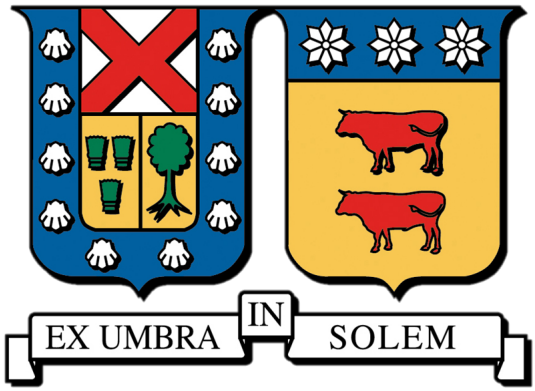
Universidad Técnica Federico Santa María
Departamento de Informática
Valparaíso, Chile



Mahalanobis Distance Loss: Nueva función de pérdida basada en mapas de distancia texturales para la tarea de segmentación de lesiones de esclerosis múltiple

Gustavo Jorge Ulloa Poblete

Tesis para optar al grado de

DOCTOR EN INGENIERÍA INFORMÁTICA

Director de Tesis:
Dr. Héctor Allende Olivares
Co-Director de Tesis:
Dr. Alejandro Veloz Baeza

Enero 2026.

CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☐ Memoria o trabajo de título; ☒ Tesis de Postgrado;

Título del trabajo: Mahalanobis Distance Loss: Nueva función de pérdida basada en mapas de distancia texturales para la tarea de segmentación de lesiones de esclerosis múltiple

Nombre del candidato(a): Gustavo Jorge Ulloa Poblete

Carrera / Grado: Doctorado en Ingeniería Informática / Doctor en Ingeniería Informática

Campus: Casa Central Valparaiso ; **Departamento**: Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Héctor Manuel Allende Olivares, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

☒ El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

☐ 6 meses; ☐ 12 meses; ☐ 2 años; ☐ 3 años; ☐ 5 años; ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 14/01/2026 ; **Firma:** _____

Estudiante o Candidato(a):

Fecha: 14/01/2026 ; **Firma:** _____

COMITÉ EVALUADOR

Dr. Héctor Allende Olivares
Universidad Técnica Federico Santa María, Chile

Director de Tesis

Dr. Alejandro Veloz Baeza
Universidad de Valparaíso, Chile

Co-Director de Tesis

Dra. Pamela Guevara Álvez
Universidad de Concepción, Chile

Evaluadora Externa Nacional

Dr. Claudio Moraga Roco
Technische Universität Dortmund, Alemania

Evaluador Externo Internacional

Dra. Raquel Pezoa Rivera
Universidad Técnica Federico Santa María, Chile

Evaluadora Interna y Presidente Comisión Examen

Agradecimientos

Agradezco a todas las personas que me han apoyado en estos años: a mi familia, compañeros y amigos; también a la universidad por confiar en mí, dándome la oportunidad de desarrollarme como profesional. Este trabajo ha sido apoyado por los proyectos ANID CCTVal CIA250027 y DGIIP-UTFSM PI-LIR23-13.

Agradezco a mi profesor guía, Dr. Héctor Allende, por su incentivo y apoyo técnico y humano. A mi profesor co-guía Dr. Alejandro Veloz y al profesor Dr. Claudio Moraga, quienes me brindaron su apoyo y ayuda. También agradezco al partner Sebastián Sánchez por su ayuda y disposición. Agradezco a Pabla Valdebenito, quien siempre nos ayudó con gran calidez humana a todos los alumnos de postgrado.

Agradezco a mi hermana Angélica y a mi hermano Pablo quienes me apoyaron e incentivaron. A mi sobrina Natalia, a Paulita y Munay, a quienes espero poder dedicarles un poco más de tiempo. También agradezco a mi tía madrina Silvia Poblete, muchas gracias por apoyarme siempre y darme cariño; como también a mi tío Antonio y a mis primos Emanuel y María Isabel. Agradezco a mi tía Beatriz quien también ha sido muy buena conmigo.

Agradezco de manera especial a mi papá Jorge Ulloa Martínez, quien siempre a mí, a mi hermano y a mi hermana nos mostró y habló de lo importante que es el conocimiento y lo entretenidas que son la tecnología y la ciencia, y también lo importante que es el esfuerzo y la constancia.

Quiero agradecer de la manera más especial que pueda existir a mi querida mamá, María Angélica Poblete Caballero, la mejor mamá que hubiera podido tener, te echo mucho de menos y siempre estás conmigo aunque no hable mucho de ti con otras personas, es que aún me cuesta mucho. Espero seguir desarrollándome y continuar por el camino de la academia y docencia, como tú, quien fuiste una profesora normalista muy preparada y muy dulce con todos.

Quiero agradecer a todas mis amigas y amigos: Luis, Tania, Karina, Victor y Paulina (en el cielo), como también a las personas que no alcanzo a mencionar que me tienen estima, a quienes no he podido dedicarles mucho tiempo.

Le agradezco a Dios y a la Virgen por estar siempre conmigo y ayudarme a tomar las decisiones correctas.

A mi mamá María Angélica Poblete Caballero y a mi papá Jorge Ulloa Martínez.

Resumen

La segmentación automática de lesiones de esclerosis múltiple en imágenes de resonancia magnética es una tarea fundamental para el diagnóstico, el monitoreo de la enfermedad y la evaluación de tratamientos. No obstante, la presencia del efecto del volumen parcial, el solapamiento de intensidades entre tejidos y el alto desbalance de clases, dificultan la segmentación de las lesiones de EM.

En esta tesis se propone una nueva función de pérdida, denominada Mahalanobis Distance Loss (MDL), basada en el Mapa de Distancias de Mahalanobis (MDM), también propuesto, que integra información espacial y textural mediante características radiómicas extraídas de la modalidad FLAIR. A diferencia de los mapas de distancia construidos con la distancia euclidiana, el MDM incorpora dependencias estadísticas entre características, capturando mejor las variaciones sutiles en regiones ambiguas cercanas a los bordes de las lesiones.

La MDL es combinada con la Generalized Dice Loss mediante un parámetro ϵ que regula el equilibrio entre solapamiento global y precisión en los bordes. La evaluación en los conjuntos de datos públicos ISBI-MS y MSSEG2016, utilizando una U-Net, demuestra que la función de pérdida propuesta supera a las basadas en mapas de distancia euclidiana, como Boundary Loss y Hausdorff Loss, en métricas de solapamiento (Dice, precisión), de borde (HD95, ASSD) y de detección bajo desbalance de clases (AUC-PR), además de reducir los falsos positivos.

Los resultados validan que incorporar información de textura en la función de pérdida mediante el MDM mejora la segmentación automática de lesiones de EM y ofrece un marco prometedor para generalizar estas ideas a otros tipos de lesiones, tejidos y órganos.

Palabras clave: Segmentación de imágenes, Esclerosis múltiple, Redes neuronales convolucionales, Función de pérdida, Mapa de distancia, Características radiómicas

Abstract

Automatic segmentation of multiple sclerosis (MS) lesions in magnetic resonance imaging (MRI) is a fundamental task for diagnosis, disease monitoring, and treatment evaluation. However, partial volume effects, intensity overlap between tissues, and severe class imbalance make MS lesion segmentation particularly challenging.

This thesis introduces a new loss function, termed Mahalanobis Distance Loss (MDL), built upon a newly proposed Mahalanobis Distance Map (MDM) that integrates spatial and textural information through radiomic features extracted from the FLAIR modality. Unlike traditional distance maps constructed using Euclidean distance, the MDM incorporates statistical dependencies between features, enabling the capture of subtle variations in ambiguous regions near lesion boundaries.

The MDL combines the MDM with the Generalized Dice Loss through a parameter ϵ that balances global overlap and boundary precision. Evaluation on the public ISBI-MS and MSSEG2016 datasets, using a U-Net architecture, demonstrates that the proposed loss function outperforms Euclidean-based distance losses such as Boundary Loss and Hausdorff Loss in boundary metrics (HD95, ASSD), overlap metrics (Dice, Precision), and detection under class imbalance (AUC-PR), while also reducing false positives.

Overall, the results validate that incorporating texture information into the loss function via the MDM improves automatic MS lesion segmentation, offering a promising framework to generalize these ideas to other types of lesions, tissues, and organs.

Keywords: Image segmentation, Multiple sclerosis, Convolutional neural networks, Loss function, Distance map, Radiomics features

Lista de Abreviaturas

ABL	Active Boundary Loss
Adam	Adaptive Moment Estimation
ASSD	Distancia Simétrica Promedio de Superficie (Average Symmetric Surface Distance)
AUC-PR	Área Bajo la Curva de Precisión-Recall
BCE	Entropía Cruzada Binaria (Binary Cross-Entropy)
CBL	Conditional Boundary Loss
CNN	Red Neuronal Convolucional (Convolutional Neural Network)
CT	Tomografía Computarizada (Computed Tomography)
DNN	Red Neuronal Profunda (Deep Neural Network)
DTM	Mapa de Transformación de Distancia (Distance Transform Map)
EM	Esclerosis Múltiple
FCN	Red Neuronal Completamente Convolucional (Fully Convolutional Network)
FLAIR	Fluid-Attenuated Inversion Recovery
FN	Falsos Negativos (False Negatives)
FP	Falsos Positivos (False Positives)
GDice	Generalized Dice Loss
GLCM	Matriz de Co-ocurrencia de Niveles de Gris (Gray-Level Co-occurrence Matrix)
GLN	Gray Level Non-Uniformity
GLRLM	Matriz de Longitud de Rachas de Niveles de Gris (Gray-Level Run-Length Matrix)

HD	Distancia de Hausdorff (Hausdorff Distance)
HD95	Percentil 95 de la distancia de Hausdorff
ISBI	International Symposium on Biomedical Imaging
ISBI-MS	Conjunto de datos del ISBI de pacientes con esclerosis múltiple
KL	Divergencia de Kullback-Leibler
LCR	Líquido Cefalorraquídeo
LRE	Long Run Emphasis
MB	Materia Blanca
MDL	Mahalanobis Distance Loss
MDM	Mapa de Distancia de Mahalanobis
MG	Materia Gris
MICCAI	Medical Image Computing and Computer Assisted Intervention
MNI	Instituto Neurológico de Montreal (Montreal Neurological Institute)
MRI	Imágenes de Resonancia Magnética (Magnetic Resonance Imaging)
MSSEG	Multiple Sclerosis Lesion Segmentation Challenge
MSSEG2016	Conjunto de datos del MSSEG de pacientes con esclerosis
PCA	Análisis de Componentes Principales (Principal Component Analysis)
PD	Densidad de Protones (Proton Density)
PET	Tomografía por Emisión de Positrones (Positron Emission Tomography)
PPV	Valor Predictivo Positivo (Positive Predictive Value)
ReLU	Unidad Lineal Rectificada (Rectified Linear Unit)
RLN	Run Length Non-Uniformity
ROI	Región de Interés (Region of Interest)
RP	Run Percentage
RVD	Diferencia Relativa de Volumen (Relative Volume Difference)
SDF	Mapa de Distancia con Signo (Signed Distance Function)
SGD	Gradiente Descendente Estocástico (Stochastic Gradient Descent)

SNC	Sistema Nervioso Central
SRE	Short Run Emphasis
SVD	Descomposición en Valores Singulares (Singular Value Decomposition)
TE	Tiempo de Eco (Echo Time)
TN	Verdaderos Negativos (True Negatives)
TNR	Tasa de Verdaderos Negativos (True Negative Rate)
TP	Verdaderos Positivos (True Positives)
TPR	Tasa de Verdaderos Positivos (True Positive Rate)
TR	Tiempo de Repetición (Repetition Time)
WBCE	Weighted Binary Cross-Entropy

Notación

$\boldsymbol{x}$	Notación para vectores
ϵ	Parámetro de balance en funciones de pérdida combinadas
λ	Parámetro de ponderación en la Mahalanobis Distance Loss
$\mathbb{E}$	Valor esperado o esperanza matemática
$\mathbb{R}$	Conjunto de los números reales
∇	Operador gradiente (vector de derivadas parciales)
Ω	Dominio discreto de la imagen (conjunto de coordenadas)
Σ	Matriz de covarianza
Σ_X	Matriz de covarianza de variable X
Θ / Σ^{-1}	Matriz de precisión (inversa de la covarianza)
$\|\cdot\|$	Norma de un vector (usualmente norma Euclidiana o L_2)
^{T}	Operador de transposición
G	Máscara de segmentación manual (Ground Truth)
$G_{\mathbf{DTM}}$	Mapa de Transformación de Distancia del Ground Truth
$G_{\mathbf{MDM}}$	Mapa de distancia de Mahalanobis del Ground Truth
$G_{\mathbf{SDF}}$	Mapa de Distancia con Signo del Ground Truth
GB	Conjunto de vóxeles pertenecientes al borde del Ground Truth
P	Mapa de probabilidad binarizado (Predicción)
$P_{\mathbf{DTM}}$	Mapa de Transformación de Distancia de la predicción P
PB	Conjunto de vóxeles pertenecientes al borde de la predicción P
s_θ	Mapa de probabilidad continuo predicho por la red neuronal
$\|\cdot\|$	Valor absoluto (escalares) o cardinalidad (conjuntos)
$\mathcal{A}$	Algoritmo de aprendizaje
$\boldsymbol{\Delta}$	Matriz diagonal de valores singulares

$(\mathcal{Z}, \mathcal{G}, \mathbb{P})$	Espacio de probabilidad
$\mathcal{F}$	Espacio de hipótesis
F	Mapa de características radiómicas y espaciales (Propuesta MDM)
$\mathcal{G}$	σ-álgebra de subconjuntos del espacio muestral
$L_{\mathbf{ABL}}$	Función de pérdida Active Boundary Loss
$L_{\mathbf{BCE}}$	Función de pérdida Binary Cross-Entropy
$L_{\mathbf{BL}}$	Función de pérdida Boundary Loss
$L_{\mathbf{BS}}$	Función de pérdida Boundary-Sensitive Loss
$L_{\mathbf{CBL}}$	Función de pérdida Conditional Boundary Loss
$L_{\mathbf{Dice}}$	Función de pérdida Dice
$L_{\mathbf{GDice}}$	Función de pérdida Generalized Dice Loss
$L_{\mathbf{HD}}$	Función de pérdida Hausdorff Distance Loss
L	Función de pérdida (general)
$L_{\mathbf{MDL}}$	Función de pérdida Mahalanobis Distance Loss (Propuesta)
$\mathbb{P}$	Medida de probabilidad (Teoría del Aprendizaje Estadístico)
$\mathcal{R}_{emp}(f)$	Riesgo empírico de $f \in \mathcal{F}$
$\mathcal{R}(f)$	Riesgo funcional o riesgo esperado de $f \in \mathcal{F}$
$\boldsymbol{X}$	Notación para matrices
$\mathcal{Z}$	Espacio muestral

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Contexto y descripción del problema

La segmentación de imágenes se ha consolidado como una etapa fundamental en el análisis de imágenes médicas, ya que permite separar o delimitar regiones anatómicas, identificar tipos de tejidos y determinar zonas de interés clínico. Desde el punto de vista clínico, estas segmentaciones facilitan la identificación de lesiones, la estimación de volúmenes y la cuantificación de estructuras, lo que contribuye directamente al diagnóstico, seguimiento y apoyo en la toma de decisiones terapéuticas (Xia et al., 2025; Zhou et al., 2019). Si bien durante años la segmentación automática se abordó mediante modelos clásicos, ya sea basados en umbrales (Otsu, 1979), en crecimiento de regiones (Nock and Nielsen, 2004), clustering con k-means (Dhanachandra et al., 2015), en algoritmos más avanzados tales como contornos activos (Kass et al., 1988) y campos aleatorios de Markov (Plath et al., 2009). Sin embargo, los avances recientes en aprendizaje profundo, y en particular en redes convolucionales, han cambiado la forma tradicional de abordar esta tarea. Donde la arquitectura U-Net o encoder-decoder, se ha convertido en una arquitectura muy influyente, con numerosas variantes desarrolladas para distintos tipos de imágenes y dominios, incluyendo de manera destacada la segmentación de imágenes médicas (Minaee et al., 2022).

Una de las aplicaciones clínicas más críticas corresponde a la segmentación de lesiones de esclerosis múltiple (EM). La EM es una enfermedad crónica autoinmune del sistema nervioso central caracterizada por la aparición de lesiones en la materia blanca, las cuales afectan la

vaina de mielina y los axones, produciendo un deterioro neurológico progresivo (Dobson and Giovannoni, 2019). Mediante las imágenes de resonancia magnética (MRI) es posible detectar y estudiar estas lesiones. La cuantificación del volumen y del número de lesiones es un paso esencial para el diagnóstico, la planificación terapéutica y el desarrollo de nuevas estrategias de intervención (Giorgio and Stefano, 2018; Zhou et al., 2019).

A pesar de estos avances, la segmentación de lesiones de EM, continua presentando desafios debido a la presencia de varios factores como: alto desbalance de clases que se manifiesta en la desproporción entre la cantidad de vóxeles pertenecientes a los objetos de la clase de interés (foreground) y los de la clase fondo (background), efecto del volumen parcial en los bordes de los objetos, heterogeneidad de la intensidad de los tejidos y solapamiento de la distribución de intensidad de los diferentes tejidos, órganos y lesiones (Danelakis et al., 2018; Naga Karthik et al., 2025; Zhou et al., 2019). Estas dificultades afectan la precisión en la segmentación de modelos, aumentando la frecuencia de falsos positivos y falsos negativos.

Para abordar estos problemas, en los últimos años se han propuesto diversas funciones de pérdida que incorporan mapas de transformación de distancias (como Boundary Loss y Hausdorff Loss) con el objetivo de mejorar el solapamiento global y la calidad en los bordes de los objetos segmentados (Karimi and Salcudean, 2020; Kervadec et al., 2021; Ma et al., 2020). Si bien estas propuestas han obtenido mejores resultados en métricas como Dice y la distancia de Hausdorff, aún presentan limitaciones en la delineación precisa de regiones con bordes ambiguos, particularmente en áreas donde existe solapamiento significativo en la distribución de intensidades de los tejidos. La limitación principal de estos métodos reside en su dependencia exclusiva de la distancia euclidiana espacial (utilizando únicamente las coordenadas geométricas de los píxeles/vóxeles) para la construcción de los mapas de distancia (DTM y SDF). Esta restricción implica que los modelos no incorporan información textural ni las dependencias estadísticas entre las características de los vóxeles. Al ignorar la covarianza entre las características texturales (radiómicas) y espaciales, estas pérdidas son insensibles a las variaciones sutiles de textura que se manifiestan en los bordes de las lesiones, especialmente en zonas donde la distinción entre tejido sano y lesionado es menos evidente (Karimi and Salcudean, 2020; Kervadec et al., 2021; Ma et al., 2020; Wang et al., 2021).

1.2. Hipótesis

Una función de pérdida que, además de la distancia euclidiana espacial, integre información de textura mediante una distancia de Mahalanobis entre cada vóxel y el vóxel borde más cercano, y que utilice esta distancia para penalizar los errores, mejorará la segmentación de lesiones de EM tanto en métricas de solapamiento espacial como en métricas basadas en distancia a los bordes, en comparación con las actuales funciones de pérdida basadas únicamente en mapas de distancia euclidiana espacial.

1.3. Objetivos

El objetivo general consiste en desarrollar una función de pérdida basada en un mapa de distancia que además de la distancia euclidiana al vóxel borde más cercano, integre características locales texturales mediante la distancia de Mahalanobis.

Los objetivos específicos son los siguientes:

1. Diseñar un mapa de distancia basado en la distancia de Mahalanobis, que utilice características de coordenadas espaciales, intensidad y de textura entre cada vóxel y el vóxel borde más cercano de la clase opuesta.

2. Construir una función de pérdida que utilice el mapa de distancia propuesto para ponderar los errores del modelo, modulando dicha ponderación mediante un parámetro λ.

3. Implementar y optimizar el método propuesto en el modelo de segmentación de tipo encoder-decoder U-Net, donde se evalúen distintos valores del parámetro λ.

4. Evaluar el desempeño del método en los conjuntos de datos ISBI-MS y MSSEG2016 correspondientes a MRI de pacientes con esclerosis múltiple, mediante métricas de solapamiento espacial y distancia a los bordes.

5. Comparar el desempeño de la función de pérdida propuesta con funciones de pérdida para la segmentación de imágenes médicas, especificamente aquellas basadas en regiones y distancia a los bordes.

Con el objetivo de facilitar y fomentar la reproducibilidad de la investigación, todo el código de la implementación del método se encuentra publicado y de libre acceso en el repositorio de GitHub `https://github.com/GustavoUlloaPoblete/MSLesionSegmentation`.

1.4. Organización de la tesis

En el capítulo 2, Marco Teórico, se abordan los fundamentos necesarios para desarrollar la investigación junto con el estado del arte de funciones de pérdida utilizadas en la tarea de segmentación de imágenes médicas.

En el capítulo 3 se presenta la propuesta del trabajo tesis. En primer lugar, se presenta el algoritmo para la generación del mapa de distancia propuesto, denominado Mapa de Distancia de Mahalanobis (MDM), y a continuación se presenta la función de pérdida propuesta, Mahalanobis Distance Loss (MDL).

A continuación, en el capítulo 4 se presentan y discuten los resultados de la propuesta aplicada en dos conjuntos de datos públicos de imágenes de resonancia magnética de pacientes con esclerosis múltiple.

Finalmente, en el capítulo 5 se pueden encontrar las conclusiones generales, los principales aportes y el trabajo futuro.

