$	$\mu_{FLAIR,T2}$
CHB01	77.89	73.70
CHB02	37.76	62.18
CHB03	38.08	80.85
CHB04	46.40	100.96
CHB05	45.26	64.65
CHB06	50.01	61.64
CHB07	53.88	59.78
CHB08	43.88	60.70
CHB09	58.38	57.84
CHB10	58.33	56.88

Tabla 4.2: Parámetros de medias obtenidos para la Materia Blanca

tamaño del filtro a $6 \times 6 \times 6$. Los autores del paper informan haber observado una pequeña cantidad de vóxeles en la frontera aún siendo segmentados como lesión, por lo cual se fijó su tamaño final a $9 \times 9 \times 9$.

La segmentación inicial para obtener los estimadores de los parámetros iniciales del modelo se realizó sobre la imagen T1 mediante el algoritmo *K-Means*.

4.2.1. Algoritmo K-Means

El algoritmo *K-Means* es un algoritmo de aprendizaje no supervisado comúnmente aplicado en tareas de clustering. Dado un conjunto de puntos $X = \{x_1, x_2, \dots, x_n\}$ y un parámetro k , correspondiente a un entero positivo, el algoritmo se inicializa eligiendo k puntos al azar c_1, c_2, \dots, c_k dentro del mismo dominio de X conocidos como centroides. Luego, para cada punto dentro del conjunto X , se calcula la distancia entre x_i y cada uno de los centroides c_j (comúnmente, se utiliza la distancia euclidiana) y se asigna el punto x_i al cluster correspondiente al centroide más cercano. Finalmente, se calculan nuevamente los centroides como:

$$c_j = \frac{1}{n_j} \sum_{x_i \in C_j} x_i$$

Donde C_j es el cluster que contiene los puntos que anteriormente eran más cercanos a c_j y n_j es el número de puntos en el cluster C_j . Estos 2 pasos se repiten iterativamente hasta que la asignación de clusters para cada punto no cambie de una iteración a la siguiente.

Este algoritmo es aplicable a la tarea de segmentación de imágenes. Dado el conjunto de píxeles de una imagen en escala de grises, los puntos a agrupar corresponden a las intensidades de los píxeles, mientras que los centroides corresponden a distintos valores de intensidad en el rango $[0, 255]$. Como resultado, el algoritmo *K-Means* realiza una agrupación de los píxeles con valores similares de intensidad. El algoritmo *K-Means* ha sido utilizado para realizar segmentación de los distintos tejidos cerebrales en imágenes de resonancia magnética [6].

El algoritmo *K-Means* fue utilizado para realizar una segmentación preliminar del cerebro en los 3 distintos tipos de tejido basado en la intensidad de los píxeles en la imagen T1 obteniendo, de menor a mayor intensidad, etiquetas para los píxeles correspondientes a líquido cefalorraquídeo, materia gris y materia blanca.

4.3. Redes Neuronales Convolucionales

El segundo método escogido para realizar segmentación, consiste en la primera arquitectura propuesta por Birenbaum [4], la cual hace uso de un submuestreo de los parches de la imagen FLAIR para entrenar una red convolucional. Se tomó la arquitectura SISTP (*Single Image, Single Time Point*) propuesta en el paper, debido a que se posee solo una imagen para cada paciente, en lugar de una MRI longitudinal.

Además de la arquitectura propuesta, se experimentó utilizando una arquitectura sin la capa de *Max Pooling* que prosigue a las capas de convolución, con el fin de comparar el desempeño de la arquitectura utilizada ante la ausencia de esta capa.

Para ambas arquitecturas se probó con 2 distintos tipos de entrada: 1 imagen en vista axial y 1 imagen para cada tipo de vista (axial, sagital y coronal). Se eligió la vista axial, ya que es la que fue obtenida con mejor resolución acorde a la documentación de la base de datos, siendo cada *slice* obtenida con una resolución 0.5 mm, con una distancia de 1 mm entre cada *slice*, teniéndose que interpolar los valores intermedios durante el proceso de registrado para obtener la imagen final con resolución de 0.5 mm. Para la arquitectura multivista, los vectores de características obtenidos tras la primera capa densa de la red para cada imagen se concatenan, obteniendo un vector de largo 48, el cual se pasa a una red densa de 16 neuronas seguido por la capa *Softmax*.

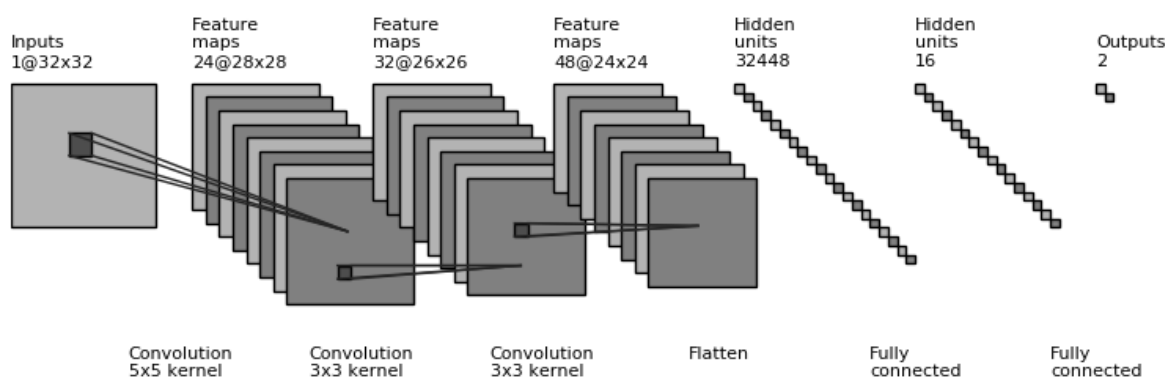


Figura 4.4: Arquitectura para 1 entrada implementada de la CNN. Elaboración Propia.