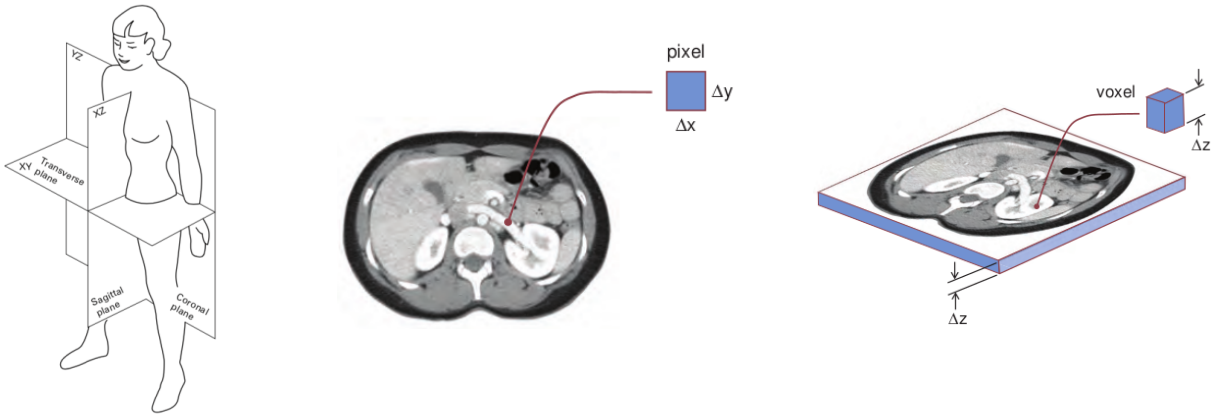
Capítulo 2

Marco Teórico

2.1. Imágenes médicas

Una imagen médica digital corresponde a una representación visual del interior del cuerpo humano, obtenida mediante diferentes tipos de tecnologías con diversos fines, como el diagnóstico de enfermedades, la planificación de terapias, el seguimiento de tratamientos y para la investigación. Es posible definir una imagen como una función $I : \Omega \rightarrow \mathbb{R}^D$ (Frery and Perciano, 2013). Aquí, Ω es el dominio discreto de la imagen, donde $p \in \Omega \subset \mathbb{N}^{2,3}$ representa cada vector, elemento o coordenada espacial. Específicamente, para imágenes 2D, $\Omega = \{p = (i, j) | 1 \leq i \leq H, 1 \leq j \leq W\}$, donde H y W corresponden al alto y ancho, respectivamente. Cada coordenada p es mapeada al vector $I(p) \in \mathbb{R}^D$, siendo D la cantidad de canales de la imagen. Para el caso de imágenes a color-RGB se tienen $D = 3$ canales correspondientes a rojo (R), verde (G) y azul (B). Esta representación se materializa como una o varias matrices, cuyos elementos reciben nombres de acuerdo a su dimensionalidad. Para matrices 2D, estos elementos reciben el nombre de píxeles (picture elements), en cambio, para el caso de matrices 3D, los elementos reciben el nombre de vóxeles (volume elements). Un vóxel corresponde al par $(p, I(p))$, es decir, es un elemento de una matriz que contiene información de ubicación espacial y de intensidad de gris. Para el caso de las imágenes 3D, es posible obtener vistas 2D o slices en tres orientaciones: (1) transversal o axial, (2) sagital, y (3) coronal, como se ilustra en la Figura 2.1(a). La magnitud o intensidad de gris de los píxeles y vóxeles representa un área y un volumen respectivamente. Las dimensiones de los

píxeles y vóxeles pueden ser iguales (isotrópico) o diferentes (anisotrópico), como se observa en las Figuras 2.1(b) y (c). El concepto de resolución espacial de una imagen se define como el tamaño del objeto distinguible más pequeño, o la mínima separación que permite diferenciar entre objetos diferentes (Dougherty, 2009).



(a) **(b)** **(c)**

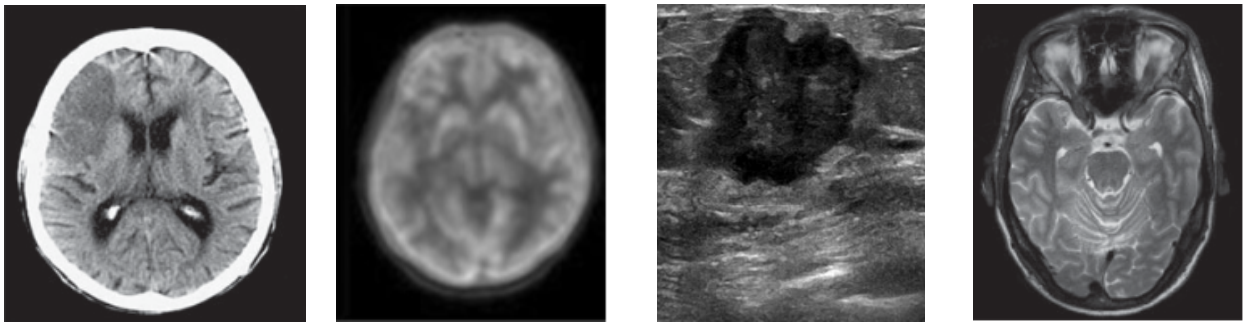
Figura 2.1: Representación espacial y resolución en imágenes médicas. (a) Planos anatómicos de referencia (axial, sagital y coronal) (Dougherty, 2009). (b) Píxel y (c) vóxel (Bushberg et al., 2012).

Las imágenes médicas se obtienen a través de diversas modalidades de adquisición (Bushberg et al., 2012). Estas modalidades se clasifican principalmente según el tipo de radiación que emplean: ionizante o no ionizante.

La radiación ionizante comprende ondas electromagnéticas de alta energía capaces de ionizar átomos y moléculas al remover electrones. Ejemplos comunes en imágenes médicas incluyen los rayos X, utilizados en la tomografía computarizada (CT), y los rayos gamma, empleados en la medicina nuclear (PET/SPECT).

Por otro lado, las radiaciones no ionizantes abarcan las ondas de ultrasonido, fundamentales en la ecografía, y las ondas electromagnéticas en la banda de radiofrecuencias que en combinación con un potente campo magnético, se utilizan en la resonancia magnética (MRI). La Figura 2.2 ilustra ejemplos de estas modalidades mediante imágenes 2D, también conocidas como slices.

Las modalidades de imágenes médicas como la tomografía computarizada (CT) y resonancia magnética nuclear (MRI) se han convertido en las de mayor uso en medicina, principalmente por su capacidad de ofrecer una alta resolución en tres dimensiones.



(a) **(b)** **(c)** **(d)**

Figura 2.2: Principales modalidades de imágenes médicas: (a) CT, (b) PET, (c) Ecografía y (d) MRI.

2.1.1. Imágenes de resonancia magnética nuclear

La resonancia magnética (MR, por sus siglas en inglés) es una técnica de imagenología no ionizante que emplea radiofrecuencia y campos magnéticos de gran intensidad, típicamente entre 1.5 y 3 Tesla (T). Para dar una perspectiva de las magnitudes, esto es aproximadamente $60,000$ veces superior al campo magnético terrestre (Smith and Webb, 2011).

Los núcleos atómicos están compuestos por nucleones, los que a su vez los están por neutrones y protones. Aquellos núcleos con una cantidad impar de nucleones se comportan como pequeños imanes, cada uno con un momento magnético asociado. Las imágenes de resonancia magnética (MRI) se generan principalmente a partir de las señales de los átomos de hidrógeno. Estos átomos, al estar formados por un solo protón y ser altamente abundantes en los tejidos biológicos, principalmente en las moléculas de agua, son la base para la señal detectada por los escáneres de resonancia magnética actuales (Dougherty, 2009). Desde una perspectiva atómica, el protón es una partícula cargada que rota sobre su propio eje, poseyendo un momento angular (P) y un momento magnético (μ) (Figura 2.3(a)) (Smith and Webb, 2011).

En estado natural o de equilibrio, la orientación de los espines de los protones están dispuestas de manera aleatoria (Figura 2.3(b)), de este modo, la magnitud de la suma vectorial de todos los momentos magnéticos individuales en cada lugar del cuerpo es aproximadamente cero. Sin embargo, al aplicar un campo magnético externo B_0, los protones se alinean a este campo magnético en sólo dos valores discretos de ángulos de $\theta = 54.7$ con respecto a la

dirección de B_0, es decir en la misma dirección o en la dirección opuesta (llamadas paralela y antiparalela). En la Figura 2.3(c) se representan las alineaciones, donde la cantidad de protones alineados en la dirección paralela es mayor debido a que esta requiere una menor cantidad de energía. Los protones alineados al campo B_0 también presentan el fenómeno de precesión (Figura 2.3(d)). Este movimiento consiste en un giro alrededor del eje B_0 a una frecuencia proporcional a la magnitud de este campo magnético, llamada frecuencia de Larmor. La red de magnetización resultante de la interacción de todos los protones presenta solo una componente longitudinal M_Z, paralela a B_0 (Figura 2.3(e)). Esto se debe a que la mayor cantidad de alineaciones paralelas respecto a la antiparalela produce una señal neta en esa dirección, mientras que las componentes transversales M_X y M_Y se anulan mutuamente debido a la distribución aleatoria de las fases de precesión.

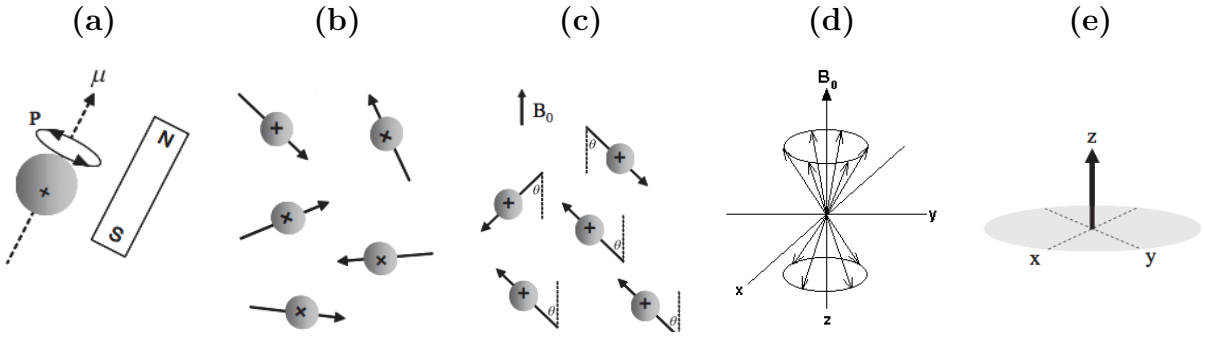


Figura 2.3: Principios físicos de la magnetización nuclear. (a) Momento angular P y magnético μ de un protón. (b) Orientación aleatoria de los momentos magnéticos. (c) Alineación paralela y antiparalela. (d) Distribución aleatoria de vectores de magnetización y (e) Magnetización neta resultante.

Cuando un pulso de radiofrecuencia B_1 es aplicado a la frecuencia de Larmor (Figura 2.4), cierto número de protones adquirirán energía suficiente para que sus espines pasen de la alineación paralela a antiparalela, lo cual implicará una disminución en la magnitud de la magnetización longitudinal M_Z (Smith and Webb, 2011). Este fenómeno es conocido como resonancia magnética. Otro efecto que ocurre con la presencia de B_1 es la coherencia de fases entre los spin de acuerdo a su configuración de giro paralelo o antiparalelo, lo cual evitará la cancelación de las componentes transversales M_X y M_Y. El tiempo de aplicación y la intensidad del pulso B_1 es proporcional a la cantidad de alineaciones antiparalelas y por

tanto al ángulo de inclinación de la magnetización neta. Un pulso más largo o más intenso puede inclinar la magnetización neta de M_Z completamente hacia el plano transversal (un pulso de 90) o incluso invertirla (180), modificando así la proporción de espines en alineación paralela y antiparalela.

En el proceso de detección en imágenes de resonancia magnética (MRI), la energía que se detecta es la liberada desde los protones en el proceso de relajación de los momentos magnéticos inducidos, es decir cuando el pulso de radiofrecuencias B_1 es desactivado y por tanto la magnetización neta comienza a retornar a su estado de equilibrio referenciado por el campo magnético B_0. La amplitud y la duración de esta señal detectada dependen de tres parámetros inherentes a los tejidos (Balafar et al., 2010):

- Densidad de Protones (PD): Representa la concentración de protones de hidrógeno en el tejido. A mayor densidad, mayor es el número de espines disponibles para contribuir a la señal.

- Tiempo de relajación spin-lattice (T_1): corresponde al tiempo de relajación de la componente longitudinal M_Z, es decir, en alinearse nuevamente con B_0.

- Tiempo de relajación spin-spin (T_2): corresponde al tiempo que tarda la magnetización transversal M_{XY}, o uno de los componente M_X o M_Y, en decaer debido a la pérdida de coherencia de fase.

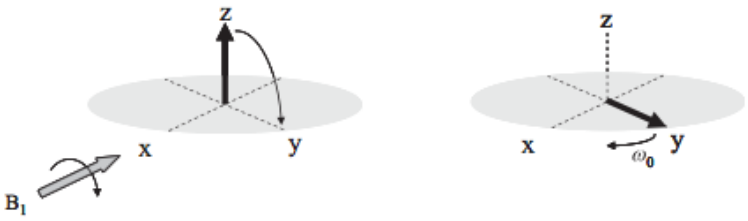


Figura 2.4: Efecto del campo B_1 sobre la magnetización. (Izquierda) Magnetización neta alineada con B_0 en el eje z. (Derecha) Magnetización rotada 90° al plano transversal (ejes x e y).

Mediante la aplicación de secuencias de pulsos de excitación B_1 como spin-echo, gradient-echo e inversion recovery, es posible obtener distintos tipos de secuencias de MRI como T1-weighted, T2-weighted, PD-weighted y FLAIR (Fluid-Attenuated Inversion Recovery).

Dependiendo de parámetros ajustables por el operador del resonador magnético como el tiempo de repetición (TR) de la secuencia de pulsos y el tiempo de eco (TE) (tiempo de muestreo de la señal), es posible formar cada una de los tipos de imágenes recién mencionados. Estos parámetros estan relacionados con el grado de contraste en la imagen que puede ser obtenido entre los tejidos cerebrales.

De acuerdo a (Romo-Sanchez M, 2020) es posible relacionar las distintas secuencias o canales de MR de acuerdo a la intensidad de señal dominante que exhiben en diferentes tejidos. En la tabla 2.1 se presenta un resumen de la intensidad de señal que presentan tejidos como el líquido cefalorraquideo (LCR), la materia blanca (MB), la materia gris (MG), la grasa y la inflamación, respecto a las secuencias T1-weighted, T2-weighted y FLAIR.

Tabla 2.1: Nivel de intensidad de señal de canales de MRI en diferentes tejidos.

Tejido	**T1-weighted**	**T2-weighted**	**FLAIR**
LCR	Hipointenso	Hiperintenso	Hiperintenso
MB	Hiperintenso leve	Hipointenso leve	Hipointenso leve
MG	Hipointenso leve	Hiperintenso leve	Hiperintenso leve
Grasa	Hiperintenso	Hiperintenso leve	Hiperintenso leve
Inflamación	Hipointenso	Hipointenso	Hipointenso

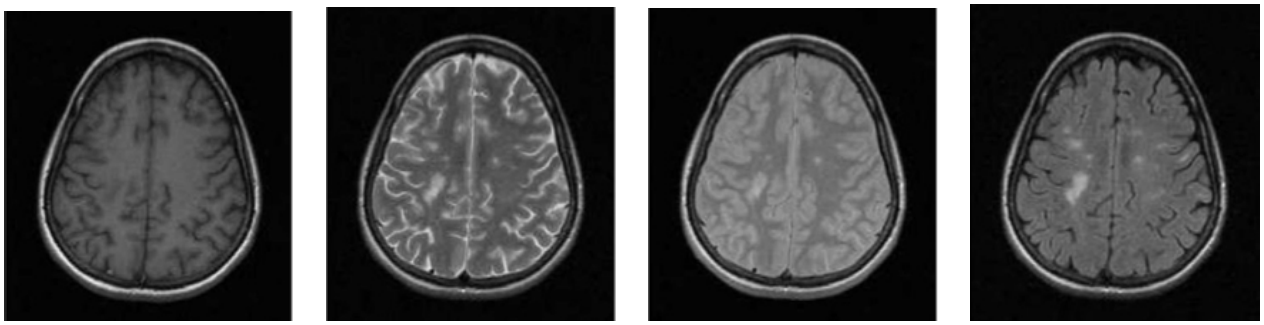
2.1.2. Lesiones de esclerosis múltiple

El encéfalo (también llamado cerebro) junto con la médula espinal constituyen el sistema nervioso central (SNC), el cual es responsable de la recepción, procesamiento y respuesta a la información sensorial. Dentro del sistema nervioso central, el cerebro es la estructura más compleja conocida, formada por más de 10^9 neuronas interconectadas. Estas conexiones, denominadas sinapsis, varían en número de manera significativa según el tipo de neurona, oscilando desde unos pocos cientos hasta aproximadamente 200.000 conexiones por neurona (John Hall, 2020). La materia gris está compuesta principalmente por capas de somas (cuerpo) de neuronas (Richard L Drake, 2022), lo que le da la tonalidad gris característica. La materia gris se localiza predominantemente en la superficie del cerebro, de este modo constituyendo el componente principal de la corteza cerebral. En el interior del cerebro se encuentra la materia blanca, formada principalmente por los axones mielinizados de las neuronas que establecen

conexiones entre ambos hemisferios. Por su parte, el líquido cefalorraquídeo es un plasma filtrado por células especializadas presentes en los ventrículos cerebrales. Este líquido fluye por el interior del cerebro además de rodear el cerebro y la médula espinal, desempeñando funciones esenciales como amortiguación, nutrición y limpieza de residuos.

La esclerosis múltiple es una enfermedad del sistema nervioso central que genera daños tanto en la materia blanca como en la materia gris del cerebro (Haider and Lassmann, 2024). Estas lesiones pueden ser observadas en la materia blanca mediante diferentes secuencias de imágenes de resonancia magnética nuclear: en las modalidades FLAIR, T2-weighted y PD-weighted estas lesiones destacan como hiperintensidades presentes en la materia blanca, en cambio en el canal T1-weighted las lesiones crónicas se presentan como zonas hipointensas (Danelakis et al., 2018; Gao et al., 2022).

Las lesiones de esclerosis múltiple suelen presentar una morfología redonda u ovoide, ubicándose en proximidad de pequeños vasos sanguíneos. Es más probable encontrar estas lesiones en la materia blanca periventricular (alrededor de los ventrículos cerebrales), en las regiones yuxtacorticales (cerca de la corteza cerebral) y en la sustancia blanca infratentorial del cerebelo (cerca del tentorium cerebelli el cual es la extensión de la duramadre que separa el cerebelo de la porción inferior del lóbulo occipital del cerebro) (Fazekas et al., 1999). En la Figura 2.5 se presentan diferentes MRI de un paciente con lesiones de esclerosis múltiple. Se puede notar que las lesiones en la imagen T1-weighted (Figura 2.5(a)) son más oscuras que la materia blanca, en cambio en las Figuras 2.5(b)-(d) las lesiones presentan una hiperintensidad respecto al resto de los tejidos cerebrales.



(a) **(b)** **(c)** **(d)**

Figura 2.5: Secuencias de MRI de un paciente con esclerosis múltiple: (a) T1-weighted, (b) T2-weighted, (c) PD-weighted y (d) FLAIR.

2.2. Segmentación de imágenes

La tarea de segmentación de imágenes tiene un rol muy importante en un amplio rango de aplicaciones, desde el análisis de imágenes médicas hasta la conducción autónoma de vehículos, la videovigilancia, el reconocimiento biométrico y la realidad aumentada (Minaee et al., 2022; Yu et al., 2023). Esta tarea se subdivide en tres categorias:

- Segmentación Semántica: Consiste en clasificar cada píxel (o vóxel en 3D) de una imagen en una de las clases predefinidas. Por ejemplo, en una imagen de un paisaje natural, las clases podrían ser "suelo", "cielo", "árbol" y "animal". El objetivo es asignar de manera exhaustiva una sola etiqueta de clase a cada píxel (vóxel) de la imagen. Es importante destacar que esta asignación se realiza sin diferenciar entre instancias individuales de una misma clase.

- Segmentación por Instancias: A diferencia de la segmentación semántica, la segmentación por instancias es una tarea de segmentación binaria que busca identificar y segmentar cada objeto individual (instancia) de la clase de interés por separado. Por ejemplo, esto quiere decir que si una de las clases de interés corresponde a "árbol" y existen múltiples árboles en la imagen, cada árbol sería segmentado como una instancia distinta.

- Segmentación Panóptica: Esta categoría es una combinación de las dos anteriores. Su objetivo es segmentar todos los píxeles de la imagen, asignándolos a una de $N \geq 1$ clases, y además identificar las instancias individuales dentro de cada clase. Así, por ejemplo, con $N = 2$ (árbol, persona), no solo clasificaría todos los píxeles de las clases "árbol" y "persona", sino que también diferenciaría "árbol 1", "árbol 2", ..., "persona 1", "persona 2", ..., etc.

De acuerdo con González y Woods (Gonzalez and Woods, 2001), el proceso de segmentación de una imagen I se puede ver como un proceso que la particiona en n subregiones, denotadas como $I_1, I_2, \ldots, I_n$, tal que:

(a) $\cup_{i=1}^{n} I_i = I$: Establece que la segmentación debe abarcar toda la imagen I, es decir, cada píxel debe ser asignado a una de las n subregiones I_i.

(b) I_i es una región conectada para $i = 1, 2, \ldots, n$: Esta condición establece que cada subregión I_i debe ser un grafo conexo, es decir, debe existir a lo menos un camino entre cada par de píxeles pertenecientes a la región que este completamente contenido en la región.

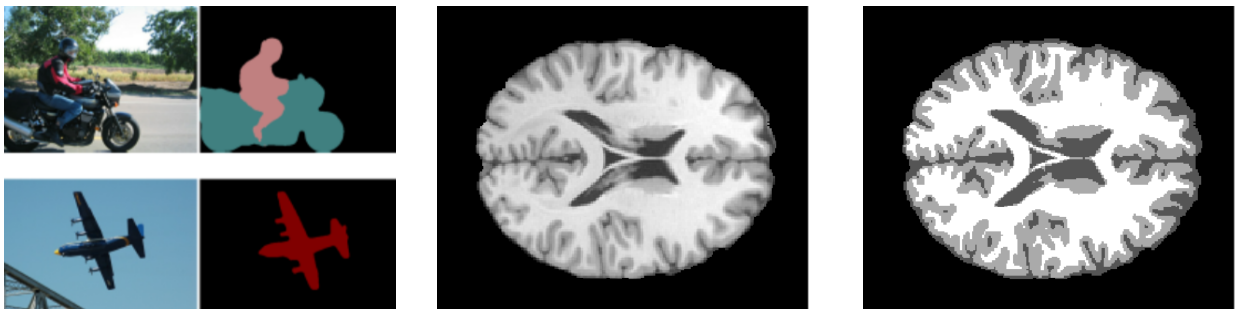
(c) $I_i \cap I_j = \emptyset$ para todo i y j, $i \neq j$: Aquí se impone la restricción que la regiones deben ser mutuamente disjuntas, es decir, cada píxel debe pertenecer a una y solo una región. De este modo evitando ambigüedad en la clasificación.

(d) $P(I_i) = \text{TRUE}$ para $i = 1, 2, \ldots, n$: Aquí $P(\cdot)$ representa un predicado lógico o una función de homogeneidad. Este predicado establece una propiedad o un conjunto de reglas que deben cumplir todos los píxeles pertenecientes a una región I_i. Como ejemplo de propiedad, esta puede basarse en la intensidad o textura de la subregión.

(e) $P(I_i \cup I_j) = \text{FALSE}$ para $i \neq j$: Esta condición hace hincapié en la adecuada separación entre regiones, ya que si dos subregiones adyacentes se unieran, la región resultante ya no cumpliría con el predicado lógico de homogeneidad en (d). Esta condición asegura que el predicado es verdadero solo cuando se evalúa en las regiones disjuntas ya establecidas.

En la Figura 2.6 se muestran ejemplos de segmentación de imágenes en diferentes contextos. En la Figura 2.6(a) se presentan ejemplos de segmentación de escenas de exteriores, seleccionadas del reconocido conjunto de datos COCO (Lin et al., 2014). La primera columna presenta una escena sin segmentar, mientras que la segunda columna muestra la segmentación de los objetos de interés realizada por un experto.

Por otro lado, en las Figuras 2.6(b) y (c) se ilustra la segmentación en el área de las imágenes médicas. La Figura 2.6(b) corresponde a una resonancia magnética nuclear del cerebro, obtenida en secuencia T1-weighted y la Figura 2.6(c) corresponde a la segmentación de la imagen en la Figura 2.6(b), en las regiones materia blanca, materia gris y líquido céfalorraquideo. Esta imagen médica fue obtenida del conjunto de datos del atlas ICBM Template (http://www.loni.usc.edu/research/atlases).



(a) **(b)** **(c)**

Figura 2.6: Ejemplos de segmentación de imágenes: (a) Segmentación de escenas, (b) secuencia T1-weighted y (c) Segmentación de MB, MG y LCR.

2.3. Fundamentos del Aprendizaje Automático

Considérese el espacio de probabilidad $(\mathcal{Z}, \mathcal{G}, \mathrm{P})$, donde $\mathcal{Z}$ es el espacio de muestreo, $\mathcal{G}$ es la σ-algebra de subconjuntos de $\mathcal{Z}$ y $\mathbb{P}$ es una medida de probabilidad definida sobre el espacio medible $(\mathcal{Z}, \mathcal{G})$. Se asume que el espacio medible es conocido, a diferencia de la medida de probabilidad $\mathbb{P}$, la cual es desconocida.

La única información disponible de P proviene de una muestra aleatoria $S = \{z_1, \ldots, z_n\} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, correspondiente a la observación de n variables aleatorias independientes e idénticamente distribuidas (i.i.d.) $Z = (X, Y)$, es decir, que provienen de un mismo espacio de probabilidad. Es importante notar que el espacio muestral $\mathcal{Z}$ puede corresponder únicamente al espacio de entrada $\mathcal{Z} = \mathcal{X}$ o al producto cartesiano entre el espacio de entrada y el espacio de salida $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. Por consiguiente, la medida $\mathbb{P}(\mathcal{Z})$ puede expresarse como $\mathbb{P}(X, Y) = \mathbb{P}(X) \cdot \mathbb{P}(Y|X)$. De este modo, $\mathbb{P}(X)$ describe el proceso generador de las entradas X y $\mathbb{P}(Y|X)$ describe el proceso generador de las salidas Y condicionadas a la entradas X, donde este último es denominado como supervisor.

El proceso de aprendizaje automático consiste en estimar una función que, a partir de una entrada, reproduzca la salida deseada proporcionada por el supervisor. Este proceso de aprendizaje está clasificado en tres paradigmas, de acuerdo a la disponibilidad de un supervisor:

- Aprendizaje supervisado: En este tipo de aprendizaje el supervisor esta disponible para todas las entradas.

- Aprendizaje no supervisado: El supervisor no esta disponible, por tanto, el algoritmo debe identificar por si mismo patrones o estructuras en los datos.

- Aprendizaje semisupervisado: Es un tipo de aprendizaje híbrido, es decir, existe una disponibilidad parcial del supervisor.

De acuerdo con (Vapnik, 1998), el modelo de aprendizaje supervisado basado en ejemplos está compuesto de tres elementos fundamentales: el generador, el supervisor y la máquina de aprendizaje. En la Figura 2.7 se ilustra el aprendizaje supervisado y el no supervisado. En el caso del aprendizaje supervisado, la tarea consiste en identificar una función f perteneciente a una clase de hipótesis $\mathcal{F}$ que emule el comportamiento del supervisor, es decir, que aprenda el mapeo de entrada a la salida a partir de una muestra aleatoria S.

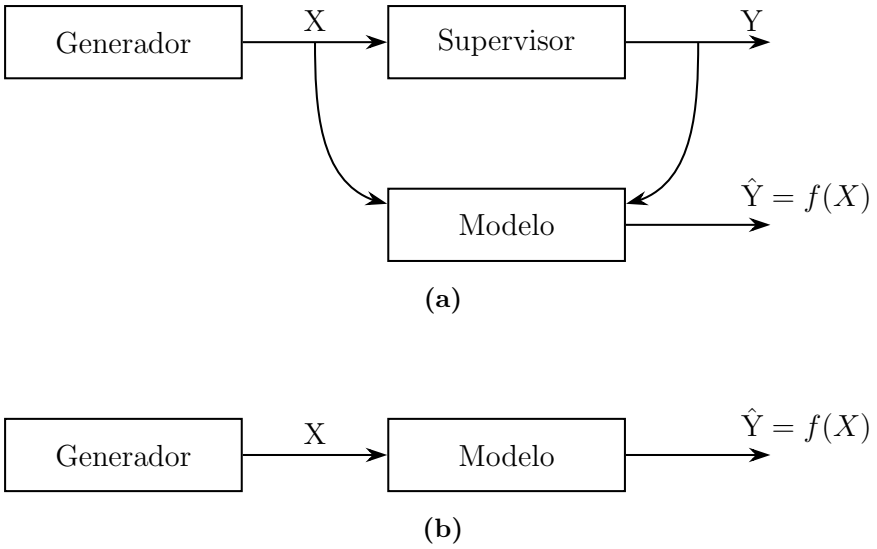


Figura 2.7: Paradigmas de aprendizaje automático. (a) Aprendizaje supervisado y (b) aprendizaje no supervisado.

Para construir la aproximación f del supervisor, se utiliza un algoritmo de aprendizaje $\mathcal{A}$ que realiza un mapeo desde la muestra S hacia una hipótesis f_s (Poggio et al., 2004)

Definición 2.1 *Un algoritmo de aprendizaje $\mathcal{A}$ es un proceso que toma como entrada un conjunto de entrenamiento finito $S = \{z_1, z_2, \ldots, z_n\} \in \mathcal{Z}^n$ y entrega como salida una hipótesis*

$f_s \in \mathcal{F}$, es decir,

$$\begin{aligned} \mathcal{A} : \mathcal{Z}^n &\rightarrow \mathcal{F} \\ \mathcal{A}(z_1, \ldots, z_n) &\rightarrow f_s. \end{aligned}$$

Para evaluar la calidad de la hipótesis obtenida por el algoritmo, se introduce una función de pérdida

$$\begin{aligned} L : \mathcal{X} \times \mathcal{Y} &\rightarrow \mathbb{R}_0^+ \\ (x, y) &\rightarrow L(f(x), y), \end{aligned} \tag{2.1}$$

la cual mide la discrepancia entre la salida real del supervisor y y la salida predicha $f(x)$ para una misma entrada x. La calidad global de la aproximación se mide mediante el valor esperado de la función de pérdida sobre todo el espacio $\mathcal{Z} = (\mathcal{X}, \mathcal{Y})$ respecto a la medida de probabilidad P, lo cual es conocido como riesgo funcional o riesgo esperado:

$$\mathcal{R}(f) = E_P[L(f(X), Y)] = \int_{\mathcal{X}\times\mathcal{Y}} L(f(x), y) dP(x, y), \tag{2.2}$$

sin embargo, debido a que $P(x, y)$ es desconocido, no es posible evaluar ni minimizar directamente el riesgo funcional. Es por esto, que en la práctica se minimiza su aproximación basada en la muestra observada S, conocida como riesgo empírico:

$$\mathcal{R}_{emp}(f) = \frac{1}{n} \sum_{i=1}^{n} L(f(x_i), y_i). \tag{2.3}$$

El principio de minimización del riesgo empírico consiste en encontrar la hipótesis f_s que minimice $\mathcal{R}_{emp}(f)$, esperando que esta aproximación también minimice el riesgo funcional. La capacidad del modelo para lograr este objetivo en datos no observados es conocido como su capacidad de generalización.

2.4. Redes neuronales convolucionales para la segmentación de imágenes

A continuación, se presenta una breve revisión de las Redes Neuronales Convolucionales (CNNs), con un enfoque en sus componentes fundamentales, los algoritmos de entrenamiento más utilizados y las principales arquitecturas empleadas para la tarea de segmentación de imágenes médicas.

2.4.1. Capas convolucionales

Las redes neuronales convolucionales (CNN) corresponden a una familia de redes neuronales artificiales multicapa o profundas (DNN). Su arquitectura les confiere propiedades favorables respecto a las redes multicapas tradicionales, tales como: la capacidad de extracción automática de características mediante una representación jerárquica, una menor cantidad de parámetros, invarianza o robustez ante traslaciones y distorsiones de los datos de entrada, y un menor requerimiento de ejemplos de entrenamiento (Bishop, 2006; Goodfellow et al., 2016; Tajbakhsh et al., 2016).

La operación que da el nombre a estas redes corresponde a la de convolución. Específicamente, la operación de convolución discreta se define entre una entrada I (una imagen $I \in \mathcal{R}^{H \times W}$) y una función o kernel K (un arreglo de $k_r \times k_c$ elementos). El resultado de esta operación es conocido como mapa de características:

$$\begin{aligned} S(i,j) &= (I * K)(i,j) \\ &= \sum_m \sum_n I(m,n) K(i-m, j-n). \end{aligned} \tag{2.4}$$

Dado que la propiedad de conmutatividad entre I y K no es factor crítico, ni influye en el proceso de aprendizaje, en la implementación de las redes neuronales, las bibliotecas de aprendizaje profundo utilizan en su lugar la operación de correlación cruzada (Goodfellow et al., 2016):

$$S(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n), \tag{2.5}$$

cuya operación se ilustra en la Figura 2.8.

Una vez realizada la operación de convolución sobre una zona de la entrada (campo receptivo), el elemento correspondiente del mapa de características resultante es procesado por una función de activación no lineal. En las CNNs profundas, es común utilizar la función de activación Rectified Linear Unit (ReLU), definida como $\text{ReLU}(x) = \text{máx}(0, x)$ (Fukushima, 1980), y generalizaciones como $\text{Leaky-ReLU}(x) = \text{máx}(a \cdot x, x)$ (Maas et al., 2013). Esta familia de funciones son ampliamente utilizadas debido a que por una parte mantienen la no linealidad, mientras que también disminuyen de manera drástica el efecto del gradiente desvaneciente que afecta a redes profundas con funciones de activación del tipo squashing como la sigmoide (logística) y tangente hiperbólica. En la Figura 2.9 se grafican estas funciones de activación.

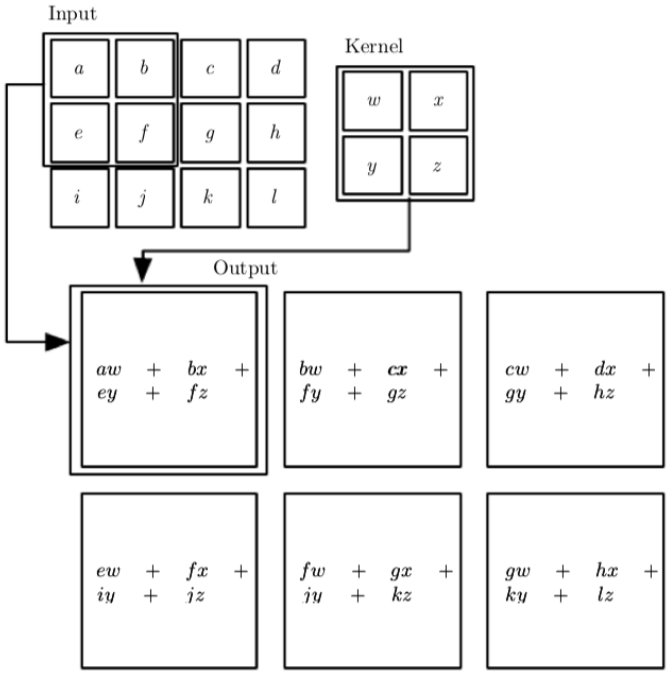
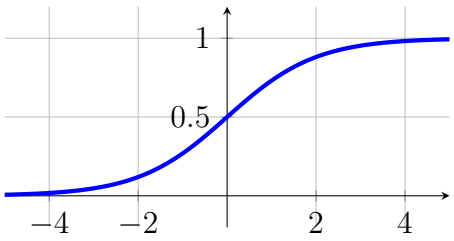


Figura 2.8: Operación de convolución implementada mediante correlación cruzada (Goodfellow et al., 2016).

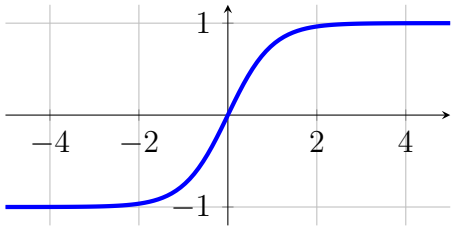
La función de activación softmax es una generalización de la función de activación sigmoide para $K > 2$ clases. Esta función de activación mapea un vector de entrada $\boldsymbol{x}$ a uno cuyos elementos forman una distribución de masa de probabilidad. Es decir, el largo del vector corresponde a la misma cantidad K de categorías, sus elementos son $x_i >= 0, \forall i \in \{1, 2, \ldots, K\}$, y la suma de sus elementos es 1:

$$\sigma(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}. \tag{2.6}$$

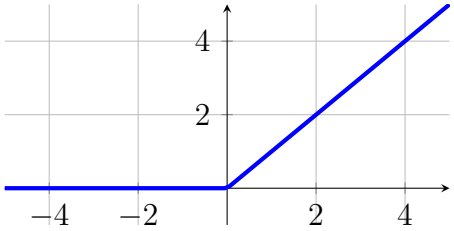
La salida de cada capa convolucional es una representación de la capa anterior conocida con el nombre de mapa de características, la que representa la intensidad con la que se ha encontrado similitud con el filtro o kernel que se ha aplicado a todo el mapa de características de la capa anterior. Una capa convolucional con n filtros es capaz de aprender a detectar n características locales donde cada una es representada en un mapa de características individual. La aplicación del mismo kernel a todo un mapa de características, es decir los mismos pesos, permite a la CNN disminuir su complejidad y, al mismo tiempo, obtener una representación



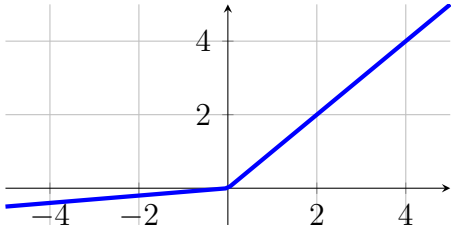
(a) $\sigma(x) = \frac{1}{1+e^{-x}}$



(b) $\tanh(x) = \frac{1-e^{-x}}{1+e^{-x}}$



(c) $\text{ReLU}(x) = \text{máx}(0, x)$



(d) $\text{Leaky-ReLU}(x) = \text{máx}(\alpha \cdot x, x)$

Figura 2.9: Funciones de activación más utilizadas en las redes neuronales: (a) Sigmoide, (b) Tangente hiperbólica, (c) ReLU y (d) Leaky-ReLU.

invariante ante traslaciones. Esto significa que el kernel puede detectar una característica específica en cualquier lugar del mapa de características.

Cuando una CNN solo contiene capas convolucionales, de pooling y upsampling, sin la presencia de capas densas o completamente conectadas (por lo general en la capa de salida), esta red es clasificada como completamente convolucional (FCNN).

Para la tarea de segmentación de imágenes, una CNN completamente convolucional puede presentar como capa de salida un mapa de características del mismo tamaño que la imagen de entrada, donde la cantidad K de mapas de características corresponde a la misma cantidad de K clases presentes en el problema de segmentación semántica. Por otra parte, para la tarea de clasificación, la CNN presenta una capa de salida con K nodos, donde K representa el número de clases a clasificar. Esta capa de salida puede ser implementada como una capa densa (completamente conectada) o, en el contexto de FCNNs, como una capa convolucional. Cuando la salida es densa, el resultado de la CNN es un vector $\hat{y}$ que representa las probabilidades de pertenencia a cada clase. Por otro lado, para las FCNNs diseñadas para la segmentación de imágenes completas, cada coordenada espacial en el dominio de la imagen de salida corresponde a un vector cuyas componentes están distribuidas entre los K mapas

de características de la capa final.

Suponiendo que los mapas de características y el kernel son arreglos cuyos lados presentan la misma longitud, el tamaño de los mapas de características de salida de una capa convolucional depende de los siguientes parámetros:

- i: tamaño de un eje del mapa de características,
- k: tamaño de un eje del kernel (filtro) convolucional,
- p: padding (relleno con ceros o intensidad representante de vecindario) aplicado en los dos extremos de cada eje,
- s: stride (paso) de aplicación del kernel sobre el mapa de características.

De este modo, el tamaño del mapa de características de salida en cada eje tendrá el siguiente tamaño (Dumoulin and Visin, 2016):

$$o = \left\lfloor \frac{i + 2p - k}{s} \right\rfloor + 1. \tag{2.7}$$

Como ejemplo gráfico, en la Figura 2.10 se presenta una operación de convolución con una entrada de tamaño 5×5, un kernel de 3×3, un padding de 1 y un stride de 2.

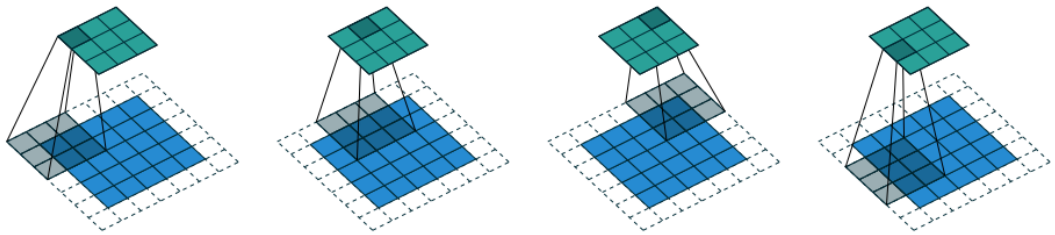


Figura 2.10: Operación de convolución de un kernel sobre el mapa de características (Dumoulin and Visin, 2016).

2.4.2. Capas de Pooling

A continuación de la capa convolucional, en muchos modelos de CNN se incorpora una capa de pooling (submuestreo). Este procedimiento tiene como objetivo reducir la dimensionalidad de los mapas de características mediante la selección de un valor representativo de vecindarios definidos por una ventana deslizante. Los tipos de pooling más comunes son:

- Max Pooling: Se selecciona el valor máximo dentro de la ventana deslizante.
- Average Pooling: Se calcula el promedio de todos los valores dentro de la ventana deslizante.
- Global Average Pooling: Se calcula el promedio de todos los valores en el mapa de características completo, resultando en un único valor por mapa.

Por lo general, la ventana deslizante en el pooling utiliza un stride mayor a 1, siendo el stride de 2×2 el más popular. La disminución en el tamaño del mapa de características implica para la CNN un decremento en la complejidad, y en un incremento en la invarianza ante traslaciones y deformaciones de objetos dentro de la imagen (Goodfellow et al., 2016; Taye, 2023). La Figura 2.11 ilustra un esquema de las tres operaciones de pooling más utilizadas.

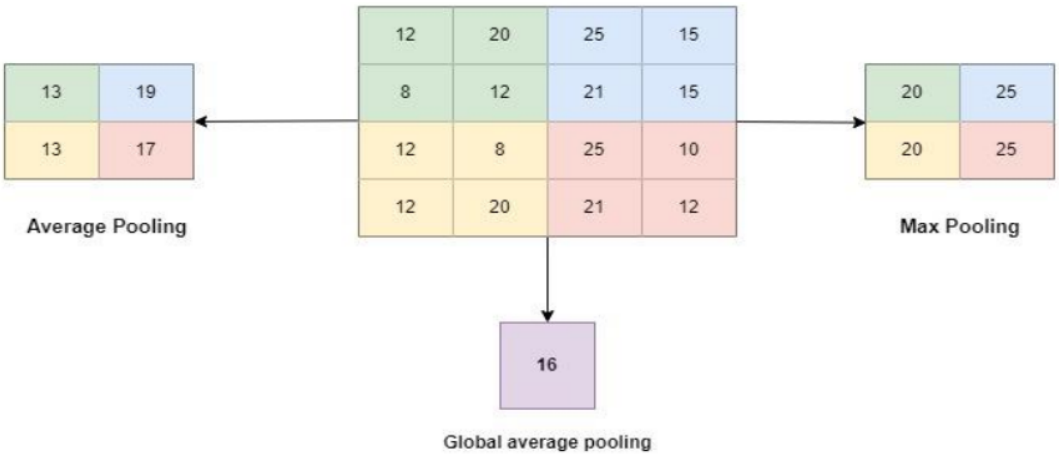


Figura 2.11: Principales operaciones de pooling (Taye, 2023).

2.4.3. Capas de Upsampling

Como se mencionó anteriormente, las capas convolucionales y de pooling pueden generar mapas de características de menor tamaño que el mapa de características de entrada. Esto ocurre cuando no se aplica padding (lo que equivale a $p = 0$) en las capas convolucionales o en el caso de emplear padding para mantener el tamaño, pero en conjunto con un stride $s > 1$. En las arquitecturas de redes neuronales convolucionales donde es necesario recuperar el tamaño original de la imagen de entrada, como en los autoencoders o las redes para

segmentación densa, se utilizan capas de upsampling como: capas de interpolación y capas de desconvolución (también conocidas como convolución transpuesta).

Para los tipos de upsampling correspondientes a interpolación bilineal y de deconvolución, el stride utilizado se considera como el inverso multiplicativo del stride s que se usó previamente para reducir el tamaño (es decir, $1/s$). De esta forma, la deconvolución o convolución con stride $1/s$ recupera el tamaño del mapa de características original antes de aplicar la convolución o el pooling con stride s. En la Figura 2.12 se presenta un ejemplo de upsampling mediante deconvolución, utilizando un kernel de tamaño 3×3 sobre una entrada de 3×3, con un stride de 2×2 (equivalente a $1/2 \times 1/2$ para la convolución) y zero padding de 1×1 en los bordes.

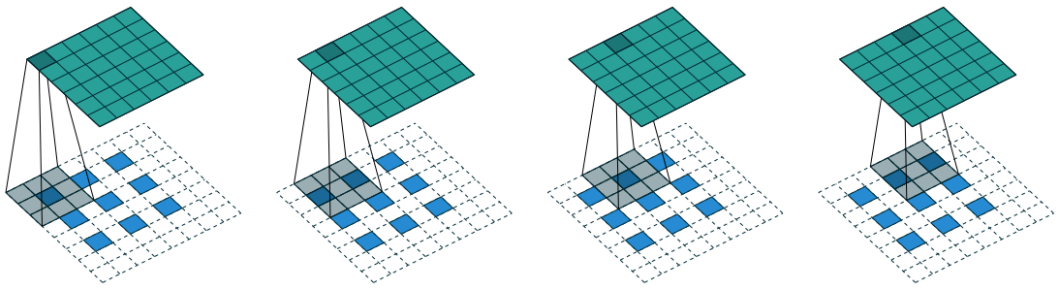


Figura 2.12: Upsampling mediante deconvolución (Dumoulin and Visin, 2016).