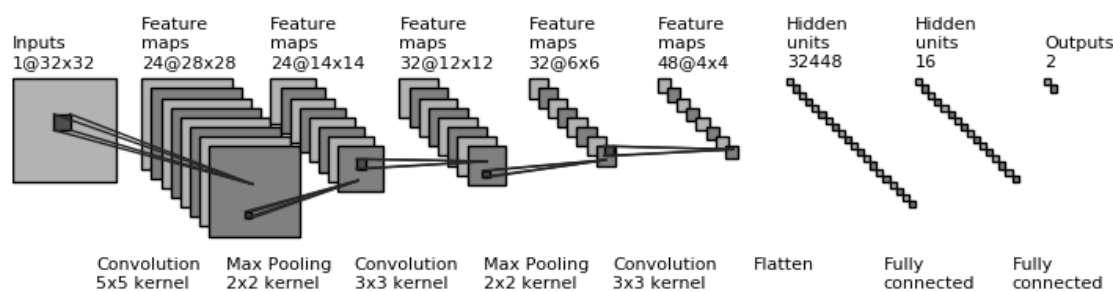


Figura 4.5: Arquitectura para 1 entrada implementada de la CNN sin capa de *Max Pooling*. Elaboración Propia.

La implementación de la red se realizó utilizando Keras [1], biblioteca escrita en Python para el manejo y construcción de redes neuronales de manera modular, utilizando la biblioteca Tensorflow [2] como *backend*.

La selección de vóxeles candidatos se realizó en base a la segmentación de materia blanca preliminar obtenida a partir del algoritmo *K-Means* aplicado a la imagen T1. Esta máscara para la materia blanca se utilizó en reemplazo de la segmentación obtenida a partir del registro de un Atlas a la imagen T1. El parámetro de umbral para la imagen FLAIR, similar al método anterior, se definió en base a la distribución de intensidades de la materia blanca en la imagen FLAIR, eligiéndose como candidatos aquellos vóxeles cuya intensidad estuviera por sobre el umbral dado

por $\mu_{WM,FLAIR} + k_{FLAIR} \cdot \sigma_{WM,FLAIR}$. El parámetro se fijó de tal manera que la proporción de vóxeles candidatos calzara con la descrita en el paper de Birenbaum et. al.

El número de candidatos obtenido finalmente para la red fue de 1242109 vóxeles, tomando el parámetro $k_{FLAIR} = 2$, lo cual corresponde aproximadamente a 1,5 % del total de vóxeles obtenidos de la imagen una vez aplicado el paso de *Skull Stripping*, lo cual es cercano al porcentaje informado en el paper de Birenbaum, quienes aseguran que el muestreo de candidatos redujo las computaciones de la red en un 97,5 % [4]. Cabe destacar que dentro de este submuestreo, se está dejando fuera un porcentaje de los vóxeles de la imagen que corresponden a lesiones.

Además del muestreo de parches, otra medida tomada para mitigar el problema de desbalance de clases fue el utilizar una función de pérdida que penalizara de mayor manera los falsos negativos de las lesiones. El parámetro de peso de esta función de pérdida fue determinado en base al porcentaje de lesiones observado en el conjunto de entrenamiento, el cual se terminó fijando en 20 % para todas las redes entrenadas.

Los resultados para la red se evaluarán utilizando *leave-one-out Cross Validation*. Sea n el número de imágenes de entrenamiento, este método de validación consiste en entrenar la red utilizando $n - 1$ imágenes. Luego, se aplica el modelo obtenido por la red a la imagen con la que no se entrenó el modelo, obteniéndose las métricas de evaluación del modelo para esta última imagen. Este proceso se repite para cada imagen del conjunto de entrenamiento. Finalmente, se obtiene un promedio de las métricas de evaluación para el modelo obtenido.

5 | Resultados

Los resultados de la selección de parámetros para el GMM se presentan en los siguientes gráficos (5.1, 5.2, 5.3, 5.4), detallando el DSC promedio obtenido para cada parámetro a lo largo de todas las imágenes de la base de datos. La mejor combinación de parámetros encontrada, con respecto al DSC obtenido fue $h = 0,2$; $d_{\min} = 0,4$; $k_{T2} = 0$ y $k_{FLAIR} = 1$.