Interpolación

En las capas de upsampling con interpolación es posible emplear métodos de interpolación del tipo: vecino más cercano, bicúbica y bilineal (Dougherty, 2009; Gonzalez and Woods, 2001). La interpolación bilineal es el tipo de interpolación más utilizado en CNNs debido a su equilibrio entre buenos resultados y bajo costo computacional. Esto se debe a que puede implementarse como una convolución con un kernel de pesos constantes, como el siguiente ejemplo $k = \begin{bmatrix} 0.25 & 0.5 & 0.25 \\ 0.5 & 1 & 0.5 \\ 0.25 & 0.5 & 0.25 \end{bmatrix}$. A modo de ejemplo, se presenta la siguiente operación de upsampling del mapa de características de 3×3, con $I = \begin{bmatrix} 5 & 3 & 1 \\ 0 & 2 & 4 \\ 4 & 3 & 5 \end{bmatrix}$ utilizando un stride

de 2×2 y zero-padding de 1×1 en los bordes. La operación se muestra a continuación:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{5} & 0 & \mathbf{3} & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{0} & 0 & \mathbf{2} & 0 & \mathbf{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{4} & 0 & \mathbf{3} & 0 & \mathbf{5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0.25 & 0.5 & 0.25 \\ 0.5 & 1 & 0.5 \\ 0.25 & 0.5 & 0.25 \end{bmatrix} = \begin{bmatrix} \mathbf{5} & 4 & \mathbf{3} & 2 & \mathbf{1} & 0.5 \\ 2.5 & 2.5 & 3.5 & 2.5 & 2.5 & 1.3 \\ \mathbf{0} & 1 & \mathbf{2} & 3 & \mathbf{4} & 2 \\ 2 & 2.3 & 2.5 & 3.5 & 4.5 & 2.3 \\ \mathbf{4} & 3.5 & \mathbf{3} & 4 & \mathbf{5} & 2.5 \\ 2 & 1.8 & 1.5 & 2 & 2.5 & 1.3 \end{bmatrix} \tag{2.8}$$

A diferencia de la interpolación, en la capa de desconvolución se reemplaza el kernel de pesos constantes por un kernel con pesos ajustables. Esto permite a la capa de upsampling participar en el entrenamiento, lo que confiere la capacidad de optimizarse de manera automática.

Una desventaja del tipo de upsampling mediante el uso de capas con interpolación bilineal y deconvolución es que requieren agregar muchas filas y columnas con ceros, de este modo aumentando el tamaño del mapa de características, el que se debe recorrer aplicando la convolución con el kernel, lo que puede ser ineficiente.

Convolución Transpuesta

Si bien los métodos de upsampling por interpolación bilineal y desconvolución son efectivos, a menudo requieren la inserción de un gran número de ceros, lo que aumenta el costo computacional. En cambio la convolución transpuesta ofrece una solución más eficiente para lograr el mismo resultado de upsampling. Esta mayor eficiencia radica en que el paso forward y el paso backward se pueden realizar mediante operaciones matriciales. Esto se consigue construyendo una matriz sparse C obtenida utilizando los pesos del kernel y utilizando su transpuesta C^T (Dumoulin and Visin, 2016). Por ejemplo, para imitar la operación de convolución entre un kernel de tamaño 3×3, $k = \begin{bmatrix} w_{0,0} & w_{0,1} & w_{0,2} \\ w_{1,0} & w_{1,1} & w_{1,2} \\ w_{2,0} & w_{2,1} & w_{2,2} \end{bmatrix}$ se construye la matriz

$$C = \begin{bmatrix} w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} \end{bmatrix} \tag{2.9}$$

que al ser multiplicada por la matriz columna A de 16×1

$$A = \left[\; I_{0,0}\, I_{0,1}\, I_{0,2} \; \ldots \; I_{3,1}\, I_{3,2}\, I_{3,3} \; \right]^T$$

construida con los nodos de un mapa de características de 4×4, genera una matriz de 4×1. Esta matriz al ser reordenada genera el mapa de características de salida para el paso forward, con un tamaño menor, en este caso de 2×2 (stride = 2). En cambio, para la operación backward, es decir, para obtener un mapa de características con el tamaño mayor, especificamente el original 4×4, es necesario multiplicar la matriz C^T con el vector de salida de 4×1 y luego ordenar los elementos. Gracias a esta formulación matricial y su eficiente implementación en bibliotecas de deep learning, la convolución transpuesta ha ganado mucha popularidad en modelos que requiren upsampling como aquellos con arquitectura encoder-decoder utilizados en la tarea de segmentación.

2.4.4. Ajuste de los parámetros de la red neuronal

La tarea de optimización en deep learning consiste en la búsqueda de los parámetros de un modelo que minimicen el riesgo empírico (Eq. (2.3)) en lugar del riesgo funcional. Este se debe a que no se conoce la verdadera distribución conjunta generadora de los pares ejemplo/etiqueta (x, y), sino que solo se conoce una muestra aleatoria $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. Para abordar esta tarea, se utilizan diversos algoritmos de optimización, los que a su vez se basan en la técnica del gradiente descendente.

Gradiente descendente

Para la estimación de los parámetros en modelos de redes neuronales profundas, el objetivo del algoritmo del gradiente descendente (SGD) es buscar los parámetros que minimicen la función de pérdida $l(f(x;\theta), y)$. Es decir, encontrar el vector de pesos θ del modelo $f(x;\theta)$, donde se considera a la entrada x como un valor constante proveniente de la muestra aleatoria S. La dirección en el espacio de parámetros que minimiza la función de pérdida $l(f(x;\theta), y)$ es la dirección opuesta al gradiente g. Este gradiente por lo general se estima utilizando batchs, es decir, submuestras aleatorias de tamaño m extraídas del conjunto de entrenamiento. La

estimación $\hat{g}$ del gradiente se define como:

$$\hat{g} = \frac{1}{m} \cdot \nabla_\theta \sum_{i=1}^{m} l(f(x_i; \theta), y_i). \tag{2.10}$$

A partir de esta estimación, los parámetros del modelo se actualizan iterativamente siguiendo la siguiente regla:

$$\theta^t = \theta^{t-1} - \varepsilon \cdot \hat{g}, \tag{2.11}$$

donde t es a la última actualización del conjunto de parámetros θ y ε es la tasa de aprendizaje, corresponde a un hiperparámetro que controla el tamaño de paso en cada actualización.

El algoritmo de optimización del gradiente descendente estocástico (SGD) se presenta en pseudocódigo en Algoritmo 1. Las entradas corresponden al valor inicial de la tasa de apren-

Algoritmo 1 SGD

Input: ε, $\theta^{(0)}$, m.
$t = 1$
while no se cumpla criterio de término **do**
  Muestrear batch $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ desde conjunto de entrenamiento
  Estimación del gradiente: $\hat{g} \leftarrow \frac{1}{m} \nabla_\theta \sum_i l(f(x_i; \theta^{(t-1)}), y_i)$
  for $k \leftarrow 1$ to $\#\theta$ **do**
    Estimar gradiente $\hat{g}_{\theta_k} = \frac{\partial \hat{g}(\theta)}{\theta_k}$ mediante algoritmo backpropagación
    $\theta_k^{(t)} \leftarrow \theta_k^{(t-1)} - \varepsilon \cdot \hat{g}_{\theta_k}$
  end for
  $t \leftarrow t + 1$
end while
Output: θ

dizaje ε, los valores iniciales de los parámetros θ, y el tamaño m del batch. Por consistencia en los términos de los algoritmos presentados, se consideran los siguientes supuestos: se asume la existencia de un modelo de redes neuronales profundas, un conjunto de entrenamiento (batch) que es subconjunto de S, que el criterio de término del entrenamiento existe y que la tasa de aprendizaje ε es constante o puede disminuir de acuerdo a algún criterio.

Gradiente descendente con momentum

Si bien el algoritmo de optimización SGD es ampliamente utilizado en deep learning, presenta algunos problemas como: una baja velocidad de convergencia, una estimación ruidosa del gradiente debido al muestreo aleatorio de cada batch, y una falta de robustez frente a cambios abruptos de la curvatura del espacio de parámetros (Goodfellow et al., 2016). Una manera exitosa de mitigar estos problemas consiste en agregar un término adicional en la estimación del gradiente, conocido como momentum o velocidad v. Este término corresponde a un promedio móvil ponderado de las estimaciones anteriores del gradiente

$$v^{(t)} = \alpha \cdot v^{(t-1)} - \varepsilon \cdot \nabla_\theta \left(\frac{1}{m} \sum_i l(f(x_i; \theta^{(t)}), y_i) \right), \tag{2.12}$$

$$\theta^{(t)} = \theta^{(t-1)} + v^{(t)}, \tag{2.13}$$

donde α toma valores en el intervalo continuo $[0, 1)$. Cuando $\alpha = 0$ corresponde al algoritmo SGD. Para valores $\alpha \in (0, 1)$ la ponderación de los términos anteriormente estimados del gradiente disminuyen de manera exponencial, dando más peso a los gradientes recientes. En Figura 2.13(a) se ilustra un ejemplo de la actualización de θ en el espacio de parámetros para el algoritmo SGD, mientras que la Figura 2.13(b) muestra la versión con momentum, donde se puede notar que el algoritmo SGD con momentum converge en una menor cantidad de pasos.

Una variante del algoritmo SGD con momentum fue introducida en (Sutskever et al., 2013), inspirada en el método del gradiente acelerado de Nesterov. La regla de actualización de los parámetros utilizando SGD con momentun Nesterov esta definida como:

$$v^{(t)} = \alpha \cdot v^{(t-1)} - \varepsilon \cdot \nabla_\theta \left(\frac{1}{m} \sum_i l(f(x_i; \theta^{(t-1)} + \alpha \cdot v^{(t-1)}), y_i) \right) \tag{2.14}$$

$$\theta^{(t)} = \theta^{(t-1)} + v^{(t)} \tag{2.15}$$

cuya diferencia con el método de momentum estandar reside en el punto donde se evalúa el gradiente, es decir, con Nesterov el gradiente es evaluado después que es aplicado el componente de momentum, pero antes de actualizar los parámetros. Lo anterior permite obtener la estimación del gradiente antes de aplicarlo y por tanto incluirlo en el promedio ponderado de gradientes. Este método hace más robusto al algoritmo de optimización al anticipar grandes

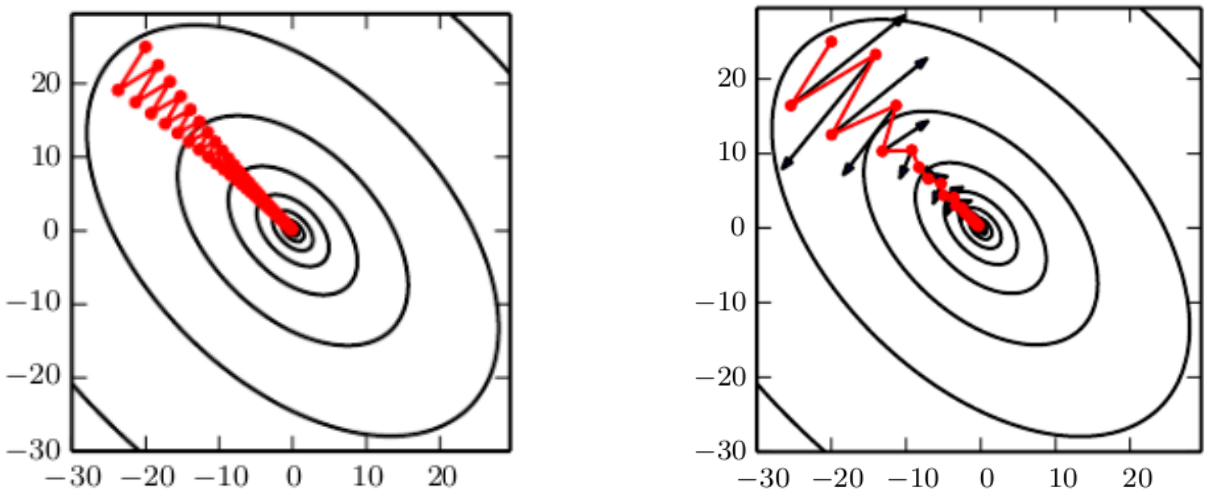


Figura 2.13: Comparación de algoritmos de optimización. (a) SGD estándar y (b) SGD con momentum. Las flechas negras indican el vector del gradiente actual (paso sin momentum) (Goodfellow et al., 2016).

gradientes producto de curvaturas abruptas en el espacio de parámetros, de este modo ayudando a evitar una posible pérdida de un mínimo. El algoritmo SGD con momentum tipo Nesterov se presenta en el Algoritmo 2.

Gradiente descendente con tasa de aprendizaje adaptativa

Hasta ahora, la tasa de aprendizaje ε ha permanecido constante en los algoritmos de optimización presentados. Sin embargo, dado que el espacio de parámetros de una red neuronal puede estar compuesto por miles o incluso millones de elementos (parámetros), la convergencia puede ser mejorada si la actualización de sus parámetros se realiza de manera individual y adaptativa para cada uno.

El algoritmo AdaGrad (Duchi et al., 2011) ajusta de manera individual las tasas de aprendizajes de todos los parámetros del modelo mediante la estimación del segundo momento r, como el inverso de la raíz cuadrada de la suma acumulada de todas sus derivadas parciales históricas. De este modo, los parámetros con grandes gradientes ven su tasa de aprendizaje decrecer rápidamente, en cambio los parámetros con pequeños gradiente parciales decrecen más lentamente. Como resultado de la aplicación de este algoritmo el progreso en el espacio

Algoritmo 2 SGD con momentum Nesterov

Input: ε, $\theta^{(0)}$, m y $v^{(0)}$.
$t = 1$
while no se cumpla criterio de término **do**
  Muestrear batch $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ desde conjunto de entrenamiento
  Aplicar actualización interina: $\tilde{\theta} \leftarrow \theta^{(t-1)} + \alpha \cdot v^{(t-1)}$
  Estimación del gradiente en punto interino: $\hat{g} = \frac{1}{m} \cdot \nabla_{\tilde{\theta}} \sum_i l(f(x_i; \tilde{\theta}), y_i)$
  for $k \leftarrow 1$ to $\#\theta$ **do**
    Estimar gradiente $\hat{g}_{\theta_k} = \frac{\partial \hat{g}(\theta)}{\theta_k}$ mediante algoritmo backpropagación
    Actualizar momentum: $v^{(t)} = \alpha \cdot v^{(t-1)} - \varepsilon \cdot \hat{g}_{\theta_k}$
    $\theta_k^{(t)} = \theta_k^{(t-1)} + v^{(t)}$
  end for
  $t \leftarrow t + 1$
end while
Output: θ

de parámetros es en direcciones de pendientes más suaves, pero el efecto adverso es que en redes neuronales profundas la tasa de aprendizaje puede decrecer de manera prematura y excesiva. Por tanto, AdaGrad no funciona bien para todos los modelos de redes neuronales profundas (Goodfellow et al., 2016).

Posteriormente, en (Hinton, 2012) se presenta el algoritmo RMSProp como una modificación en el uso de las gradiente históricos de los parámetros. En lugar de acumular la suma de los cuadrados del gradiente históricos, se acumulan estos gradientes pero ponderados con decaimiento exponencial, de este modo ponderando con mayor peso a los gradientes recientes, mientras que los gradientes más antiguos son ponderados con un menor peso, por tanto su influencia es significativamente menor. En Algoritmo 3 se detallan los pasos del algoritmo de optimización RMSProp, donde ρ corresponde a un nuevo hiperparámetro que controla la velocidad del decaimiento exponencial. La constante δ es una pequeña constante utilizada para dar estabilidad numérica y para evitar divisiones por 0, y la variable r corresponde a la acumulación ponderada del cuadrado del gradiente.

Mas tarde, fue propuesto el algoritmo de optimización Adam (Kingma and Ba, 2015), que combina las características del algoritmo RMSProp con el uso de momentum. A diferencia

Algoritmo 3 RMSProp

Input: ε, $\theta^{(0)}$, m, ρ y δ.
$r = 0$
$t = 1$
while no se cumpla criterio de término **do**
  Muestrear batch $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ desde conjunto de entrenamiento
  Estimación del gradiente en punto interino: $\hat{g} = \frac{1}{m} \cdot \nabla_{\tilde{\theta}} \sum_i l(f(x_i; \theta^{(t-1)}), y_i)$
  for $k \leftarrow 1$ to $\#\theta$ **do**
    Estimar gradiente $\hat{g}_{\theta_k} = \frac{\partial \hat{g}(\theta)}{\theta_k}$ mediante algoritmo backpropagación
    $r \leftarrow \rho \cdot r + (1 - \rho) \cdot \hat{g}^2_{\theta_k}$
    $\triangle\theta_k = -\frac{\varepsilon}{\sqrt{\delta + r}} \cdot \hat{g}_{\theta_k}$
    $\theta_k^{(t)} = \theta_k^{(t-1)} + \triangle\theta_k$
  end for
  $t \leftarrow t + 1$
end while
Output: θ

de RMSProp, Adam introduce la estimación de los dos primeros momentos r y s en la ponderación de los gradientes históricos, y además, realiza una corrección del sesgo de estas estimaciones. Adam es presentado en el Algoritmo 4.

2.4.5. Arquitecturas de redes convolucionales

Las redes neuronales convolucionales (CNNs, por sus siglas en inglés) corresponden al tipo de arquitectura más exitosa y más utilizada por la comunidad de aprendizaje profundo para tareas de visión por computadora (Minaee et al., 2022). Las CNNs tuvieron sus inicios en el modelo *Neocognitron*, propuesto por Fukushima en el paper seminal (Fukushima, 1980), cuya idea se basó en los modelos del campo receptivo de la corteza visual. Años mas tarde, se introdujo el uso de pesos compartidos en CNNs para el reconocimiento de fonemas (Waibel et al., 1989). Posteriormente, se desarrolló una arquitectura compuesta por capas convolucionales, de submuestreo (pooling) y capas completamente conectadas (fully connected) (LeCun et al., 1998). Esta arquitectura es considerada el primer diseño moderno de una

Algoritmo 4 Adam

Input: ε, $\theta^{(0)}$, m, ρ_1, ρ_2 y δ.
$s = 0$, $r = 0$
$t = 1$, $q = 0$
while no se cumpla criterio de término **do**
  Muestrear batch $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ desde conjunto de entrenamiento
  Estimación del gradiente en punto interino: $\hat{g} = \frac{1}{m} \cdot \nabla_{\tilde{\theta}} \sum_i l(f(x_i; \theta^{(t-1)}), y_i)$
  $q \leftarrow q + 1$
  for $k \leftarrow 1$ to $\#\theta$ **do**
    Estimar gradiente $\hat{g}_{\theta_k} = \frac{\partial \hat{g}(\theta)}{\theta_k}$ mediante algoritmo backpropagación
    $s \leftarrow \rho_1 \cdot s + (1 - \rho_1) \cdot \hat{g}_{\theta_k}$
    $r \leftarrow \rho_2 \cdot r + (1 - \rho_2) \cdot \hat{g}^2_{\theta_k}$
    $\hat{s} \leftarrow \frac{s}{1-\rho_1^q}$
    $\hat{r} \leftarrow \frac{r}{1-\rho_2^q}$
    $\triangle\theta_k = -\varepsilon \cdot \frac{\hat{s}}{\sqrt{\delta + \hat{r}}} \cdot \hat{g}_{\theta_k}$
    $\theta_k^{(t)} = \theta_k^{(t-1)} + \triangle\theta_k$
  end for
  $t \leftarrow t + 1$
end while
Output: θ

CNN, como se puede apreciar en la Figura 2.14. Fue en 2012 donde las CNNs alcanzaron un notable reconocimiento y un salto en popularidad cuando la red AlexNet (Krizhevsky et al., 2012) obtuvo el primer lugar en el desafío ImageNet 2012, superando ampliamente a todos sus competidores. Explicaciones de este exito se atribuyen al uso de la función de activación ReLU, previamente propuesta en (Fukushima, 1980), la incorporación de la técnica Dropout para reducir el sobreajuste, y lo más importante, la implementación de la arquitectura en dos unidades de procesamiento gráfico (GPUs). De este modo, permitiendo el uso de redes neuronales con gran cantidad de parámetros y, al mismo tiempo, acelerando significativamente el tiempo de convergencia durante el entrenamiento.

Posteriormente, surgieron otras arquitecturas de CNNs muy influyentes y ampliamente citadas en la literatura, como VGG-19 (Simonyan and Zisserman, 2015) (Figura 2.16(b))

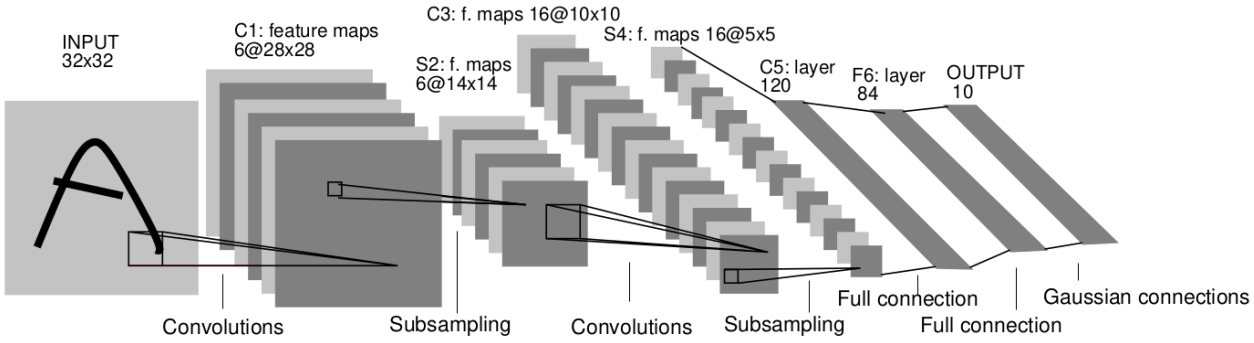


Figura 2.14: Arquitectura LeNet-5 propuesta por LeCun et al. (1998).

y ResNet (He et al., 2016). La red VGGNet constituida por 19 capas, aunque demostró buenos resultados en la tarea de clasificación de imágenes, su principal limitación radica en la dificultad para incrementar su profundidad debido al problema del gradiente desvaneciente, fenómeno que derivó en la disminución de la capacidad de generalización.

Esta desventaja de la arquitectura VGGNet fue abordada de manera eficaz por la red ResNet, la cual introdujo el concepto de bloques residuales (Figura 2.16(a)). Estos bloques residuales permiten que los gradientes fluyan directamente a la capa siguiente a través de conexiones de tipo skip connections (de salto), de este modo mitigando el problema del gradiente desvaneciente y, al mismo tiempo, posibilitando un aumento en la profundidad de las redes (34 capas para ResNet), por consiguiente permitiendo una mayor capacidad de generalización (Figura 2.16(b)).

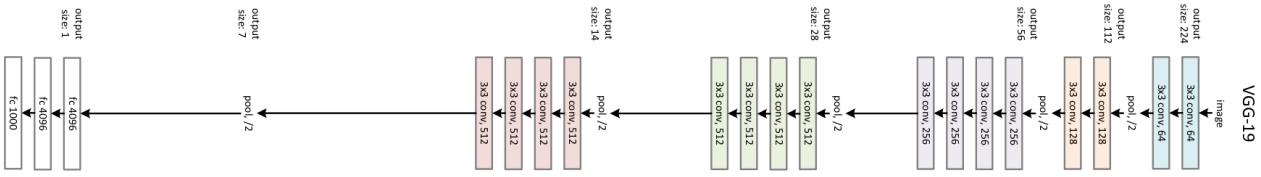


Figura 2.15: Arquitectura de la red VGG-19 propuesta por Simonyan and Zisserman (2015).

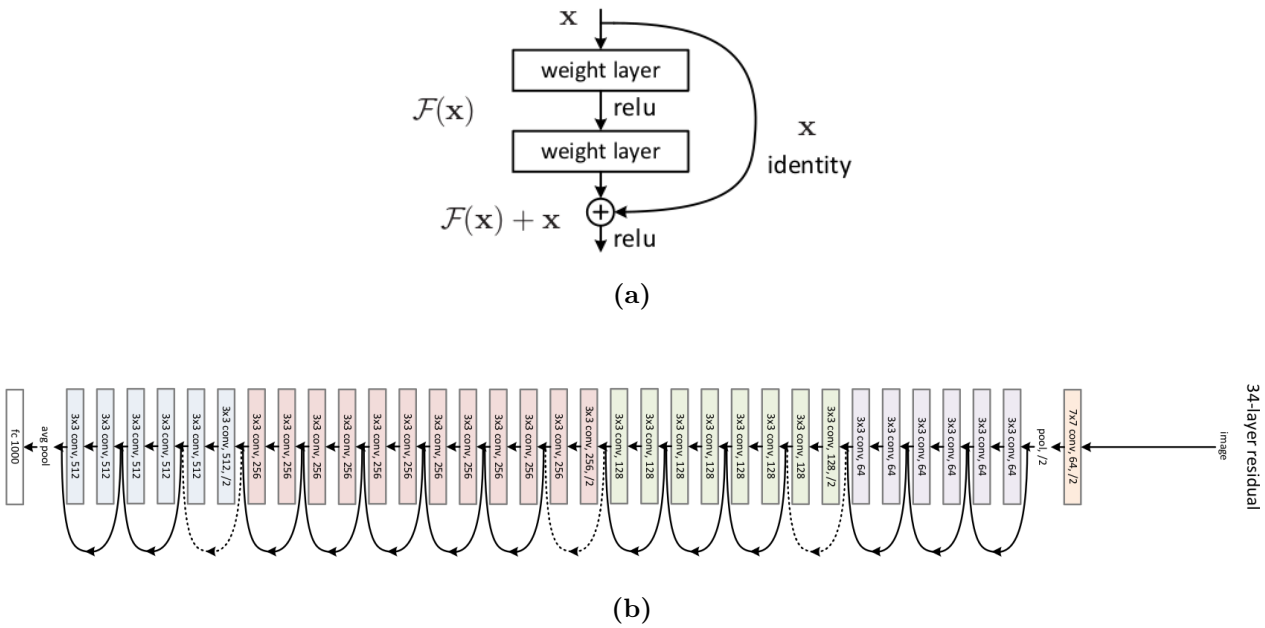


Figura 2.16: Componentes de la red residual profunda. (a) Bloque residual y (b) arquitectura general de ResNet (He et al., 2016).

2.4.6. Arquitecturas de redes convolucionales para segmentación de imágenes

Las CNNs no solo han sido utilizadas con éxito en la clasificación de imágenes, sino también en la segmentación semántica. La arquitectura propuesta en (Long et al., 2015) es considerada como la arquitectura basal en el desarrollo de arquitecturas de CNNs para la tarea de segmentación semántica de imágenes (Yu et al., 2023), (Punn and Agarwal, 2022). Para esto, Long et al. proponen una red completamente convolucional (FCN), donde las capas densas (completamente conectadas) de redes como AlexNet (Krizhevsky et al., 2012), VGGNet (Liu and Deng, 2015) y GoogLeNet (Szegedy et al., 2015) son reemplazadas por capas convolucionales. La última capa de la FCN emplea una capa de deconvolución (o transposed convolution) para restaurar la resolución espacial del mapa de características, conectándose con mapas de características de capas anteriores mediante skip connections (Figura 2.17)(a)). De esta forma, generando una arquitectura de CNN capaz de segmentar imágenes de entrada de tamaño variable (Fig. 2.17)(b)).

Este tipo de segmentación se realiza de manera densa, es decir, la salida de la red corres-

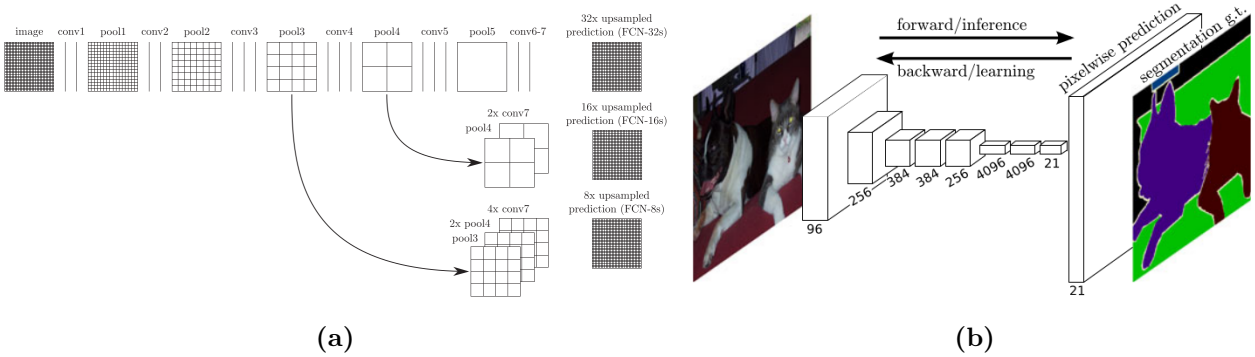


Figura 2.17: Arquitectura FCN propuesta por Long et al. (2015). (a) Mecanismo de conexiones de salto (skip connections) y (b) esquema general de la red.

ponde a un mapa de características del mismo tamaño que la imagen de entrada. Dicho mapa de características contiene valores continuos en el intervalo $[0, 1]$, generados por la función de activación softmax en la capa de salida. La máscara de segmentación se obtiene mediante binarización, aplicando un umbral de 0.5 (Figura 2.18).

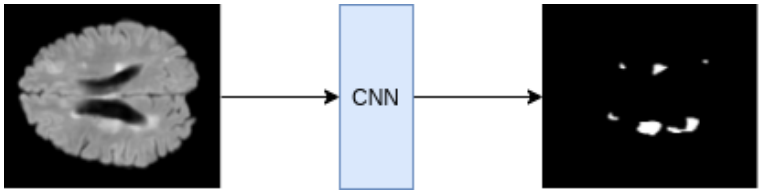


Figura 2.18: Segmentación densa producida por una CNN completamente convolucional.

Si bien las derivaciones de la arquitectura FCN alcanzaron buenos resultados en la tarea de segmentación semántica, se identificó un uso ineficiente del contexto global en la etapa de reconstrucción de la resolución original. Para abordar esta limitación, Noh et al. en (Noh et al., 2015) propusieron el modelo DeConvNet con una arquitectura del tipo encoder-decoder (codificador-decodificador). La etapa de codificación está compuesta por una red VGGNet de 16 capas, mientras que la etapa de decodificación emplea secuencias de capas deconvolucionales para reconstruir la imagen segmentada (Figura 2.19).

Posteriormente, Badrinarayanan et al. (Badrinarayanan et al., 2015) desarrollaron SegNet, otra arquitectura de tipo encoder-decoder con la que se buscaba ser más eficiente en el número de parámetros a entrenar. Este modelo utiliza capas de upsampling basadas en interpolación

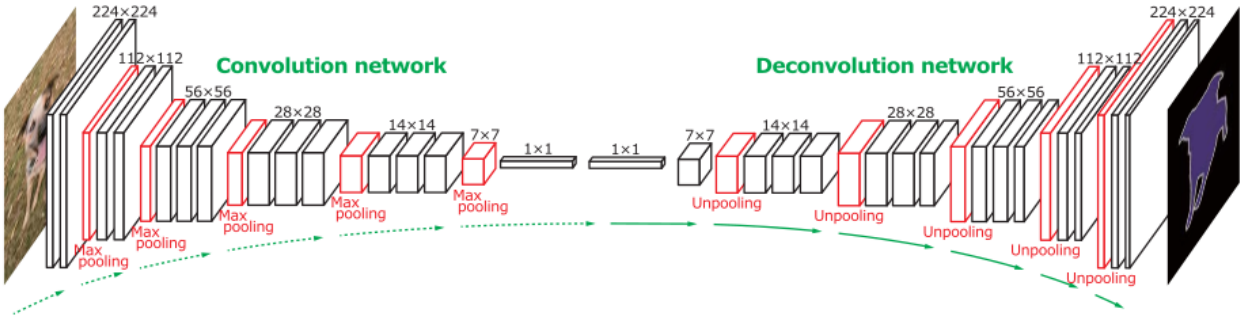


Figura 2.19: Arquitectura de la red DeConvNet para segmentación semántica (Noh et al., 2015).

en lugar de utilizar capas deconvolucionales para la reconstrucción de la resolución. Para guiar la interpolación, utiliza las posiciones de los máximos obtenidos en las capas de max-pooling de la etapa de codificación. Sin embargo, este enfoque no aprovecha toda la información espacial, ya que omite los píxeles adyacentes a cada máximo.

En esta misma línea de arquitecturas codificador-decodificador para la tarea de segmentación semántica, especialmente en imágenes médicas, en (Ronneberger et al., 2015) los autores propusieron la red U-Net. En esta arquitectura, durante el proceso de aumento de resolución, los mapas de características se transfieren directamente desde la etapa de codificación hacia la correspondiente etapa de decodificación mediante skip connections. Este mecanismo permite recuperar el contexto espacial en múltiples escalas (Figura 2.20). Se demostró la eficacia de U-Net al obtener el primer lugar, por un amplio margen, en la competencia *ISBI Cell Tracking Challenge 2015* para la segmentación de células en imágenes de microscopía. Desde su publicación, U-Net se consolidó como la arquitectura de referencia basal, adoptada y extendida por numerosas variantes que han alcanzado el estado del arte en la tarea de segmentación de imágenes médicas (Azad et al., 2024; Punn and Agarwal, 2022). Entre estas variantes destacan 3D U-Net (Çiçek et al., 2016) y V-Net (Milletari et al., 2016), ambas diseñadas para procesar volúmenes tridimensionales. En V-Net, además, se incorporaron bloques residuales en cada nivel de resolución del codificador y del decodificador, mejorando la propagación del gradiente y la capacidad de generalización (Figura 2.21). V-Net mostró un desempeño sobresaliente en el conjunto de datos PROMISE12, correspondiente a imágenes de próstata en modalidad MRI, consolidándose como una arquitectura ampliamente utilizada en el ámbito de la segmentación médica (Ma et al., 2020).

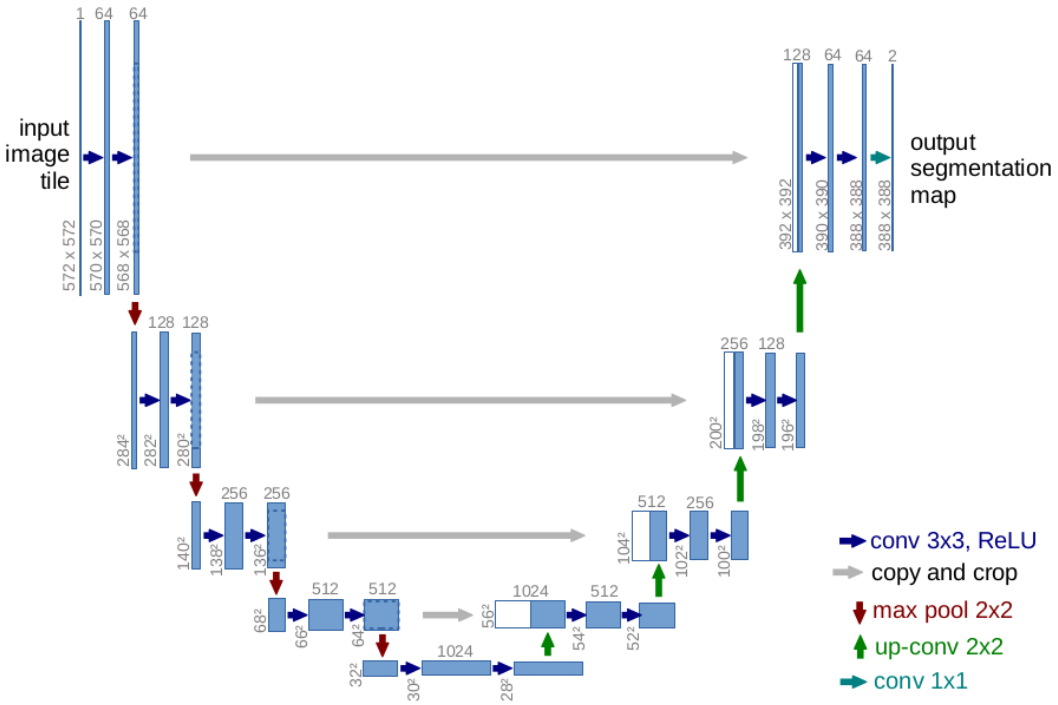


Figura 2.20: Arquitectura U-Net con sus caminos de contracción (encoder) y expansión (decoder) (Ronneberger et al., 2015).

2.5. Funciones de pérdida

Siguiendo la notación introducida en la sección 2.1, sea $G : \Omega \to \{0, 1\}$ la segmentación binaria o ground truth y $s_\theta : \Omega \to [0, 1]$ la salida softmax de la red neuronal convolucional, la cual representa una probabilidad de pertenencia de cada píxel (vóxel) a la clase de interés.

Los bordes de los objetos (por ejemplo, lesiones) en el ground truth G se denotan como GB. Del mismo modo, los bordes predichos PB pueden obtenerse a partir de P al umbralizar la salida de la red ($P = 1_{[s_\theta > 0.5]}$), lo cual es una práctica habitual para crear segmentaciones binarias a partir de predicciones de probabilidad.

Considerando que las funciones de pérdida que se revisarán a continuación utilizan los elementos de la matriz de confusión para un problema de clasificación binaria (ver Tabla 2.2), resulta necesario definir sus componentes en función de G y s_θ.

Las expresiones de los verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN) son las siguientes:

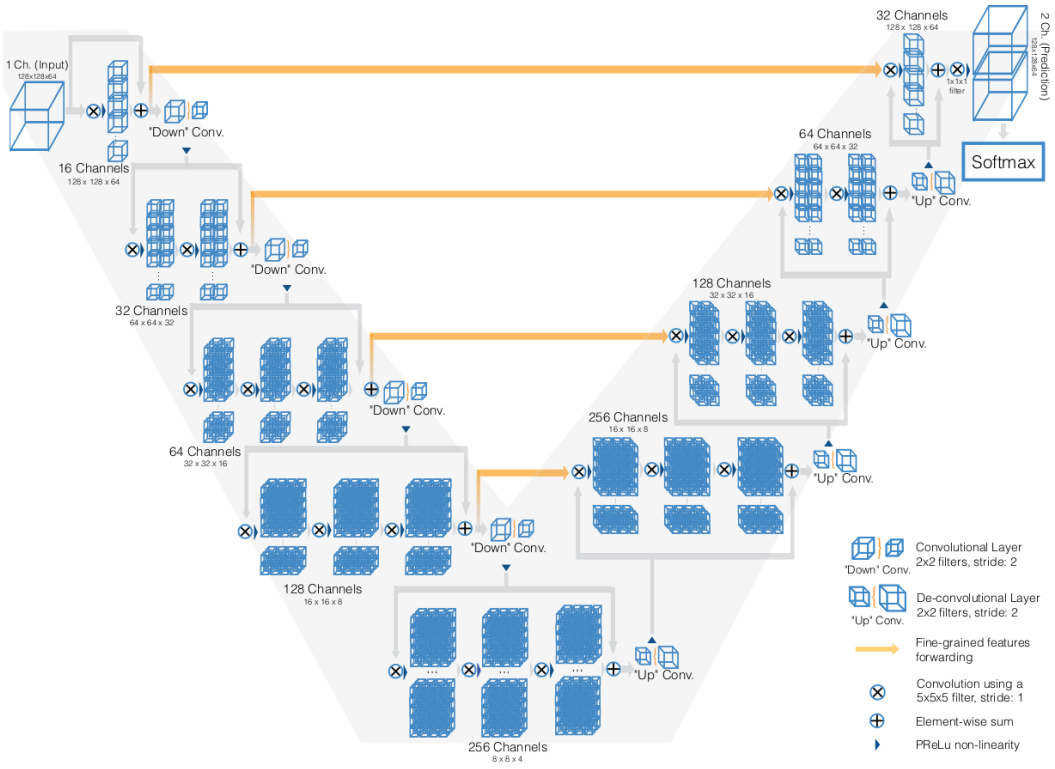


Figura 2.21: Arquitectura V-Net para segmentación 3D (Milletari et al., 2016).

Tabla 2.2: Matriz de confusión binaria

		Predictor	
		0 (-)	1 (+)
Experto	0 (-)	TN	FP
	1 (+)	FN	TP

- $\text{TP} = \sum_{p \in \Omega} G(p) \cdot s_\theta(p)$
- $\text{TN} = \sum_{p \in \Omega} (1 - G(p)) \cdot (1 - s_\theta(p))$
- $\text{FP} = \sum_{p \in \Omega} (1 - G(p)) \cdot s_\theta(p)$
- $\text{FN} = \sum_{p \in \Omega} G(p) \cdot (1 - s_\theta(p))$

A diferencia de las imágenes de escenas naturales (o de ambientes abiertos), adquiridas con sensores que capturan la luz reflejada y que suelen presentar alta resolución y contraste, las imágenes médicas presentan características particulares y a menudo adversas. Estas limitaciones derivan de la propiedades físicas de la tecnología de adquisición, donde las ondas electromagnéticas y los campos magnéticos deben atravesar tejidos de distinta composición y por consiguiente, de distintas densidades.

Debido a las dificultades mencionadas, este tipo de imágenes médicas suelen presentar diversos efectos adversos, como:

- Efecto de volumen parcial: En los bordes de órganos o lesiones, un píxel (o vóxel) puede representar más de un tipo de tejido o estructura, especialmente en imágenes de baja resolución espacial (cada píxel abarca un área muy grande).
- Bajo contraste: Dificultad para distinguir entre diferentes tejidos o estructuras debido a intensidades de gris similares.
- Solapamiento de distribuciones de intensidad de gris: Similitud entre las intensidades y texturas de la región de interés y del fondo (background).
- Baja resolución: Menor nivel de detalle en comparación con imágenes de escenas naturales.
- Desbalance de clases: La proporción de píxeles (vóxeles) pertenecientes a la clase de interés (por ejemplo, lesión) suele ser mucho menor que la de la clase fondo (background).

Por tanto, el diseño y selección de funciones de pérdida para la tarea de segmentación de imágenes médicas debe considerar estas dificultades. De acuerdo con Jadon (Jadon, 2020), las funciones de pérdidas pueden clasificarse en tres grandes grupos: basadas en distribución, basadas en regiones y basadas en bordes.

2.5.1. Funciones de pérdida basadas en distribución

Este tipo de funciones de pérdida se caracterizan por incorporar en su formulación conceptos derivados de las funciones de distribución de probabilidad. Es decir, miden la discrepancia entre la distribución de probabilidad predicha por la red neuronal y la distribución de probabilidad real dada por el experto a través del ground truth.

Función de pérdida de entropía cruzada binaria

La función de pérdida de entropía cruzada binaria (Binary Cross-Entropy, BCE) se define como (Yi-de et al., 2004):

$$L_{\mathrm{BCE}}(G, s_\theta) = -\sum_{p\in\Omega} \big[G(p)\cdot \log(s_\theta(p)) + \big(1-G(p)\big)\cdot \log\big(1-s_\theta(p)\big)\big], \tag{2.16}$$

donde su formulación proviene de la teoría de la información y mide la disimilitud entre la distribución de probabilidad real $G(p)$ y la distribución estimada $s_\theta(p)$ mediante el término de entropía cruzada.

El uso de la función de pérdida BCE en la tarea de segmentación de lesiones de esclerosis múltiple (EM) presenta efectos adversos, debido a que los conjuntos de datos de entrenamiento suelen estar afectados por un marcado desbalance de clases. En específico, las lesiones suelen ocupar menos del 1 % del volumen cerebral total. Como consecuencia, la BCE tiende a sesgar la predicción de la red hacia la clase mayoritaria (background), lo que conlleva a una subsegmentación de las lesiones pequeñas y, por ende, a una menor sensibilidad del modelo.

Función de pérdida de entropía cruzada binaria ponderada

La entropía cruzada binaria ponderada (Weighted Binary Cross-Entropy, WBCE) corresponde a una variante de la función de pérdida BCE (Yi-de et al., 2004), cuyo objetivo es mitigar el desbalance de clases presente en los conjuntos de datos de segmentación.

En la versión ponderada (Weighted o Balanced BCE), se introduce el parámetro $\beta \in [0, 1]$ que asigna mayor peso a la clase minoritaria, equilibrando así la contribución de ambas clases a la pérdida total (Xie and Tu, 2015). La WBCE se define como:

$$L_{\mathrm{WBCE}}(G, s_\theta) = -\sum_{p\in\Omega} \Big[\beta\cdot G(p)\cdot \log(s_\theta(p)) + (1-\beta)\cdot \big(1-G(p)\big)\cdot \log\big(1-s_\theta(p)\big)\Big], \tag{2.17}$$

donde típicamente: $\beta = \frac{\sum_{p\in\Omega}(1-G(p))}{|\Omega|}$ de modo que el peso asignado a la clase positiva (lesión) sea inversamente proporcional a su frecuencia relativa. A pesar de esta mejora, la WBCE no logra capturar información espacial ni correlaciones entre vóxeles adyacentes, por lo que suele obtener resultados subóptimos en segmentación de imágenes, especialmente en lesiones pequeñas o en aquellas con bordes difusos (Cui et al., 2019). Su uso es más frecuente en tareas de clasificación o detección donde el contexto espacial no es determinante.