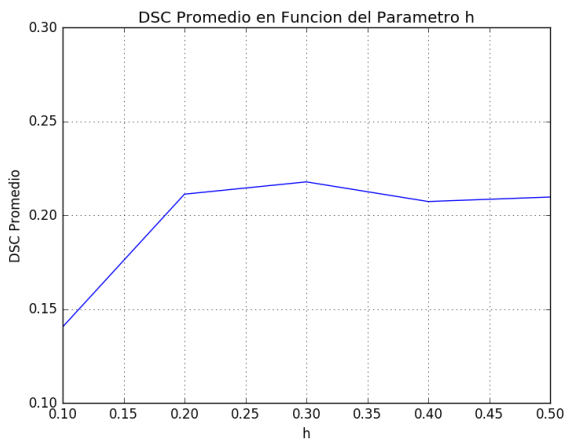


Figura 5.1: Parámetro h

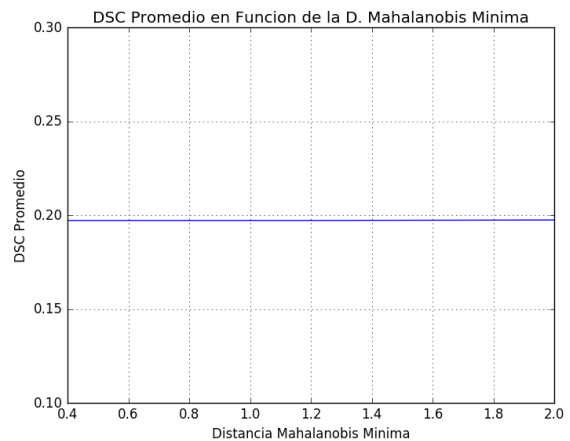


Figura 5.2: Parámetro d_{\min}

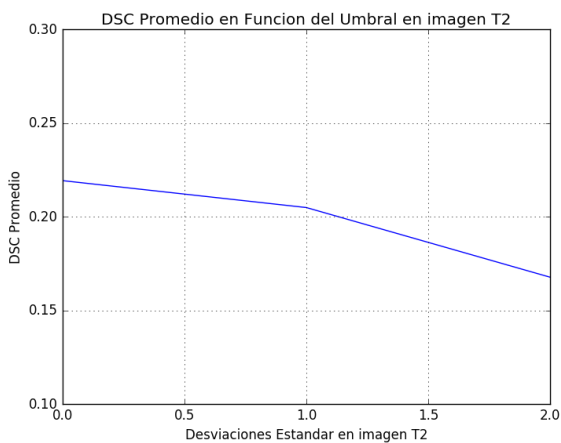


Figura 5.3: Parámetro k_{T2}

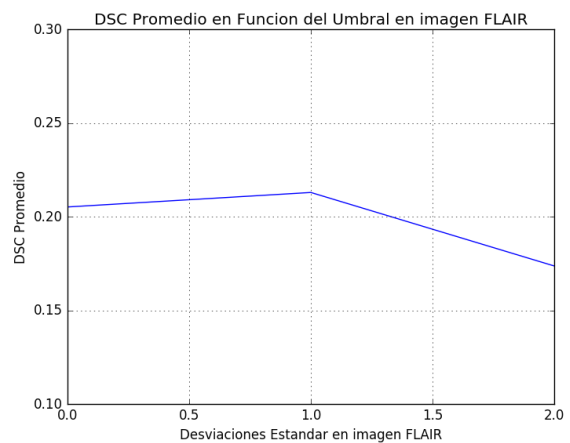


Figura 5.4: Parámetro k_{FLAIR}

Naturalmente, a medida que se aumentan el parámetro h y se disminuyen los parámetros d_{\min} , k_{T2} y k_{FLAIR} el número de verdaderos positivos aumenta, debido a que se está tomando una mayor región del cerebro como candidatos a lesión, sin embargo esto también trae como consecuencia la segmentación de una mayor cantidad de falsos positivos, de los cuales solo una fracción es eliminada mediante la aplicación de las reglas de disminución de falsos positivos. El parámetro h además trae el problema de que si se aumenta demasiado se pierde mucha información en el GMM, debido a que se estaría entrenando con una menor cantidad de vóxeles, provocando que los resultados de la segmentación inicial sean pobres.

La adición de la regla de eliminación de vóxeles cercanos a la frontera de la materia gris, mediante la erosión de la máscara de materia blanca, disminuyó de manera considerable el número de falsos positivos, ya que una gran parte de los outliers detectados en el modelo se hallaban en esta región y no estaban siendo filtrados por las reglas de selección de lesiones. La comparación del *Positive Predictive Value* antes y después de la aplicación de esta regla se presenta en la siguiente tabla:

Imagen	PPV del GMM	PPV post erosión
CHB01	0.74	0.79
CHB02	0.04	0.53
CHB03	0.05	0.70
CHB04	0.06	0.30
CHB05	0.30	0.47
CHB06	0.18	0.43
CHB07	0.33	0.65
CHB08	0.10	0.65
CHB09	0.03	0.09
CHB10	0.01	0.16

Tabla 5.1: Comparación de PPV antes y después de aplicación de Erosión a la máscara de Materia Blanca



Figura 5.5: Segmentación obtenida mediante GMM en la imagen CHB02

Figura 5.6: Segmentación tras eliminación de candidatos fronterizos

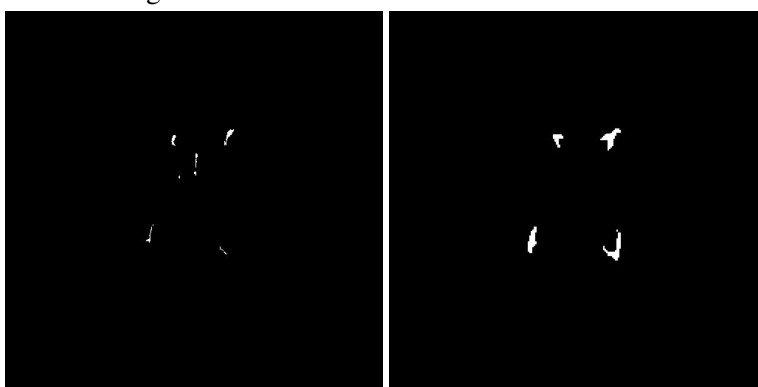


Figura 5.7: Segmentación tras aplicación de reglas de selección de lesiones

Figura 5.8: Segmentación del experto realizada por miembros del CHB

Para comparar las técnicas implementadas con el Estado del Arte, se investigaron los resultados obtenidos por otras técnicas en la base de datos pública del *MS Lesion Segmentation Challenge*, ya que los papers originales de los que provenían las técnicas implementadas utilizan distintas bases de datos para validar sus experimentos. Los resultados de dichas técnicas y las implementadas se encuentran detallados en la tabla 5.2.

De manera general, los resultados obtenidos se caracterizan por lograr una segmentación relativamente precisa de las lesiones de esclerosis múltiple, sin arrojar una gran cantidad de falsos positivos, lo cual se ve reflejado en el elevado PPV obtenido en la mayoría de las técnicas aplicadas. El *tradeoff* de esto, es que en general el TPR obtenido a lo largo de las técnicas fue relativamente bajo comparado con las técnicas del estado del arte. En el caso de la segmentación obtenida a partir del GMM, esto se puede atribuir a la aplicación del conjunto de reglas, dentro de las cuales no

Imagen	Souplet et. al. [9]			Geremia et. al. [7]			Weiss et. al. [16]			Mechrez et. al. [18]		
	TPR	PPV	DSC	TPR	PPV	DSC	TPR	PPV	DSC	TPR	PPV	DSC
CHB01	0.22	0.41	0.28	0.49	0.64	0.75	0.60	0.58	0.59	0.40	0.33	0.36
CHB02	0.18	0.29	0.22	0.44	0.63	0.69	0.27	0.45	0.34	0.41	0.11	0.17
CHB03	0.17	0.21	0.19	0.22	0.57	0.39	0.24	0.56	0.34	0.47	0.19	0.27
CHB04	0.12	0.55	0.20	0.31	0.78	0.50	0.27	0.66	0.38	0.37	0.07	0.12
CHB05	0.22	0.42	0.29	0.40	0.52	0.45	0.29	0.33	0.31	0.58	0.25	0.35
CHB06	0.13	0.46	0.20	0.32	0.52	0.40	0.10	0.36	0.16	0.43	0.38	0.41
CHB07	0.13	0.39	0.20	0.40	0.54	0.46	0.14	0.48	0.22	0.34	0.50	0.41
CHB08	0.13	0.55	0.21	0.46	0.65	0.54	0.21	0.73	0.33	0.48	0.52	0.50
CHB09	0.03	0.18	0.05	0.23	0.28	0.25	0.05	0.22	0.08	0.31	0.23	0.26
CHB10	0.05	0.18	0.08	0.23	0.39	0.29	0.15	0.12	0.13	0.13	0.28	0.18
Promedio	0.14	0.36	0.19	0.35	0.55	0.47	0.23	0.45	0.29	0.39	0.29	0.30