Focal Loss

La función de pérdida Focal Loss, introducida en (Lin et al., 2020), fue diseñada con el objetivo de abordar el problema de desbalance de clases en detectores de objetos de una etapa. Esta función de pérdida pondera de manera dinámica la contribución de cada ejemplo a la pérdida total durante el entrenamiento. De acuerdo a Eq.(2.18) es posible notar que Focal loss es una generalización de la función de pérdida de entropía cruzada binaria

$$L_{FL}(G, s_\theta) = \sum_{p \in \Omega} \left[-G(p) \cdot (1 - s_\theta(p))^\gamma \cdot \log(s_\theta(p)) - (1 - G(p)) \cdot s_\theta(p))^\gamma \cdot \log(1 - s_\theta(p)) \right], \tag{2.18}$$

donde los términos $(1 - s_\theta(p))^\gamma$ y $s_\theta(p)^\gamma$ disminuyen el peso asignado a los ejemplos con errores pequeños, pero aumentándolo en los ejemplos con mayor error de predicción. Este mecanismo descansa bajo el supuesto que los errores de mayor amplitud son obtenidos en los ejemplos pertenecientes a la clase de menor representación en la imagen y los errores de menor amplitud corresponden a los ejemplos de la clase con mayor representación.

En el contexto de segmentación de imágenes médicas, Focal Loss consigue mitigar el efecto del desbalance de clases extremo, por ejemplo, en la segmentación de lesiones pequeñas, aunque su desempeño puede verse limitado al no considerar explícitamente la dependencia espacial entre vóxeles vecinos.

2.5.2. Funciones de pérdida basadas en región

Las funciones de pérdida basadas en región evalúan el desempeño del modelo cuantificando el grado de solapamiento entre la segmentación predicha y el ground truth. A diferencia de las funciones basadas en distribución, este tipo de pérdidas tienden a ser más robustas frente al desbalance de clases, ya que se centran en las propiedades espaciales de las regiones segmentadas.

Sensitivity-Specificity Loss

La función de pérdida Sensitivity-Specificity Loss, introducida en (Brosch et al., 2015), se formula como una combinación lineal convexa de los complementos de las métricas sensi-

bilidad (recall) y especificidad, de acuerdo a la siguiente expresión:

$$\begin{aligned} L_{ss}(G, s_\theta) &= w \cdot (1 - \text{sensibilidad}) + (1 - w) \cdot (1 - \text{especificidad}) \\ &= w \cdot \frac{\sum_{p \in \Omega} G(p) \cdot (1 - s_\theta(p))}{\sum_{p \in \Omega} G(p)} + (1 - w) \cdot \frac{\sum_{p \in \Omega} (1 - G(p)) \cdot s_\theta(p)}{\sum_{p \in \Omega} (1 - G(p))}, \end{aligned} \tag{2.19}$$

donde sensibilidad $= \frac{TP}{TP+FN}$ y especificidad $= \frac{TN}{TN+FP}$. El parámetro $w \in [0, 1]$ corresponde al peso que permite equilibrar la importancia relativa de ambos términos. Por lo general, se asigna un mayor peso al término sensibilidad, dado que contiene a los falsos negativos, cuya frecuencia relativa aumenta en presencia de la clase minoritaria de interés, como lo es la clase lesión de esclerosis múltiple.

Dice Loss

La función de pérdida Dice corresponde al complemento del coeficiente Dice (también conocido como índice de Sørensen-Dice, coeficiente de similitud de Dice o F_1-score), ampliamente utilizado en tareas de segmentación y clasificación de imágenes médicas.

El coeficiente Dice, cuantifica el grado de solapamiento espacial entre dos conjuntos: el ground truth G y la predicción S_θ. Este coeficiente puede expresarse como:

$$\begin{aligned} \text{Dice}(G, s_\theta) &= \frac{\sum_{p \in \Omega} G(p) \cdot s_\theta(p)}{\sum_{p \in \Omega} G(p) \cdot s_\theta(p) + \frac{1}{2} \cdot \sum_{p \in \Omega} s_\theta(p) \cdot (1 - G(p)) + \frac{1}{2} \cdot \sum_{p \in \Omega} G(p) \cdot (1 - s_\theta(p))} \\ &= \frac{2 \cdot \sum_{p \in \Omega} G(p) \cdot s_\theta(p)}{2 \cdot \sum_{p \in \Omega} G(p) \cdot s_\theta(p) + \sum_{p \in \Omega} s_\theta(p) \cdot (1 - G(p)) + \sum_{p \in \Omega} G(p) \cdot (1 - s_\theta(p))}, \end{aligned} \tag{2.20}$$

donde un valor $\text{Dice}(G, s_\theta) = 0$ indica total ausencia de intersección espacial entre los conjuntos y $\text{Dice}(G, s_\theta) = 1$ representa una coincidencia perfecta entre ambos conjuntos.

De este modo, a partir de esta métrica de similitud, la función de pérdida Dice Loss se

define como:

$$L_{\text{Dice}}(G, s_\theta) = 1 - \frac{2 \cdot \sum_{p \in \Omega} G(p) \cdot s_\theta(p)}{2 \cdot \sum_{p \in \Omega} G(p) \cdot s_\theta(p) + \sum_{p \in \Omega} s_\theta(p) \cdot (1 - G(p)) + \sum_{p \in \Omega} G(p) \cdot (1 - s_\theta(p))}, \tag{2.21}$$

dado que el coeficiente Dice es una medida de similitud, Dice Loss corresponde al complemento, donde $L_{Dice}(G, s_\theta) = 1 - \text{Dice}(G, s_\theta)$. Esta función de pérdida fue propuesta en (Milletari et al., 2016) en conjunto con el desarrollo de la red V-Net, demostrando gran efectividad en escenarios de alto desbalance de clases.

Tversky Loss

La función de pérdida Tversky Loss, introducida en (Hashemi et al., 2018), fue diseñada para mitigar el problema del desbalance de clases presente en imágenes médicas. Al igual como con la función de pérdida Dice Loss, esta función se construye a partir del complemento del índice de similitud asimétrico de Tversky (Tversky, 1977). Dicho índice generaliza el coeficiente Dice mediante la incorporación de los parámetros α y β, los cuales permiten ponderar de manera diferenciada los falsos positivos y falsos negativos, respectivamente. La función se define como:

$$L_{\text{Tversky}}(G, s_\theta) = 1 - \frac{\sum_{p \in \Omega} G(p) \cdot s_\theta(p)}{\sum_{p \in \Omega} G(p) \cdot s_\theta(p) + \alpha \cdot \sum_{p \in \Omega} s_\theta(p) \cdot (1 - G(p)) + \beta \cdot \sum_{p \in \Omega} G(p) \cdot (1 - s_\theta(p))}, \tag{2.22}$$

donde es posible observar que para los valores de parámetros $\alpha = 0.5$ y $\beta = 0.5$ Tversky Loss se reduce a la Dice Loss. Ajustando estos parámetros es posible dar mayor peso a la penalización de los falsos negativos, por ejemplo, eligiendo $\beta > \alpha$, de este modo, contribuyendo a mitigar los efectos negativos del desbalance de la clase de interés.

Generalized Dice Loss

Sudre et al. propusieron la Generalized Dice Loss (GDice) con el objetivo de aumentar la robustez frente a grandes desbalances de clases y de mejorar la detección de lesiones pequeñas

(Sudre et al., 2017). En esta función de pérdida se introducen pesos específicos para cada clase, definidos de manera inversamente proporcional a su cardinalidad (frecuencia absoluta), de modo que las clases menos representadas contribuyen proporcionalmente más a la pérdida total. La función de pérdida Dice generalizada para dos clases se define como:

$$L_{\text{GDice}}(G, s_\theta) = 1 - 2; \frac{\sum_{l=0}^{1} w_l \sum_{p\in\Omega} G_l(p), s_{\theta,l}(p)}{\sum_{l=0}^{1} w_l \sum_{p\in\Omega} \big(G_l(p) + s_{\theta,l}(p)\big)}, \tag{2.23}$$

donde los pesos por clase se definen como $w_l = 1/\big(\sum_{p\in\Omega} G_l(p)\big)^2$ y el índice l denota la clase (por ejemplo, $l = 0$ para el fondo y $l = 1$ para la clase positiva).

De este modo, al reducir la influencia de las clases mayoritarias y fortalecer el aprendizaje de las clases minoritarias, Generalized Dice Loss resulta adecuada para tareas de segmentación médica multiclase o en contextos con un alto desbalance entre clases.

Asymmetric Similarity Loss

En (Hashemi et al., 2019) se propone la función de pérdida Asymmetric Similarity Loss con el objetivo de abordar el problema del desbalance de clases en la tarea de segmentación de imágenes médicas. Esta función de pérdida se basa en el índice F_β, correspondiente a una generalización de $\mathrm{F}_1 -$ score, que a su vez, es un caso especial del índice Tversky cuando se cumple $\alpha + \beta = 1$.

Mediante la ponderando de los términos precisión y recall con $= \frac{1}{1+\beta^2}$ y $\frac{\beta^2}{1+\beta^2}$ respectivamente, es posible obtener

$$F_\beta = (1 + \beta^2)\frac{precision \times recall}{\beta^2 \times precision + recall}, \tag{2.24}$$

cuyo desarrollo algebraico, utilizando los términos provenientes de la tabla de confusión, permite notar que la ponderación aplicada a los falsos negativos es mayor que la aplicada a los falsos positivos en una proporción de β^2. De este modo, esta función de pérdida favorece la detección de la clase minoritaria como lesiones.

De este modo, Asymmetric Similarity Loss es definida por los autores como:

$$L_{F_\beta}(G, s_\theta) = 1 - \frac{(1+\beta^2)\sum_{p\in\Omega} G(p)\cdot s_\theta(p)}{(1+\beta^2)\sum_{p\in\Omega} G(p)\cdot s_\theta(p) + \beta^2\sum_{p\in\Omega} G(p)\cdot(1-s_\theta(p)) + \sum_{p\in\Omega} s_\theta(p)\cdot(1-G(p))}. \tag{2.25}$$

Focal Tversky Loss

La función de pérdida Focal Tversky Loss fue propuesta como una generalización de la función de pérdida Tversky Loss (Abraham and Khan, 2019), cuyo objetivo fue el de mejorar el desempeño en presencia de alto desbalance de clases. Su formulación es la siguiente:

$$L_{\mathrm{FT}}(G, s_\theta) = \sum_{c\in C}(1 - L_T^c(G, s_\theta))^{1/\gamma}, \tag{2.26}$$

donde C corresponde al conjunto de etiquetas de las clases de interés presentes en la imagen (la clase background puede ser excluida), $|C|$ es su cardinalidad, y el parámetro γ mediante el cual es posible concentrar la penalización en las clases con menor desempeño en el índice Tversky. Esto contrasta con Focal loss, donde la penalización se concentra a nivel de ejemplos.

Exponential Logarithmic Loss

La función de pérdida Exponential Logarithmic Loss fue propuesta como una combinación lineal convexa entre la función de pérdida Exponential Logarithmic Dice Loss y Binary Cross Entropy. A estos términos se les agrega un exponente, que por simplicidad se suele utilizar como $\gamma = \gamma_{Dice} = \gamma_{BCE}$. La función se define como:

$$L_{\mathrm{Exp}}(G, s_\theta) = w_{\mathrm{Dice}}\cdot L_{\mathrm{Dice}}(G, s_\theta) + w_{\mathrm{BCE}}\cdot L_{\mathrm{BCE}}(G, s_\theta), \tag{2.27}$$

donde $L_{\mathrm{Dice}}(G, s_\theta) = -\ln(\mathrm{Dice})^{\gamma_{\mathrm{Dice}}}$, $L_{\mathrm{BCE}}(G, s_\theta) = \mathrm{BCE}(G, s_\theta)^{\gamma_{\mathrm{BCE}}}$, y w_{Dice} y w_{BCE} son los coeficientes de ponderación que satisfacen $w_{\mathrm{Dice}} + w_{\mathrm{BCE}} = 1$.

De acuerdo a los autores (Wong et al., 2018), la función de pérdida exponential logarithmic loss mejora los resultados de segmentación de pequeñas estructuras.

2.5.3. Funciones de pérdida basadas en borde

En los últimos años, los mapas de distancia se han incorporado de manera exitosa en el diseño de funciones de pérdida para la tarea de segmentación de imágenes (Karimi and Salcudean, 2020; Kervadec et al., 2021; Ma et al., 2020). Un mapa de distancias corresponde a la transformación del ground truth en un mapa cuyas intensidades representan la distancia euclidiana de cada coordenada al borde más cercano del objeto de interés (foreground).

El Mapa de Transformación de Distancia (DTM) asociado a un objeto G[1] se denota como G_{DTM}. Siguiendo la formulación unificada presentada por Ma et al. (2020), este mapa se define formalmente para cada posición $p \in \Omega$ como:

$$G_{\text{DTM}}(p) = \begin{cases} \inf\limits_{z \in \text{GB}} \{\|p - z\|_2\} & \text{si } p \in G_{\text{in}} \cup G_{\text{out}}, \\ 0 & \text{si } p \in \text{GB}, \end{cases} \tag{2.28}$$

donde $\|\cdot\|_2$ es la distancia euclidiana, $\text{GB} \subset \Omega$ corresponde al conjunto de coordenadas del borde del objeto, y $G_{\text{in}}, G_{\text{out}} \subset \Omega$ representan las regiones interior y exterior respectivamente.

A diferencia del DTM, el Mapa de Distancia con Signo (SDF) codifica la dirección relativa al borde de la lesión. Este mapa se denota como G_{SDF} y, de acuerdo con la estandarización de Ma et al. (2020), se define como:

$$G_{\text{SDF}}(p) = \begin{cases} -\inf\limits_{z \in \text{GB}} \{\|p - z\|_2\} & \text{si } p \in G_{\text{in}}, \\ 0 & \text{si } p \in \text{GB}, \\ \inf\limits_{z \in \text{GB}} \{\|p - z\|_2\} & \text{si } p \in G_{\text{out}}. \end{cases} \tag{2.29}$$

De este modo, como referencia de ubicación, G_{SDF} asigna valores negativos a las posiciones dentro del objeto de interés y valores positivos fuera de él.

Las funciones de pérdida basadas en bordes, independientemente si éstas utilizan mapas de distancia en sus formulaciones, se suelen ver enfrentadas a problemas de inestabilidad durante las etapas iniciales del entrenamiento. Esto se debe a la alta variabilidad de las predicciones en las primeras épocas, especialmente en la delineación de los bordes. Observaciones empíricas mostraron que la aplicación de funciones de pérdida basadas en bordes en la gran mayoría de las corridas experimentales condujo al fenómeno de gradiente explosivo.

[1]Este concepto es análogo si se aplica sobre la máscara de predicción P obtenida desde una CNN.

Para mitigar este problema, las funciones de pérdida basadas en borde son implementadas en una combinación lineal convexa con una función de pérdida basadas en región como Dice Loss (Eq.(2.21)) o Generalized Dice Loss (Eq.(2.23)). La combinación utilizada está dada por la siguiente expresión:

$$L = \epsilon \cdot L_{\text{region}} + (1 - \epsilon) \cdot L_{\text{borde}}, \tag{2.30}$$

donde el parámetro ϵ, después de cada época, disminuye de manera lineal desde su valor inicial 1.0. De esta manera, durante las primeras etapas la penalización total estará dominada por el término basado en región, lo cual dota de estabilidad al entrenamiento. A medida que ϵ disminuye, el término basado en borde adquiere mayor influencia ya que su peso se incrementa de una manera gradual y lineal, mientras disminuye el del término basado en región.

Hausdorff Distance Loss

La función de pérdida Hausdorff Distance Loss fue diseñada con el objetivo de reducir la distancia de Hausdorff, la cual es un métrica muy utilizada para cuantificar la discrepancia entre los bordes de la segmentación del modelo y del ground truth (Karimi and Salcudean, 2020).

La distancia de Hausdorff unidireccional entre dos conjuntos de puntos (u objetos) X e Y, se define como la máxima distancia desde un punto de X al punto más cercano en Y:

$$\text{hd}(X, Y) = \max_{x \in X} \min_{y \in Y} \|x - y\|_2. \tag{2.31}$$

Dado que esta distancia unidireccional no es simétrica, se define la distancia de Hausdorff bidireccional (o simplemente distancia de Hausdorff) como el máximo de las dos distancias de Hausdorff unidireccionales

$$\text{HD}(X, Y) = \max(\text{hd}(X, Y), \text{hd}(Y, X)). \tag{2.32}$$

En (Karimi and Salcudean, 2020) los autores proponen la función de pérdida Hausdorff Distance Loss integrando los mapas de transformación de distancia DTM tanto del ground truth G como de la segmentación binaria del modelo P ($P = s_\theta > 0.5$)

$$L_{\text{HD}}(G, s_\theta) = (1 - \epsilon) \cdot L_{\text{Dice}} + \epsilon \cdot \frac{1}{|\Omega|} \sum_{p \in \Omega} \left[(G(p) - s_\theta(p))^2 \cdot (G^\alpha_{\text{DTM}}(p) + P^\alpha_{\text{DTM}}(p)) \right], \tag{2.33}$$

donde α es el parámetro a ajustar cuyo valor recomendado por los autores es 2.0.

Boundary Loss

La función de pérdida Boundary Loss fue propuesta en (Kervadec et al., 2021) con el objetivo de abordar el problema del alto desbalance de clases presente en la segmentación de imágenes médicas. A diferencia de las funciones de pérdida basadas en regiones (por ejemplo, Dice Loss), las cuales penalizan de manera uniforme sobre todos los vóxeles, Boundary Loss pondera la penalización de cada píxel según su distancia al borde más cercano del ground truth. Es decir, los vóxeles mal clasificados que se encuentran lejos de los bordes son penalizados con mayor intensidad, lo cual induce al modelo a disminuir los falsos positivos. Los autores definen Boundary Loss como:

$$L_{\mathrm{B}}(G, s_\theta) = \epsilon \cdot L_{\mathrm{GDice}} + (1-\epsilon) \cdot \frac{1}{|\Omega|} \sum_{p \in \Omega} G_{\mathrm{SDF}}(p)\ s_\theta(p), \tag{2.34}$$

donde G_{SDF} corresponde a la función de distancia con signo (SDF) del ground truth.

Dado que esta ponderación es sobre la distancia al borde más cercano, el desbalance de clases deja de influir de forma significativa en el cálculo de la pérdida. Sin embargo, esta formulación presenta una limitación teórica importante que fue observada de manera empírica. Dado que dentro de las lesiones los valores de $G_{\mathrm{SDF}}(p)$ son negativos, cuando la segmentación es correcta y el modelo predice valores cercanos a uno en el interior de las lesiones y cercanos a cero en el fondo, el producto $G_{\mathrm{SDF}}(p)\, s_\theta(p)$ aporta un valor negativo a la sumatoria. Este comportamiento puede producir valores de pérdida negativos, lo cual resulta indeseable dado que afecta la estabilidad del proceso y dificulta la interpretación de la magnitud del error durante el entrenamiento.

Es más, este comportamiento pone en duda la clasificación formal de la Boundary Loss como una función de pérdida estricta, ya que, de acuerdo con los fundamentos presentados en la Sección 2.3, no satisface la condición de no-negatividad, la cual establece que la imagen de una función de costo debe ser siempre mayor o igual a cero. La ausencia de una cota inferior permite que la optimización continúe indefinidamente hacia valores negativos, provocando inestabilidad numérica y gradientes explosivos. Esto conlleva a que, el error termine aumentando en lugar de converger, un comportamiento anómalo que fue observado empíricamente en las últimas épocas del entrenamiento.

Boundary-Sensitive Loss

La función de pérdida Boundary-Sensitive Loss fue propuesta en (Du et al., 2023) con el propósito de abordar el desbalance intra-clase presente dentro del objeto de interés (foreground). Este desbalance fue mitigado mediante la penalización de los falsos positivos y falsos negativos ubicados tanto en el interior como en los bordes del ground truth (G) y de la predicción (P). Las ponderaciones para la predicción y el ground truth se definen como:

$$\mathrm{WF}^P = \alpha \cdot (\mathrm{FBP}_{\mathrm{in}} + \mathrm{FBG}_{\mathrm{out}}) + (1-\alpha) \cdot \mathrm{FIP}, \tag{2.35}$$

$$\mathrm{WF}^G = \alpha \cdot (\mathrm{FBP}_{\mathrm{out}} + \mathrm{FBG}_{\mathrm{in}}) + (1-\alpha) \cdot \mathrm{FIG}, \tag{2.36}$$

donde un $\alpha > 0.5$ implica una mayor penalización a los errores cometidos en los bordes del ground truth (FBG) y de la predicción (FBP), en comparación con los vóxeles pertenecientes al interior del ground truth (FIG) y de la predicción (FIP). Este tipo de ponderación es implementado en la función de pérdida Dice, reemplazando los términos de falsos positivos y falsos negativos. La nueva formulación presentada por los autores es:

$$L_{\mathrm{BS}} = 1 - \frac{2 \cdot \mathrm{TIG}}{2 \cdot \mathrm{TIG} + \mathrm{WF}^P + \mathrm{WF}^G}. \tag{2.37}$$

La implementación final de esta función de pérdida incluye un término adicional denominado location constraint, mediante el cual se busca minimizar las diferencias entre G y P en las frecuencias positivas (vóxeles de clase foreground) a lo largo de los ejes horizontal y vertical:

$$L = L_{\mathrm{BS}} + \beta \cdot C_{\mathrm{Loc}}, \tag{2.38}$$

donde:

$$C_{\mathrm{Loc}} = \sum_{i=1}^{H} \left| \sum_{j=1}^{W} s_\theta(i,j) - \sum_{j=1}^{W} G(i,j) \right| + \sum_{j=1}^{W} \left| \sum_{i=1}^{H} s_\theta(i,j) - \sum_{i=1}^{H} G(i,j) \right|. \tag{2.39}$$

Active Boundary Loss

En (Wang et al., 2021), los autores proponen la función de pérdida Active Boundary Loss (ABL), cuyo objetivo es mejorar el alineamiento entre los bordes predichos (PB) y los bordes ground truth (GB). Esta función de pérdida es formulada como un problema de predicción del vector de dirección desde los vóxeles de PB hacia el vóxel borde más cercano en GB, de este modo guiando el movimiento de PB en cada iteración del entrenamiento.