Tabla 5.2: Resultados de técnicas del Estado del Arte

Imagen	GMM			CNN			CNN (Max Pooling)			Multiview CNN			Multiview CNN (MP)		
	TPR	PPV	DSC	TPR	PPV	DSC	TPR	PPV	DSC	TPR	PPV	DSC	TPR	PPV	DSC
CHB01	0.41	0.79	0.54	0.46	0.54	0.50	0.27	0.57	0.37	0.44	0.56	0.50	0.18	0.61	0.27
CHB02	0.13	0.53	0.21	0.38	0.59	0.46	0.17	0.77	0.28	0.41	0.55	0.47	0.24	0.53	0.33
CHB03	0.18	0.70	0.29	0.34	0.49	0.40	0.31	0.50	0.38	0.35	0.47	0.40	0.22	0.63	0.32
CHB04	0.42	0.30	0.35	0.15	0.13	0.14	0.34	0.18	0.23	0.28	0.18	0.22	0.30	0.29	0.29
CHB05	0.18	0.59	0.27	0.18	0.18	0.18	0.19	0.23	0.21	0.18	0.20	0.19	0.16	0.28	0.21
CHB06	0.11	0.43	0.17	0.24	0.57	0.34	0.22	0.60	0.32	0.23	0.60	0.33	0.19	0.64	0.30
CHB07	0.26	0.65	0.38	0.31	0.68	0.42	0.26	0.72	0.38	0.30	0.72	0.42	0.23	0.74	0.35
CHB08	0.23	0.60	0.32	0.33	0.76	0.46	0.25	0.76	0.38	0.33	0.77	0.46	0.22	0.75	0.34
CHB09	0.03	0.09	0.05	0.17	0.50	0.25	0.15	0.50	0.23	0.20	0.41	0.27	0.09	0.59	0.15
CHB10	0.02	0.16	0.04	0.07	0.70	0.13	0.05	0.60	0.10	0.11	0.61	0.18	0.03	0.77	0.06
Promedio	0.20	0.48	0.26	0.26	0.52	0.33	0.22	0.54	0.29	0.28	0.51	0.34	0.19	0.58	0.26

Tabla 5.3: Resultados de técnicas implementadas

solamente se están filtrando falsos positivos, si no que en algunos casos se eliminan vóxeles de la segmentación que efectivamente correspondían a lesiones. Para el caso de las redes convolucionales, el bajo número de verdaderos positivos se puede atribuir al submuestreo realizado para disminuir los tiempos de entrenamiento de la red.

La principal ventaja observada por en el GMM es el tiempo necesario para obtener una segmentación. A lo largo de las 10 imágenes, el tiempo promedio requerido para obtener la segmentación de cada imagen, incluyendo la lectura del archivo y aplicación del algoritmo EM al modelo fue de $538,61 \pm 15,47$ segundos. De este tiempo, cada iteración del algoritmo EM tomó $27,96 \pm 3,04$ segundos en tiempo de ejecución a lo largo de los distintos casos de prueba.

Las redes convolucionales obtuvieron mejores resultados en promedio, lográndose un coeficiente de similaridad de 0.33 y 0.29 en el caso de la red de 1 vista y de 0.34 y 0.26 en el caso de la red multivista, los cuales son comparables con la mayor parte de los resultados en el estado del arte.

En términos de la calidad de los resultados, el desempeño en promedio de la red fue superior ante la ausencia de la capa de *Max Pooling*. Los problemas observados al agregar la capa de *Max Pooling* consistieron en una disminución en el volumen detectado de las lesiones, lo cual indica que la red tiende a subestimar el tamaño de las lesiones (lo cual explica el bajo TPR obtenido), y el hecho de que la red simplemente no logra identificar algunas de las lesiones.

El número de lesiones que fueron detectadas, es decir aquellas para las cuales al menos 1 vóxel fue clasificado como lesión por las redes convolucionales, para cada imagen fueron los siguientes:

Imagen de test	CNN	CNN (<i>MP</i>)	Multiview CNN	Multiview CNN (<i>MP</i>)	Total de Lesiones
CHB01	47	33	50	29	58
CHB02	13	8	14	9	23
CHB03	10	9	10	8	19
CHB04	20	20	20	19	24
CHB05	29	29	29	29	30
CHB06	60	55	58	48	96
CHB07	22	17	18	17	31
CHB08	47	40	42	32	87
CHB09	14	11	21	9	27
CHB10	5	4	5	5	7

Tabla 5.4: Número de lesiones segmentadas

En la mayoría de las imágenes, la red con capa de *Max Pooling* logró detectar solo un subconjunto de las lesiones que logró detectar la red sin esta. La única excepción a esto fue la imagen CHB08, en la cual la red con *Max Pooling* para la vista axial logró hallar 2 lesiones que no fueron detectadas por la otra red, sin embargo estos aciertos podrían ser calificados como ruido simplemente viendo los resultados de la segmentación, debido a que aparecen solamente en 1 o 2 *slices* de la imagen y cubren una región muy pequeña.