Para conseguir este objetivo, ABL minimiza la entropía cruzada entre los vectores de dirección ground truth (D_u^g) y los vectores de dirección predichos (D_u^p) para los vóxeles $u \in \text{PB}$. El vector D_u^g corresponde a un one-hot vector definido como $D_u^g = \Phi(\arg\min_j G_{\text{DTM}}(u+\triangle_j),\ j \in \{0,1,\ldots,7\}$, donde $\triangle = \{\{1,0\},\{-1,0\},\{0,-1\},\{0,1\},\{-1,1\},\{-1,-1\},\{1,-1\}\}$ y Φ es una función que convierte el índice j en un one-hot vector. El vector D_u^p se obtiene utilizando la divergencia Kullback-Leibler (KL) entre la distribución de probabilidad de las clases en el píxel u y la de sus ocho vecinos en una ventana 3×3. Esta divergencia se normaliza mediante una función softmax para generar una distribución de probabilidad sobre las direcciones posibles:

$$D_u^p = \left\{ \frac{e^{\text{KL}(s_\theta(u), s_\theta(k))}}{\sum_{m=0}^{7} e^{\text{KL}(s_\theta(u), s_\theta(m))}}, k \in \{0,1,\ldots,7\} \right\}. \tag{2.40}$$

La función de pérdida Active Boundary Loss se define entonces para los vóxeles pertenecientes al conjunto PB como:

$$L(\text{GB}, s_\theta) = \frac{1}{|\text{PB}|} \sum_{u \in \text{PB}} \Lambda(G_{\text{DTM}}(u))\, \text{CE}(D_u^p, D_u^g), \tag{2.41}$$

donde $|\text{PB}|$ es la cantidad de vóxeles en PB, $\Lambda(G_{\text{DTM}}(u))$ es una función del mapa de distancias de GB que pondera el término de entropía cruzada (CE) por la distancia entre $G_{\text{DTM}}(u)$ y un valor máximo predefinido. De este modo, al minimizar la función de pérdida se busca incrementar la divergencia KL entre la distribución de probabilidad del píxel u y la correspondiente a su borde más cercano $j \in \text{PG}$, y simultaneamente se busca reducir la divergencia KL entre u y los vóxeles vecinos $j \in u + \triangle$.

Al igual que con Hausdorff Distance Loss y Boundary Loss, la implementación final de Active Boundary Loss es realizada como una combinación lineal convexa con la función Generalized Dice Loss:

$$L_{\text{AB}}(G, s_\theta, \text{GTB}) = (1-\epsilon) \cdot L_{\text{GDice}} + \epsilon \cdot \frac{1}{|N_b|} \sum_{u \in N_b} \Lambda(G_{\text{DTM}}(u))\, \text{CE}(D_u^p, D_u^g). \tag{2.42}$$

Conditional Boundary Loss

La función de pérdida Conditional Boundary Loss (CBL) fue propuesta en (Wu et al., 2023). Al igual que la función Active Boundary Loss, CBL fue diseñada con el objetivo de mejorar el alineamiento entre los bordes predichos (PB) y los bordes ground truth (GB). Para lograr esto, la función considera tres pares de vóxeles para cada vóxel borde $u = (i, j) \in \text{PB}$.

La selección de estos pares de vóxeles se realiza mediante el algoritmo de muestreo Conditional Correctness-Aware Sampling (CCAS), propuesto por los mismos autores. Este algoritmo selecciona vóxeles que cumplen dos condiciones: deben estar correctamente clasificados y pertenecer al vecindario 5×5 del vóxel de borde actual u. De este modo, CCAS identifica dos tipos de vóxeles para cada vóxel de borde u: (a) Muestras positivas $z^+ \in Z_u^+$, correspondientes a vóxeles correctamente clasificados pertenecientes a la misma clase que u, y (b) Muestras negativas $z^- \in Z_u^-$, correspondientes a vóxeles correctamente clasificados pertenecientes a una clase diferente a la de u.

La función de pérdida CBL esta definida como:

$$L_{\mathrm{CB}}(G, s_\theta, \mathrm{GB}) = (1-\epsilon) \cdot L_{\mathrm{GDice}} + \epsilon \cdot (L_{\mathrm{A2C}} + \beta L_{\mathrm{A2P\&N}}), \tag{2.43}$$

donde el primer término L_{A2C} mide la discrepancia entre cada vóxel de borde u y el representante local de su misma clase, correspondiente al vector promedio $e_u \in \mathbb{R}^D$ de las características (intensidades) de la imagen de entrada en todos sus D canales $e_u = \frac{1}{|Z_u^+|} \sum_{z^+ \in Z_u^+} z^+$. Estos dos términos se definen como:

$$L_{\mathrm{A2C}} = L_{\mathrm{A2C}}^{\mathrm{pair}} + \alpha L_{\mathrm{A2C}}^{\mathrm{SCE}}, \tag{2.44}$$

$$L_{\mathrm{A2C}}^{\mathrm{pair}} = \frac{1}{|\mathrm{PB}|} \sum_{u \in \mathrm{PB}} 1\{|Z_u^+| \geq 1\} \cdot \|u - e_u\|_2, \tag{2.45}$$

$$L_{\mathrm{A2C}}^{\mathrm{SCE}} = \frac{1}{|\mathrm{PB}|} \sum_{u \in \mathrm{PB}} \mathrm{CE}(s_\theta(u), G(u)), \tag{2.46}$$

donde $\| \cdot \|_2$ es la norma euclidiana, $\mathrm{CE}(\cdot, \cdot)$ es la función de entropía cruzada, y $G(u)$ es la etiqueta del vóxel u. El término $L_{\mathrm{A2C}}^{\mathrm{SCE}}$ promueve la similitud entre los pares de vóxeles conformados entre el vóxel borde u y los vóxeles de su misma clase (A2P), y por otro lado, promueve la disimilitud entre pares de vóxeles conformados entre el vóxel borde u y vóxeles de clases diferentes (A2N), dentro del vecindario 5×5 de u:

$$L_{\mathrm{A2P\&N}} = \frac{1}{|N_b|} \sum_{u \in \mathrm{PB}} \frac{1}{|Z_u^+| + |Z_u^-|} \sum_{z \in Z_u^+ \cup Z_u^-} (a_{uz} - \hat{a}_{uz})^2, \tag{2.47}$$

donde, $\hat{a}_{uz} = \frac{u \cdot z}{\|u\|_2 \|z\|_2}$ es la similitud del coseno y $a_{uz} = \begin{cases} 1 & \text{si} \quad z \in Z_u^+, \\ 0 & \text{si} \quad z \in Z_u^-, \end{cases}$ la etiqueta objetivo.

2.6. Preprocesamiento de MRI cerebrales

Previo al entrenamiento de los modelos de redes neuronales convolucionales, se realiza el preprocesamiento de los conjuntos de datos. El objetivo del preprocesamiento es estandarizar las imágenes de resonancia magnética cerebral que serán utilizadas en el proceso de entrenamiento y en la predicción (Zhou et al., 2019). Este preprocesamiento esta compuesto por dos etapas principales, las que a su vez se componen de varias subetapas que transforman la MRI de entrada.

La primera etapa del preprocesamiento consiste en una secuencia de pasos o transformaciones comúnmente utilizadas en la comunidad de procesamiento de imágenes de resonancia magnética cerebral (Carass et al., 2017; Zhou et al., 2019).

A continuación, se realiza una descripción de los principales pasos utilizados en el preprocesamiento:

- Corrección del Sesgo del Campo Magnético (Bias field correction): Corrige las variaciones en la intensidad gris de un mismo tejido en distintas zonas del cerebro, originadas por variaciones en la densidad y composición del tejido cerebral.
- Extracción Encefálica (Brain extraction o Skull-stripping): Se eliminan los vóxeles correspondientes al cráneo y otros tejidos como la duramadre y los globos oculares, manteniendo el cerebro, el cerebelo y el tronco encefálico.
- Registración rígida (Rigid registration): Transforma el volumen de entrada a un espacio de resolución y coordenadas estandar, específicamente al espacio MNI con resolución de $1\,\mathrm{mm}^3$. Este proceso utiliza transformaciones afines, caracterizadas por componentes lineales como escalado, rotación y una componente de traslación.

La segunda etapa de preprocesamiento comprende dos pasos adicionales: el truncamiento de las intensidades extremas y una transformación de las intensidades de todos los vóxeles. Este último paso es fundamental para homogenizar imágenes obtenidas con diferentes equipos o protocolos de adquisición, y puede realizarse mediante normalización, estandarización o ajuste de histogramas:

- Truncamiento: Las intensidades de los vóxeles se limitan (truncan) a un rango definido por un cuantil inferior y uno superior. Es decir, las intensidades menores al cuantil

inferior son actualizadas a la intensidad de gris de dicho cuantil. En cambio, las intensidades mayores al cuantil superior son actualizadas a ese límite. Este paso tiene como objetivo reducir el impacto de los valores atípicos (outliers) o ruido, de este modo mejorando la robustez de los algoritmos o modelos de aprendizaje automático.

- Normalización: consiste en escalar las intensidades de los vóxeles a un rango objetivo $[\text{máx}_{\text{obj}}, \text{mín}_{\text{obj}}]$. Este rango por lo general corresponde a los intervalos $[0, 1]$ o $[-1, 1]$. Su fórmulación es:

$$I_{\text{norm}} = (I - \text{mín}(I)) \cdot \frac{\text{máx}_{\text{obj}} - \text{mín}_{\text{obj}}}{\text{máx}(I) - \text{mín}(I)} + \underset{\text{obj}}{\text{mín}}, \tag{2.48}$$

donde I es la intensidad original del vóxel, $\text{mín}(I)$ y $\text{máx}(I)$ son las intensidades mínima y máxima del volumen de entrada, y mín_{obj} y máx_{obj} son los límites inferior y superior del rango objetivo de intensidades.

- Estandarización: transforma las intensidades de modo que el conjunto de datos resultante tenga media cero y una desviación estándar uno:

$$I_{\text{std}} = \frac{I - \mu}{\sigma}, \tag{2.49}$$

donde I, como en la normalización es la intensidad original del vóxel, μ es la media y σ es la desviación estándar de las intensidades del volumen.

- Ajuste de histogramas (Histogram matching): Corresponde a una transformación a nivel de distribución de probabilidad acumulada (F). La distribución de intensidades de una imagen se aproxima a una distribución de referencia mediante transformaciones lineales y no lineales (Dougherty, 2009). Por ejemplo, si la distribución de referencia es la distribución uniforme, a la transformación se le conoce como ecualización. Este método permite homogenizar las intensidades de gris de los mismos tipos de tejidos presentes en diferentes imágenes. Este método no consigue una buena aproximación si en la imagen existen zonas con alteraciones como tumores y lesiones de gran volumen respecto a la imagen de referencia.

2.7. Características Radiómicas

La radiómica es un área de investigación enfocada en la extracción de métricas cuantitativas a partir de imágenes médicas (Mayerhoefer et al., 2020). Para la caracterización de lesiones de EM en este trabajo, se utilizan dos tipos de información complementaria:

En primer lugar, se utiliza la información de gradiente, la cual es fundamental para identificar transiciones de intensidad abruptas, como por ejemplo en los bordes de objetos. Matemáticamente, para una imagen I, la magnitud del gradiente en cada coordenada espacial $p = (i, j)$ de un vóxel se define como la norma del vector gradiente (Gonzalez and Woods, 2001).

$$|\nabla I(i,j)| = \sqrt{\left(\frac{\partial I(i,j)}{\partial i}\right)^2 + \left(\frac{\partial I(i,j)}{\partial j}\right)^2} \tag{2.50}$$

En segundo lugar, y de manera principal, se utiliza la información de textura mediante métodos de segundo orden y orden superior, los cuales se detallan en las siguientes secciones.

2.7.1. Matriz de Co-ocurrencia de Niveles de Gris (GLCM)

Este tipo de características o descriptores de textura fue propuesto en (Haralick et al., 1973). La GLCM describe la textura de segundo orden analizando la periodicidad espacial de los niveles de gris. Se define como una matriz $P(i, j|d, \theta)$ que representa la probabilidad conjunta de que dos vóxeles con intensidades i y j aparezcan separados por una distancia d en la dirección θ.

A partir de la matriz normalizada $p(i, j)$, donde N_g es el número de niveles de gris, es posible calcular 14 descriptores. A continuación, se describen los más utilizados en la literatura:

- **Contraste:** Cuantifica las variaciones locales de intensidad. Valores altos indican cambios bruscos (bordes o texturas heterogéneas).

$$\text{Contraste} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(i-j)^2 p(i,j) \tag{2.51}$$

- **Energía:** Mide la uniformidad textural (suma de los cuadrados de las entradas de la

GLCM).

$$\text{Energía} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\{p(i,j)\}^2 \tag{2.52}$$

- **Homogeneidad:** Evalúa la cercanía de la distribución a la diagonal de la GLCM. Es inversa al contraste.

$$\text{Homogeneidad} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\frac{p(i,j)}{1+|i-j|} \tag{2.53}$$

- **Correlación:** Mide la dependencia lineal de los niveles de gris entre vóxeles vecinos.

$$\text{Correlación} = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(i-\mu_x)(j-\mu_y)p(i,j)}{\sigma_x\sigma_y} \tag{2.54}$$

donde μ_x, μ_y, σ_x y σ_y son las medias y desviaciones estándar marginales.

2.7.2. Matriz de Longitud de Rachas de Niveles de Gris (GLRLM)

Introducida en (Galloway, 1975), la GLRLM evalúa la textura mediante el análisis de "runs" o "rachas" de vóxeles consecutivos con la misma intensidad. Sea $p(i,j)$ el número de rachas con nivel de gris i y longitud j, N_r la longitud máxima de racha y N_z el número total de rachas (suma de los elementos de GLRLM). Aunque originalmente se propusieron 5 descriptores, extensiones posteriores y la estandarización actual (IBSI) han ampliado este conjunto a 16 métricas comúnmente utilizadas, entre las cuales destacan:

- **Short Run Emphasis (SRE):** Pondera las rachas cortas, siendo indicativo de texturas finas.

$$\text{SRE} = \frac{1}{N_z}\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}\frac{p(i,j)}{j^2} \tag{2.55}$$

- **Long Run Emphasis (LRE):** Pondera las rachas largas, asociado a texturas gruesas y regiones homogéneas.

$$\text{LRE} = \frac{1}{N_z}\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}p(i,j)\cdot j^2 \tag{2.56}$$

- **Gray Level Non-Uniformity (GLN):** Mide la variabilidad de los valores de intensidad en la imagen. Valores bajos indican que las intensidades son muy homogéneas.

$$\mathrm{GLN} = \frac{1}{N_z} \sum_{i=1}^{N_g} \left(\sum_{j=1}^{N_r} p(i,j) \right)^2 \tag{2.57}$$

- **Run Length Non-Uniformity (RLN):** Mide la similitud de la longitud de las rachas a lo largo de la imagen. Valores bajos indican que las longitudes de las rachas son muy uniformes.

$$\mathrm{RLN} = \frac{1}{N_z} \sum_{j=1}^{N_r} \left(\sum_{i=1}^{N_g} p(i,j) \right)^2 \tag{2.58}$$

- **Run Percentage (RP):** Mide la fracción de vóxeles que forman parte de una racha definida, indicando la granularidad de la textura. Siendo N_p el número total de vóxeles en la RoI:

$$\mathrm{RP} = \frac{N_z}{N_p} \tag{2.59}$$

2.8. Funciones de distancia

2.8.1. Distancia de Mahalanobis

La distancia de Mahalanobis es una métrica muy importante en análisis multivariado, mediante la cual se cuantifica la separación entre un elemento (vector) y otro elemento perteneciente o representante de un conjunto de datos multivariado. Una de sus aplicaciones más utilizadas es la detección de datos atípicos u outliers, es decir, datos que se alejen del representante de un conjunto de datos.

A diferencia de la distancia euclidiana, que pondera con el mismo peso a todas las dimensiones, la distancia de Mahalanobis considera la estructura de covarianza de los datos. Formalmente, para dos vectores (o puntos de datos) x_A y x_B, la distancia de Mahalanobis se define como (Varmuza and Filzmoser, 2009):

$$\mathrm{MD}(x_A, x_B) = \sqrt{(x_B - x_A)^\top \cdot \hat{\Sigma}_X^{-1} \cdot (x_B - x_A)}, \tag{2.60}$$

donde $\hat{\Sigma}_X^{-1}$ es la matriz de precisión estimada, definida como la inversa de la matriz de covarianzas muestral de los datos. Esta estimación de la matriz de precisión es fundamental,

ya que codifica la información sobre la varianza y correlación entre variables, utilizada como un peso a cada dimensión. De este modo, la distancia de Mahalanobis se vuelve invariante a la escala de las variables y a las correlaciones entre ellas, así transformando (escalando) el espacio de características en uno isotrópico. Lo anterior es una ventaja significativa cuando se trabaja con datos multivariados donde las unidades de medida o las interdependencias pueden variar considerablemente.

Distancia Euclidiana

De este modo, la distancia euclidiana corresponde a un caso particular de la distancia de Mahalanobis cuando no se considera la escala ni estructura de correlación entre los datos. Por tanto, en lugar de utilizar la matriz de precisión Σ_X^{-1}, se utiliza la matriz identidad I_D, donde D es la cantidad de características o dimensiones de los elementos del conjunto de datos

$$d(x_A, x_B) = \sqrt{(x_B - x_A)^\top I_D (x_B - x_A)} = \sqrt{(x_B - x_A)^\top (x_B - x_A)}. \tag{2.61}$$

2.9. Reducción de la dimensionalidad

Sea un conjunto de datos representados por la matriz de observaciones $\boldsymbol{X} \in \mathbb{R}^{m \times N}$, donde m corresponde a la cantidad de características y N a la cantidad de observaciones. Cuando m es grande, a menudo muchas de estas características pueden ser redundantes o irrelevantes para la predicción de la variable objetivo (Theng and Bhoyar, 2024). Reducir la dimensionalidad mientras se preserva la mayor cantidad de información puede disminuir el costo computacional, disminuir el sobreajuste y mejorar el rendimiento del modelo.

Las técnicas de reducción de la dimensionalidad se pueden agrupar en dos categorias (Jia et al., 2022): Selección de características y extracción de características. En selección de características se busca un subconjunto representativo de características originales. Por otro lado, la extracción de características genera nuevas características a partir de las originales. Mientras que la primera técnica mantiene la interpretabilidad, la segunda técnica permite una mayor compresión de la información, la cual puede ser a costa de una pérdida de explicabilidad.

Dado que en este trabajo la cantidad de características recolectadas es alta, alrededor

de 25, resulta adecuado aplicar una reducción de la dimensionalidad antes de estimar la matriz de precisión utilizada en la distancia de Mahalanobis. Considerando que en este caso la interpretabilidad no es un requisito, se decide trabajar con una técnica de extracción de características, específicamente Análisis de Componentes Principales (PCA), descrito en la siguiente subsección.

2.9.1. Análisis de Componentes Principales (PCA)

Sea $\boldsymbol{X} \in \mathbb{R}^{m \times N}$ la matriz de observaciones

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mN} \end{bmatrix}_{m \times N}, \tag{2.62}$$

donde m corresponde a la cantidad de características y N a la cantidad de ejemplos u observaciones. Se definen los vectores columna $\boldsymbol{x}_j = \begin{bmatrix} x_{1j} \dots x_{mj} \end{bmatrix}^\top \in \mathbb{R}^{m \times 1}$ y los vectores fila $\boldsymbol{x}^{(i)} = \begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{iN} \end{bmatrix} \in \mathbb{R}^{1 \times N}$. Cuya matriz de covarianzas es

$$\boldsymbol{\Sigma}_X = \frac{1}{N-1}(\boldsymbol{X} - \overline{\boldsymbol{X}})(\boldsymbol{X} - \overline{\boldsymbol{X}})^\top \in \mathbb{R}^{m \times m} \tag{2.63}$$

donde $\bar{\boldsymbol{x}} = \begin{bmatrix} \bar{x}_1 & \bar{x}_2 & \cdots & \bar{x}_m \end{bmatrix}^\top \in \mathbb{R}^{m \times 1}$, $\bar{x}_i = \frac{1}{N}\sum_{j=1}^{N} x_{ij}$, $\overline{\boldsymbol{X}} = \bar{\boldsymbol{x}}\, \mathbf{1}_N^\top \in \mathbb{R}^{m \times N}$ y cada término de la matriz de covarianzas está definido como

$$\sigma_{ik} = \mathrm{Cov}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(k)}) = \frac{1}{N-1}\sum_{j=1}^{N}\big(x_{ij} - \bar{x}_i\big)\big(x_{kj} - \bar{x}_k\big). \tag{2.64}$$

El objetivo de PCA es representar los datos contenidos en $\boldsymbol{X}$ en un nuevo sistema de coordenadas, en el cual las variables transformadas sean linealmente independientes entre sí, y al mismo tiempo, cada variable esté alineada con las direcciones de máxima variabilidad de los datos originales en la matriz $\boldsymbol{X}$. Estas direcciones corresponden a los vectores propios de la matriz de covarianzas $\boldsymbol{\Sigma}_X$. Para esto, en PCA se busca una base ortonormal $\boldsymbol{U} = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_m]$ que diagonalice $\boldsymbol{\Sigma}_X$

$$\boldsymbol{\Sigma}_Y = \boldsymbol{U}^\top \boldsymbol{\Sigma}_X \boldsymbol{U} = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_m \end{bmatrix} = \mathrm{diag}(\lambda_1, \dots, \lambda_m) \tag{2.65}$$

donde

$$\boldsymbol{\Sigma}_X \, \boldsymbol{u}_i = \lambda_i \, \boldsymbol{u}_i, \qquad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0, \tag{2.66}$$

y cada λ_i corresponde a la varianza explicada por la i-ésima componente principal. De este modo, las componentes principales (filas de $\boldsymbol{Y}$) no están correlacionadas. La proyección escalar de una observación $\boldsymbol{x}_j$ sobre un vector base $\boldsymbol{u}_i$, donde $\theta = \angle(\boldsymbol{u}_i, \boldsymbol{x}_j)$, es $y_{ij} = \|\boldsymbol{x}_j\| \cos\theta = \frac{\boldsymbol{u}_i^\top \boldsymbol{x}_j}{\|\boldsymbol{u}_i\|} = \boldsymbol{u}_i^\top \boldsymbol{x}_j$, dado que $\|\boldsymbol{u}_i\| = 1$. Por tanto, la i-ésima componente principal está dada por:

$$\boldsymbol{y}^{(i)} = \boldsymbol{u}_i^\top \boldsymbol{X} = \begin{bmatrix} \boldsymbol{u}_i^\top \boldsymbol{x}_1 & \cdots & \boldsymbol{u}_i^\top \boldsymbol{x}_N \end{bmatrix}. \tag{2.67}$$

La varianza en la dirección de cada vector propio de Σ_X, o lo que es lo mismo, la varianza que es explicada por cada componente principal es

$$\begin{aligned} \mathrm{Var}(\boldsymbol{u}_i^\top \boldsymbol{X}) &= \frac{1}{N-1} \boldsymbol{u}_i^\top (\boldsymbol{X} - \overline{\boldsymbol{X}})(\boldsymbol{X} - \overline{\boldsymbol{X}})^\top \boldsymbol{u}_i \\ &= \boldsymbol{u}_i^\top \boldsymbol{\Sigma}_X \boldsymbol{u}_i \\ &= \boldsymbol{u}_i^\top \lambda_i \boldsymbol{u}_i \\ &= \lambda_i. \end{aligned}$$

De este modo, las componentes principales se obtienen proyectando la matriz de datos $\boldsymbol{X}$ en las direcciones principales dadas por los vectores propios ortonormales:

$$\boldsymbol{Y} = \boldsymbol{U}^\top (\boldsymbol{X} - \overline{\boldsymbol{X}}). \tag{2.68}$$

Para reducir la dimensionalidad se conservan los k primeros vectores propios $\boldsymbol{U}_k = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k]$, $k < m$, y se trabaja con

$$\boldsymbol{Y}_k = \boldsymbol{U}_k^\top (\boldsymbol{X} - \overline{\boldsymbol{X}}) \ \in \ \mathbb{R}^{k \times N}. \tag{2.69}$$

La cantidad de componentes principales seleccionadas suele definirse en función de superar el umbral del 90 % de la varianza explicada acumulada. La proporción de la varianza explicada por las primeras k componentes principales está dada por

$$\sigma^2_{\mathrm{acum}}(k) = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{m} \lambda_j}. \tag{2.70}$$

En la práctica, PCA se implementa de forma estable y robusta mediante la descomposición en valores singulares (SVD) de $\boldsymbol{X}$:

$$(\boldsymbol{X} - \overline{\boldsymbol{X}}) = \boldsymbol{U} \, \boldsymbol{\Delta} \, \boldsymbol{V}^\top, \qquad \boldsymbol{\Delta} = \mathrm{diag}(\sigma_1, \ldots, \sigma_r), \quad \sigma_1 \geq \cdots \geq \sigma_r \geq 0, \tag{2.71}$$

de modo que

$$\boldsymbol{\Sigma}_X = \frac{1}{N-1}(\boldsymbol{X} - \overline{\boldsymbol{X}})(\boldsymbol{X} - \overline{\boldsymbol{X}})^\top = \boldsymbol{U}\left(\frac{\boldsymbol{\Delta}^2}{N-1}\right)\boldsymbol{U}^\top, \tag{2.72}$$

por tanto, la matriz de vectores singulares izquierdos $\boldsymbol{U}$, que está en el espacio de las características $m \times m$, corresponde a la matriz ortonormal con columnas correspondientes a los vectores propios de $\boldsymbol{\Sigma}_X$ utilizada en PCA. De este modo,

$$\boldsymbol{Y} = \boldsymbol{U}^\top(\boldsymbol{X} - \overline{\boldsymbol{X}}) = \boldsymbol{\Delta}\boldsymbol{V}^\top. \tag{2.73}$$

Los paquetes computacionales como scikit-learn calculan PCA mediante la descomposición SVD de $\boldsymbol{X}$, a través de algoritmos que evitan formar de manera explícita la matriz de covarianzas. Para esto utilizan métodos numéricos basados en la biblioteca LAPACK (Anderson et al., 1999), basados en factorizaciones QR y procesos de bidiagonalización, los cuales ofrecen una mayor estabilidad y eficiencia computacional, sobre todo en conjuntos de datos de alta dimensionalidad (Halko et al., 2011). De este modo, el cómputo es más robusto frente a la presencia de colinealidades entre variables y errores de redondeo.