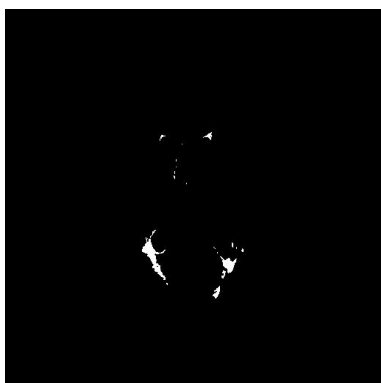


Figura 5.9: Segmentación obtenida por la CNN sin capa de *Max Pooling*

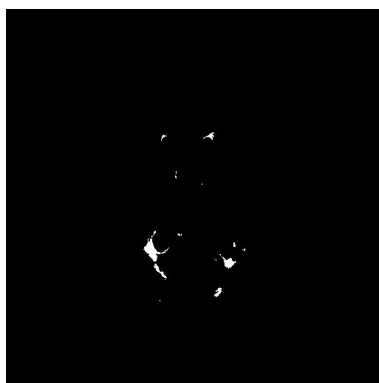


Figura 5.10: Segmentación obtenida por la CNN con capa de *Max Pooling*



Figura 5.11: Segmentación del experto realizada por miembros del CHB

Además, en estas imágenes (5.23, 5.24) se puede observar que las regiones segmentadas co-

rrespondientes a una lesión no necesariamente son contiguas, pudiendo aparecer múltiples lesiones segmentadas que corresponden en realidad a una única lesión.

Imagen	CNN		CNN (Max Pooling)		Multiview CNN		Multiview CNN (MP)	
	N° Regiones	Tamaño > 6	N° Regiones	Tamaño > 6	N° Regiones	Tamaño > 6	N° Regiones	Tamaño > 6
CHB01	709	171	397	110	531	127	514	114
CHB02	199	58	149	40	418	104	496	123
CHB03	477	119	430	110	460	111	195	53
CHB04	1725	283	2105	303	1388	219	834	164
CHB05	1206	235	686	149	898	185	395	91
CHB06	677	167	517	151	605	158	356	116
CHB07	506	107	401	96	366	85	393	99
CHB08	551	130	470	118	424	105	388	87
CHB09	354	74	382	94	620	123	158	40
CHB10	161	38	134	32	312	81	61	20

Tabla 5.5: Número de regiones segmentadas

Se puede observar que el número de regiones que son segmentadas por la CNN es bastante grande, lo cual se observa en los resultados como un gran número de regiones de pequeño tamaño que aparecen en la segmentación alrededor de regiones más grandes (por lo general lesiones bien clasificadas), regiones alrededor de los ventrículos laterales del cerebro y en los bordes exteriores de la materia blanca, en la frontera con la materia gris. Una alternativa, estudiada más adelante, para mitigar este problema de discontinuidad consiste en realizar una etapa de postprocesamiento en la cual se apliquen operaciones morfológicas a la máscara de la segmentación, permitiendo mejorar su calidad.

Otro problema observado con la segmentación de las redes es la dependencia en la selección inicial de candidatos. Si bien gran parte de la aproximación de la materia blanca utilizando *K-Means* arrojó buenos resultados, existen algunos casos particulares en los cuales las zonas de hipointensidad en la materia blanca fueron clasificadas como otro tipo de tejido.

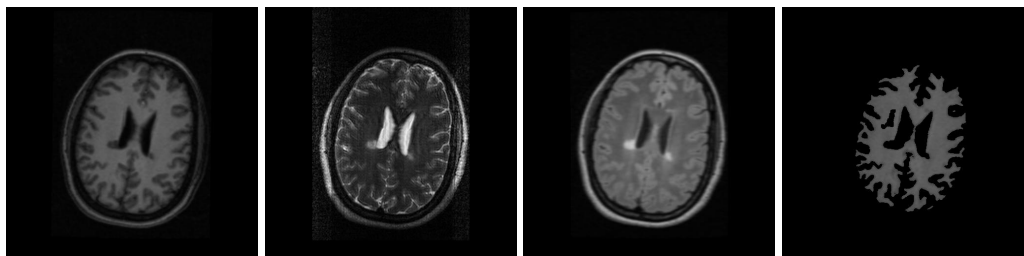


Figura 5.12: Imagen T1 Preprocesada **Figura 5.13:** Imagen T2 Preprocesada **Figura 5.14:** Imagen FLAIR Preprocesada **Figura 5.15:** WM obtenida de *K-Means*

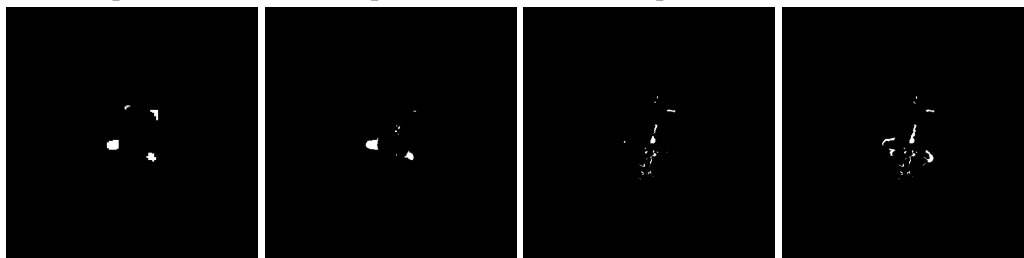


Figura 5.16: Segmentación entregada por el experto del CHB

Figura 5.17: Segmentación obtenida a partir del GMM

Figura 5.18: Segmentación obtenida por CNN (sin *Max Pooling*)

Figura 5.19: Segmentación obtenida por CNN (con *Max Pooling*)



Figura 5.20: Segmentación de CNN Multivista (sin *Max Pooling*)

Figura 5.21: Segmentación de CNN Multivista (con *Max Pooling*)

Figura 5.22: Ejemplo de imagen en la cual falla la segmentación automática de WM

Se puede observar que en este caso la segmentación del GMM es más acertada debido a que en este modelo se toman en cuenta todos los tipos de tejido al momento de determinar los candidatos a lesión. Cabe destacar que, si bien el GMM falló de igual manera que *K-Means* en segmentar la materia blanca, el hecho de que se consideren los vóxeles de todos los tipos de tejido en la selección de *outliers* permitió detectar de mejor manera las lesiones, no solo tomando en cuenta los vóxeles que se predijeron como materia blanca.

Una de las ventajas observadas para la arquitectura que hace uso de la capa de *Max Pooling* es

el tiempo requerido para el entrenamiento de esta. Debido a la disminución en el tamaño de los mapas de características tras la aplicación de la capa de *Max Pooling*, el tiempo de entrenamiento de la red disminuyó en un factor de 4 aproximadamente. En la siguiente tabla, se detalla el tiempo promedio aproximado en la CPU (en horas, minutos y segundos de ejecución) requerido para 1 *epoch* de entrenamiento:

Imagen de test	CNN	CNN (<i>Max Pooling</i>)
CHB01	21:50:09	05:23:32
CHB02	21:26:50	05:11:42
CHB03	22:47:30	05:36:25
CHB04	23:02:39	05:46:33
CHB05	23:10:10	05:31:57
CHB06	23:24:33	05:38:44
CHB07	23:03:59	05:30:15
CHB08	22:03:02	05:31:33
CHB09	20:01:39	05:32:05
CHB10	21:37:01	05:29:07

Tabla 5.6: Tiempos de ejecución para las redes convolucionales en la CPU

Cabe destacar que no se registraron los resultados de las redes multivista para este caso debido al elevado tiempo requerido para que estas se entrenaran en la CPU.

En la GPU, el tiempo de entrenamiento disminuye considerablemente. En la siguiente tabla, se detalla el tiempo promedio aproximado requerido para 1 *epoch* de entrenamiento:

Imagen de test	CNN	CNN (<i>Max Pooling</i>)	Multiview CNN	Multiview CNN (MP)
CHB01	00:02:36	00:01:18	00:05:02	00:02:30
CHB02	00:02:25	00:01:12	00:05:09	00:02:21
CHB03	00:02:21	00:01:15	00:05:13	00:02:38
CHB04	00:03:03	00:01:23	00:05:15	00:02:43
CHB05	00:02:34	00:01:17	00:05:12	00:02:34
CHB06	00:02:43	00:01:14	00:05:09	00:02:26
CHB07	00:02:27	00:01:05	00:05:03	00:02:29
CHB08	00:02:01	00:01:03	00:04:48	00:02:17
CHB09	00:02:39	00:01:15	00:04:57	00:02:25
CHB10	00:02:31	00:01:17	00:05:00	00:02:19

Tabla 5.7: Tiempos de ejecución para las redes convolucionales en la GPU

Dependiendo del número de iteraciones necesarias, se puede observar que el tiempo requerido para entrenar las redes, incluso en la GPU, supera el tiempo promedio requerido para obtener resultados a partir del GMM, el cual fue de aproximadamente 9 minutos para cada imagen, tomando en cuenta que este tiempo incluye la lectura de archivos, ejecución del algoritmo y generación de las segmentaciones. No solo esto, sino que el algoritmo GMM tiene la ventaja de poder ser ejecutado en la CPU sin comprometer el tiempo de ejecución, a diferencia de las redes convolucionales, las cuales requirieron en algunos casos de múltiples días para su entrenamiento en la CPU.

5.1. Postprocesamiento

Con el fin de mejorar la calidad de la segmentación, se aplicaron operaciones morfológicas a las segmentaciones obtenidas. La operación realizada consiste en un cierre, el cual consiste en la aplicación de un filtro de dilatación seguido de la aplicación de un filtro de erosión. El tamaño de este filtro se fijó en 5×5 , con el fin de lograr conectar aquellas regiones que se encuentren cercanas unas a otras, pero no lo suficiente como para unir regiones que no estén relacionadas entre ellas.

Los resultados obtenidos tras aplicado este filtro, además del cambio porcentual observado para cada métrica, se encuentran en las siguiente tablas:

Imagen	GMM			CNN			CNN (Max Pooling)			Multiview CNN			Multiview CNN (MP)		
	TPR	PPV	DSC	TPR	PPV	DSC	TPR	PPV	DSC	TPR	PPV	DSC	TPR	PPV	DSC
CHB01	0.45	0.78	0.57	0.56	0.55	0.55	0.32	0.58	0.41	0.53	0.57	0.55	0.31	0.62	0.31
CHB02	0.14	0.51	0.22	0.42	0.59	0.49	0.18	0.77	0.29	0.45	0.54	0.49	0.25	0.53	0.34
CHB03	0.19	0.67	0.29	0.38	0.46	0.42	0.34	0.48	0.40	0.39	0.45	0.42	0.24	0.63	0.35
CHB04	0.43	0.25	0.32	0.16	0.11	0.13	0.37	0.16	0.22	0.31	0.17	0.22	0.33	0.28	0.31
CHB05	0.19	0.61	0.29	0.24	0.20	0.22	0.25	0.25	0.25	0.24	0.22	0.23	0.22	0.31	0.25
CHB06	0.12	0.40	0.18	0.29	0.56	0.38	0.26	0.60	0.36	0.28	0.58	0.38	0.23	0.64	0.34
CHB07	0.33	0.60	0.43	0.36	0.67	0.47	0.31	0.71	0.43	0.35	0.71	0.47	0.27	0.75	0.40
CHB08	0.26	0.57	0.36	0.41	0.75	0.53	0.30	0.75	0.43	0.40	0.76	0.53	0.25	0.75	0.38
CHB09	0.04	0.07	0.05	0.23	0.51	0.32	0.29	0.51	0.29	0.29	0.41	0.34	0.11	0.59	0.18
CHB10	0.02	0.12	0.04	0.09	0.70	0.16	0.07	0.59	0.12	0.13	0.59	0.22	0.04	0.77	0.07
Promedio	0.22	0.46	0.30	0.31	0.51	0.37	0.27	0.54	0.32	0.34	0.50	0.38	0.21	0.59	0.29

Tabla 5.8: Resultados de postprocesamiento.

Métrica	GMM	CNN	CNN (Max Pooling)	Multiview CNN	Multiview CNN (MP)
TPR	+10 %	+19.23 %	+22.73 %	+21.43 %	+10.53 %
PPV	-0.04 %	-0.02 %	0 %	-0.02 %	+0.02 %
DSC	+15.38 %	+12.12 %	+10.34 %	+11.76 %	+11.54 %

Tabla 5.9: Cambios en las métricas tras la aplicación del postprocesamiento.

Se puede observar que para todos los métodos, la calidad de la segmentación mejoró considerablemente. Esto se debe a que mediante la aplicación del cierre morfológico se logró mitigar uno de los problemas principales en las segmentaciones, el cual fue la subestimación del tamaño de las lesiones, las cuales en muchos casos estaban siendo identificadas como múltiples regiones aisladas. Mediante el cierre, las regiones cercanas se unen durante la aplicación del filtro de dilatación. Luego, tras aplicado el filtro de erosión, aquellas regiones que no se conectaron a otras vuelven a su estado original, mientras que aquellas que lograron conectarse se mantienen conectadas en su mayoría.

Otro problema que este filtro logra solucionar, es la aparición de pequeños agujeros al interior de algunas regiones en la segmentación. Este problema se observó más comúnmente en las redes neuronales, donde la selección de candidatos iniciales solía dejar fuera de la segmentación en algunos casos vóxeles que estaban rodeados en gran parte por otros que sí habían sido elegidos como candidatos, por lo cuales no se tomaban en cuenta al realizar la segmentación.

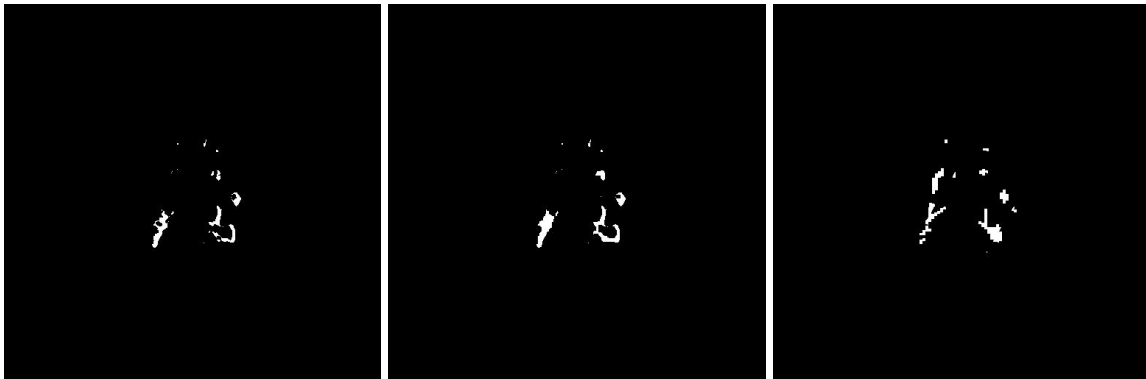


Figura 5.23: Segmentación previa al postprocesamiento

Figura 5.24: Segmentación tras realizado postprocesamiento

Figura 5.25: Segmentación del experto

Figura 5.26: Ejemplo de las ventajas del filtro de cierre

Cabe destacar que este filtro aumenta el número de falsos positivos en la segmentación, ya que permite que aquellas regiones correspondientes a falsos positivos se conecten entre sí de manera similar a las lesiones. Sin embargo, dada la mejora observada en las métricas de TPR y DSC, se puede decir que las ventajas de la aplicación del filtro superan a las desventajas de este, en especial dado que el filtro ataca las debilidades observadas en las técnicas, las cuales lograban obtener una segmentación con un bajo porcentaje tanto de falsos positivos como de aciertos a lesiones.

6 | Conclusiones

En este trabajo, se implementaron 2 distintas técnicas del Estado del Arte, además de proponerse algunas modificaciones a estas, para poder lograr obtener resultados consistentes con el Estado del Arte en la base de datos de entrenamiento del *MS Lesion Segmentation MICCAI 2008* con respecto al TPR, PPV y DSC. A partir de lo investigado en el Estado del Arte, estas técnicas, basadas en detección de outliers mediante el uso de GMM y segmentación mediante el uso de Redes Neuronales Convolucionales, no habían sido probadas en esta base de datos y no existían registros de su desempeño.

Ambas técnicas implementadas logran segmentar de manera precisa las lesiones de esclerosis múltiple, obteniéndose un alto PPV a lo largo de todas las técnicas. El principal problema observado para ambas técnicas implementadas es que el TPR promedio obtenido en general fue bajo.

La técnica que mejor desempeño mostró fueron las redes neuronales convolucionales, las cuales lograron clasificar exitosamente, en la mayoría de los casos, las lesiones de esclerosis múltiple a partir de la selección inicial de candidatos. Cabe mencionar que el número de imágenes utilizadas para obtener estas segmentaciones fue relativamente bajo, correspondiendo a aproximadamente 1,5 % de todos los parches que podrían haberse obtenido a partir de las imágenes. En este sentido, es posible que, modificando la selección de parches para esta técnica, se puedan mejorar los puntos débiles observados en ella, principalmente el bajo TPR observado debido a la poca cantidad de parches utilizados y los problemas observados en la segmentación automática de materia blanca.

Por otra parte, la segmentación obtenida mediante la detección de *outliers* en el GMM presenta algunas ventajas con respecto al uso de CNN. Por un lado, la predicción de la CNN depende en gran parte de la selección inicial de candidatos a lesión. La implementación considerando solamente parches centrados en vóxeles pertenecientes a la materia blanca muestra problemas si es que la segmentación de materia blanca posee errores debido a la presencia de irregularidades

demasiado pronunciadas en la imagen. Este problema no se observó en el GMM, debido a que dichas irregularidades tienden a presentarse como *outliers* en el modelo. Además, debido a la poca cantidad de parámetros en el GMM presentado, la ejecución del algoritmo EM es bastante rápida, permitiendo obtener la segmentación de las lesiones en muy poco tiempo. Cabe destacar que el uso de la GPU logra disminuir considerablemente el tiempo de entrenamiento de las CNN, llevándolo a un nivel comparable con el tiempo requerido para obtener una segmentación mediante el GMM en la CPU.

Finalmente, la adición de una etapa de postprocesamiento a los resultados de las segmentaciones, consistente en la aplicación de un filtro de cierre, logró mejorar los resultados de estas de manera considerable, logrando mitigar el problema de la subestimación del tamaño de las lesiones observado en ambos métodos.

7 | Trabajo Futuro

Los métodos desarrollados lograron obtener resultados cuya evaluación se encuentra dentro del rango obtenido en el Estado del Arte para la base de datos, sin embargo es posible trabajar en posibles mejoras para obtener una mejor calidad en la segmentación.

Si bien la aplicación de reglas de detección de lesiones a la selección de vóxeles outliers del GMM permite en cierto modo verificar ciertas propiedades espaciales de las lesiones, el método no hace uso de información de la vecindad de los vóxeles, lo cual presenta una desventaja para la segmentación al no siempre lograr obtener una región contigua para una misma lesión. Una alternativa de solución podría ser la adición de características locales de cada vóxel al GMM. Utilizando como entrada características tales como la diferencia entre la intensidad del vóxel y sus vecinos en adición a las intensidades en cada modalidad de imagen, sería posible captar información local que podría ayudar a la segmentación.

En el caso de las redes convolucionales existen muchas posibles mejoras. Por un lado, el submuestreo de vóxeles candidatos tiende a eliminar un porcentaje no despreciable de lesiones que no se están utilizando para entrenar el modelo. Una posible mejora podría ser el combinar técnicas de detección de *outliers*, dentro de las distintas modalidades de la MRI, con la etapa de muestreo de vóxeles candidatos.

Las redes neuronales, además, requieren de una gran cantidad de datos para poder generalizar bien en casos de prueba. Con acceso a una mayor cantidad de imágenes de distintos pacientes, es posible que la calidad de los resultados de las redes mejore.

Por último, las redes neuronales presentadas solamente funcionan para MRI con la misma resolución con la cual fueron entrenadas. Como solución, deberían estudiarse alternativas para añadir a una etapa de preprocesamiento de los datos, de manera que la imagen de entrada tenga la misma resolución que las de entrenamiento.

Bibliografía

- [1] Keras: Deep learning for python. <https://github.com/fchollet/keras>. Accessed: 2017-11-10. 4.3
- [2] Tensorflow. <https://www.tensorflow.org/>. Accessed: 2017-11-10. 4.3
- [3] Renjie He Meghana Mehta Rakesh K. Gupta Jerry S. Wolinsky Balasrinivasa Rao Sajja, Sushmita Datta and Ponnada A. Narayana. Unified approach for multiple sclerosis lesion segmentation on brain mri. *Annals of Biomedical Engineering*, 34:142–151, 2006. 4.2
- [4] Ariel Birenbaum and Hayit Greenspan. Longitudinal multiple sclerosis lesion segmentation using multi-view convolutional neural networks. In Carneiro G. et al., editor, *Deep Learning and Data Labeling for Medical Applications. LABELS 2016, DLMIA 2016. Lecture Notes in Computer Science*,, volume 10008, pages 58–67. Springer, 2016. 2.6.2, 4.3, 4.3
- [5] Sushmita Datta and Ponnada A. Narayana. A comprehensive approach to the segmentation of multichannel three-dimensional mr brain images in multiple sclerosis. *Neuroimage: Clinical*, 2:184–196, 2013. 4.2
- [6] Sonika Dhankhar, Shobha Tyagi, and T Prasad. Brain mri segmentation using k-means algorithm, 03 2010. 4.2.1
- [7] Olivier Clatz Ender Konukoglu Antonio Criminisi Ezequiel Geremia, Bjoern H. Menze and Nicholas Ayache. Spatial decision forests for ms lesion segmentation in multi-channel mr images. In T. Jiang et al., editor, *MICCAI 2010, Part I*, volume LNCS 6361, pages 111–118, Beijing, China, September 2010. Springer. ??
- [8] Bart Goossens Ivana Despotović and Wilfried Philips. Mri segmentation of the human brain: Challenges, methods, and applications. *Computational and Mathematical Methods in Medicine*, 2015:1–23, 2014. 1
- [9] N. Ayache G. Malandain JC. Souplet, C. Lebrun. An automatic segmentation of t2-flair multiple sclerosis lesions. 2008. 4.2, ??
- [10] Alireza Karimian and Simin Jafari. A new method to segment the multiple sclerosis lesions on brain magnetic resonance images. *Journal of Medical Signals and Sensors*, 5:238–244, 2015. 2.5.1, 4.2
- [11] Sean P. Morrissey Lorenzo, Sylvain Prima and Christian Barillot. A robust expectation-maximization algorithm for multiple sclerosis lesion segmentation. 2008. 4.2

- [12] T. Heskes I.W.M. van Uden F.E. de Leeuw E. Marchiori B. van Ginneken M. Ghafoorian, N. Karssemeijer and B. Platel. Non-uniform patch sampling with deep convolutional neural networks for white matter hyperintensity segmentation. In *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pages 1414–1417, Prague, Czech Republic, 2016. IEEE. ([document](#)), 2.6.1, 2.9
- [13] Xavier Lladó Jordi Freixenet Meritxell Bach Cuadra Mariano Cabezas, Arnau Oliver. A review of atlas-based segmentation for magnetic resonance brain images. *Elsevier Health*, 104:e158–e177, 2011. 2.4
- [14] Bryan S. Morse. Lecture 4: Thresholding. http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/threshold.pdf, 2000. Accessed: 2016-08-10. 2.3
- [15] Philip A. Cook Yuanjie Zheng Alexander Egan Paul A. Yushkevich Nicholas J. Tustison, Brian B. Avants and James C. Gee. N4itk: Improved n3 bias correction. *IEEE Transactions on Medical Imaging*, 29:1310–1320, 2010. 4.1.1
- [16] Daniel Rueckert Nick Weiss and Anil Rao. Multiple sclerosis lesion segmentation using dictionary learning and sparse coding. In K. Mori et al., editor, *MICCAI 2013, Part I*, volume LNCS 8149, pages 735–742, Nagoya, Japan, September 2013. Springer. ??
- [17] David C Preston. Magnetic resonance imaging (mri) of the brain and spine: Basics. <http://casemed.case.edu/clerkships/neurology/Web%20Neurorad/MRI%20Basics.htm>, 2006. Accessed: 2016-08-10. 2.2
- [18] Jacob Goldberger Roey Mechrez and Hayit Greenspan. Patch-based segmentation with spatial consistency: Application to ms lesions in brain mri. *International Journal of Biomedical Imaging*, 2016, 2016. ??