Finalmente, es importante destacar que PCA es sensible a la escala de las variables originales. Si las características en $\boldsymbol{X}$ presentan magnitudes o unidades muy diferentes, las variables con mayor varianza dominarán las primeras componentes principales, sesgando el análisis. Para evitar esto, es común estandarizar los datos no solo restando la media, sino que también dividiendo cada variable por su desviación estándar muestral s_i. Esto se formaliza definiendo una matriz diagonal de desviaciones $\boldsymbol{D} = \text{diag}(s_1, \ldots, s_m)$, de este modo, obteniendo la matriz de datos estandarizada $\boldsymbol{Z}$:

$$\boldsymbol{Z} = \boldsymbol{D}^{-1}(\boldsymbol{X} - \overline{\boldsymbol{X}}). \tag{2.74}$$

Al realizar PCA sobre esta matriz $\boldsymbol{Z}$ es equivalente a calcular los vectores propios de la matriz de correlación de los datos originales, lo que garantiza que todas las variables contribuyan de manera equitativa al análisis, independientemente de sus unidades de medida.

2.10. Graphical Lasso

Para el cálculo de la distancia de Mahalanobis es necesario la estimación de la inversa de la matriz de covarianza, conocida como matriz de precisión $\Theta = \Sigma_X^{-1}$. Sin embargo, el

estimador empírico tradicional suele ser inestable o singular en escenarios de alta dimensionalidad (donde el número de características p es comparable o mayor al número de muestras N). Para abordar este problema, se utiliza el algoritmo Graphical Lasso mediante el cual se estima directamente la matriz de precisión, evitando la estimación de la matriz de covarianzas (Friedman et al., 2008).

Graphical Lasso estima una matriz de precisión dispersa $\hat{\Theta}$ mediante una penalización L_1 (lasso) sobre sus elementos. Esto no solo garantiza que la matriz sea invertible y positiva definida, sino que también induce dispersión (es decir, ceros en la matriz), lo que permite modelar la estructura de independencia condicional entre las características.

Este método estima la matriz de precisión resolviendo el siguiente problema de optimización convexa:

$$\hat{\Theta} = \arg\min_{\Theta \succ 0} \big(\mathrm{tr}(S\Theta) - \log\det(\Theta) + \lambda\|\Theta\|_1\big), \tag{2.75}$$

donde $\Theta \succ 0$ restrige Θ a matrices definida positivas, $S = \hat{\Sigma}_X^{-1}$ es la matriz de covarianza muestral y $\lambda > 0$ es el parámetro de regularización que controla el nivel de sparsidad de Θ. El término $\|\Theta\|_1$ corresponde a la norma L_1 aplicada a los elementos fuera de la diagonal, lo que induce una estructura dispersa (sparse) en la matriz de precisión.

En este trabajo, $\hat{\Theta}$ se utiliza para calcular la distancia de Mahalanobis dentro de la función de pérdida propuesta (Mahalanobis Distance Loss).

Finalmente, como se ha revisado hasta aquí, las funciones de pérdida actuales carecen de información textural local. A continuación, en el Capítulo 3, se propone una metodología que integra características radiómicas texturales, reducción de dimensionalidad y distancia de Mahalanobis para abordar esta limitación.

Capítulo 3

Propuesta

3.1. Introducción

En este capítulo se presenta la metodología propuesta para la segmentación de lesiones de esclerosis múltiple en imágenes de resonancia magnética cerebral. El objetivo principal es mejorar la detección y delineación de las lesiones de esclerosis múltiple, especialmente en las zonas cercanas a sus bordes, donde se concentran la mayoría de los errores de segmentación debido al solapamiento de intensidades y al efecto de volumen parcial.

Previo a la presentación de los aportes de este trabajo, se describe la etapa de preprocesamiento basada en transformaciones estándar de la literatura, orientada a estandarizar y acondicionar las imágenes de entrada a la red neuronal.

La propuesta se compone de dos elementos principales. En primer lugar, se introduce el Mapa de Distancia de Mahalanobis (MDM), una representación que combina información espacial y de textura, permitiendo capturar la estructura estadística subyacente para caracterizar con mayor precisión distintas regiones. El segundo elemento, que constituye al aporte principal de la propuesta corresponde a la introducción de la función de pérdida Mahalanobis Distance Loss (MDL), la cual incorpora el MDM para penalizar los errores de segmentación utilizando una representación más rica en información espacial y de textura. Con ello, se busca mejorar el desempeño del modelo en las zonas más desafiantes, particularmente en los bordes de las lesiones y su vecindad, donde se concentran la mayoría de las dificultades debido al solapamiento de intensidades y al efecto de volumen parcial.

En las secciones siguientes se detalla cada uno de estos componentes, junto con las decisiones de diseño adoptadas y su justificación.

3.2. Preprocesamiento

Como se indicó y detalló en la sección 2.6, el preprocesamiento de las imágenes de resonancia magnética cerebrales tiene como objetivo la estandarización de los volúmenes provenientes de distintos centros de imagenología, los cuales utilizan protocolos de adquisición heterogéneos. Además, mediante este proceso se busca mejorar la calidad de los datos de entrada al modelo de segmentación y, de este modo, facilitar el aprendizaje y rendimiento de la red convolucional.

El preprocesamiento se compone de dos etapas principales, las que a su vez se componen de varias subetapas que transforman la MRI de entrada. En la Figura 3.1 se ilustra el esquema completo del preprocesamiento aplicado. La primera etapa corresponde a una secuencia de transformaciones ampliamente utilizadas en el procesamiento de imágenes de resonancia magnética cerebral (Carass et al., 2017; Zhou et al., 2019). Estas transformaciones fueron implementadas mediante módulos del *Functional Magnetic Resonance Imaging of the Brain Software Library* (FSL) (Jenkinson et al., 2012). El proceso comenzó con la corrección del sesgo del campo magnético (bias field correction) mediante el uso del módulo FAST (Zhang et al., 2001). A continuación, se realizó la extracción del tejido encefálico (cerebro, cerebelo y tronco encefálico) mediante la técnica de skull stripping utilizando el módulo BET (Jenkinson, 2005), con el fin de eliminar estructuras tales como el cráneo y los globos oculares, que pudieran interferir en el análisis posterior. Una vez aislado el tejido cerebral, se aplicó una segunda corrección del sesgo para mejorar la homogeneidad del volumen obtenido. Finalmente, se llevó a cabo una registración rígida (lineal) utilizando el módulo FLIRT (Jenkinson et al., 2002) con el objetivo de alinear los volúmenes al espacio de referencia común MNI y corregir diferencias en orientación y geometría entre cerebros.

La segunda etapa del preprocesamiento se centró en la estandarización de intensidades. Se aplicó un truncamiento de intensidades restringiendo los valores de los vóxeles al rango definido por los cuantiles [0.01, 0.995]. El cuantíl superior 0.995 fue determinado de forma empírica con el fin de preservar la resolución en intensidades altas, dado que están asociadas a lesiones en secuencias/canal FLAIR. Posteriormente, las intensidades fueron normalizadas

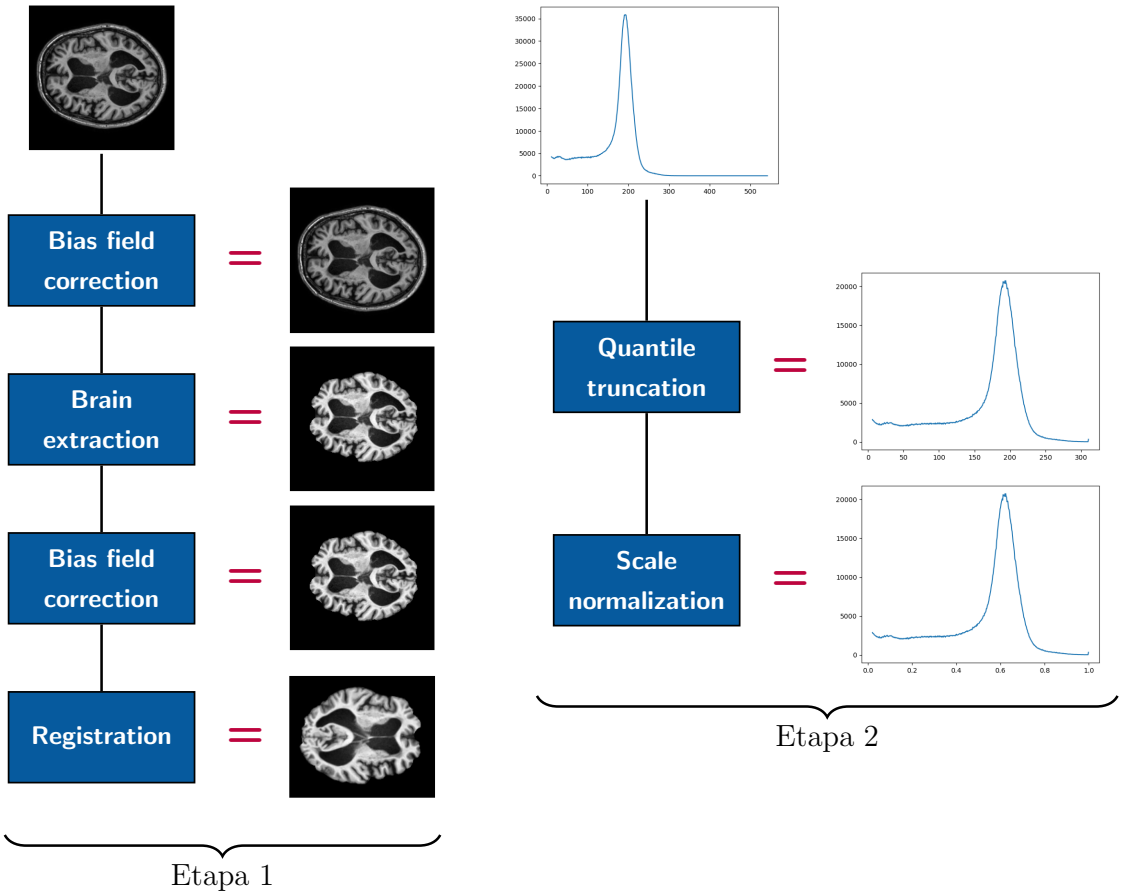


Figura 3.1: Etapas de preprocesamiento de MRI cerebrales.

al intervalo $[0, 1]$ mediante un reescalamiento lineal. Se seleccionó este método en lugar de alternativas como la estandarización o el histogram matching, ya que produjo mejores resultados en la configuración base del modelo (arquitectura U-Net con la función de pérdida Dice generalizada).

3.3. Mapa de Distancia de Mahalanobis

Como se observa en la literatura, los mapas de distancia DTM (Ecuación 2.28) y SDF (Ecuación 2.29) han sido ampliamente utilizados de manera exitosa como componentes de funciones de pérdida para la tarea de segmentación de imágenes. Si bien estos mapas aportan información espacial importante, esta se basa exclusivamente en la distancia euclidiana al

borde más cercano. En consecuencia, no incorporan ningún tipo de información de textura que permita mejorar la segmentación en regiones con apariencia similar. Esta limitación es especialmente crítica en la segmentación de lesiones de esclerosis múltiple, dado que en los bordes se suelen presentar problemas como solapamiento en la distribución de intensidades y el efecto de volumen parcial entre tejido lesionado y tejido sano. Como consecuencia, dos vóxeles pueden presentar la misma distancia euclidiana al borde pero pertenecer a regiones con textura muy diferentes, lo que restringe la capacidad discriminativa de DTM y SDF en zonas ambiguas.

Debido a lo anterior, surge la necesidad de un mapa de distancia que no solo capture la distancia euclidiana al borde de la lesión, sino que también las diferencias de textura presente en la distribución local de las intensidades o de las características derivadas de la imagen. Con este objetivo, se propone el mapa de distancia de Mahalanobis, una representación que incorpora información de textura, la cual permite detectar diferencias de textura entre el vecindario local de cada vóxel y el vecindario de los vóxeles borde de la clase opuesta.

Para construir el mapa de distancia de Mahalanobis, se utiliza la distancia de Mahalanobis previamente descrita en la sección 2.8.1 del Capítulo 2. Como fue mencionado, esta métrica permite cuantificar la divergencia entre el vector de características de un vóxel y el vector de características prototipo que representa la distribución local de una clase, incorporando la estructura de covarianza entre las características (Ver Ecuación 2.60).

A continuación se describe el procedimiento para la construcción del MDM.

El Algoritmo 5 resume el procedimiento general para producir el mapa de distancia de Mahalanobis. En la línea 2, se genera el mapa de características radiómicas y espaciales $F \in \mathbb{R}^{N\times H\times W}$, donde cada vóxel p de la imagen FLAIR $I \in \mathbb{R}^{H\times W}$ se representa mediante un vector de N características, es decir, $F(p) \in \mathbb{R}^N$. En la línea 3, se identifican los vóxeles pertenecientes a los bordes de la clase foreground (GB) y background (BB). Estos conjuntos son fundamentales para la construcción de las regiones de interés (RoIs) en la línea 4.

Cada RoI se define como la intersección entre una máscara esférica de radio $r = 5$ centrada en cada vóxel de borde y la máscara de la clase correspondiente del vóxel. Dado que las imágenes de resonancia magnética son tridimensionales, se incorpora el parámetro de adyacencia a mediante el cual se controla la incorporación de contexto volumétrico: si $a = 0$, solo se utiliza la lámina del vóxel de borde, en cambio, si $a > 0$, se incluyen las láminas adyacentes $[i - a, \ldots, i, \ldots, i + a]$. Un $a > 0$ permite aumentar el volumen del vecindario, de este modo,

Algoritmo 5 Mapa de Distancia de Mahalanobis (MDM)

1: **Input:** G : Ground truth, I : FLAIR, a : adyacencia
2: Calcular mapa de características radiómicas y espaciales F
3: Encontrar coordenadas de borde para formar conjuntos GB y BB
4: Definir RoIs con adyacencia a para cada coordenada en GB y BB
5: Para cada RoI: Calcular prototipo $\tilde{x}$ y estimar la matriz de precisión $\hat{\Sigma}_X^{-1}$
6: Obtener conjunto de coordenadas de interés $H \subset \Omega$:

$$H = \{p \in \Omega \mid G(p) = 1\} \cup \{p \in G_{\text{out}} \mid G_{\text{DTM}}(p) \leq D\} \tag{3.2}$$

7: Calcular distancia de Mahalanobis para las posiciones en H:
8: **for all** coordenada $p \in H$ **do**
9: **if** $p \in G_{\text{out}} \cap H$ **then**
10: Encontrar la coordenada borde más cercana en GB, y cargar $\tilde{x}$ y $\hat{\Sigma}_X^{-1}$ correspondientes
11: **else if** $p \in G_{\text{in}}$ **then**
12: Encontrar la coordenada borde más cercana en BB, y cargar $\tilde{x}$ y $\hat{\Sigma}_X^{-1}$ correspondientes
13: **end if**
14: Calcular:

$$G_{\text{MDM}}(p) = \text{MD}(F(p), \tilde{x}) \tag{3.1}$$

15: **end for**
16: **Output:** G_{MDM}

aumentando la cantidad de vectores de características del RoI utilizados en la construcción del vector prototipo y en la matriz de precisión obtenidos en la línea siguiente del algoritmo. En la Figura 3.2 se ilustra el proceso para $a = 0$.

En la línea 5, para cada RoI se obtiene el vector prototipo $\tilde{x}$ y se estima la matriz de precisión Σ_X^{-1}. El vector prototipo corresponde al vector mediana de la RoI, mientras que la matriz de precisión se estima directamente mediante el algoritmo Graphical Lasso (Friedman et al., 2008), de este modo evitando estimar la matriz de covarianzas. Este estimador resulta adecuado cuando el tamaño de la RoI es reducido o existe dependencia lineal entre

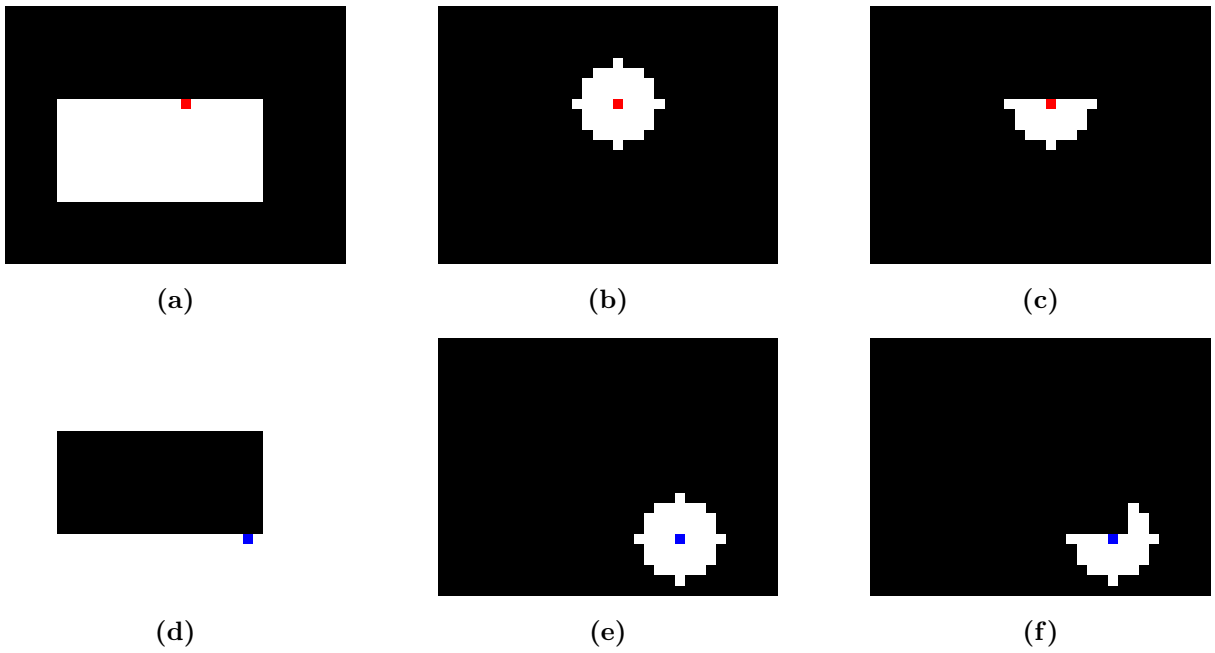


Figura 3.2: Extracción de RoI con parámetro $a = 0$. (a)-(c) Procesamiento de la clase lesión (*foreground*): (a) máscara binaria, (b) máscara circular en borde GB y (c) RoI resultante. (d)-(f) Procesamiento de la clase fondo (*background*): (d) máscara binaria, (e) máscara circular en borde BB y (f) RoI resultante.

las características, ya que permite obtener una estimación estable y dispersa de la matriz de precisión.

En la línea 6, se construye la máscara H, la cual incluye tanto los vóxeles pertenecientes a la lesión (ground truth G) como aquellos del fondo ubicados en el entorno superficial, definidos por su distancia al borde:

$$H = \{p \in \Omega \mid G(p) = 1\} \cup \{p \in G_{\text{out}} \mid G_{\text{DTM}}(p) \leq D\}. \tag{3.2}$$

El valor $D = 10$ mm fue seleccionado empíricamente desde el conjunto de opciones $D \in \{5, 10, 15\}$, considerando el rendimiento del método en las zonas críticas de la segmentación, donde se presentan la mayoría de falsos positivos y falsos negativos debido al solapamiento de intensidades y al efecto de volumen parcial.

Finalmente, entre las líneas 7 y 15, se calcula la distancia de Mahalanobis entre cada vóxel $p \in H$ y el prototipo correspondiente a la clase opuesta. Para los vóxeles pertenecientes a la lesión, se selecciona el vóxel de borde más cercano en BB de acuerdo a la distancia

euclidiana. De manera análoga, para los vóxeles background incluidos en H, se utiliza el vóxel de borde más cercano perteneciente a GB. A continuación, con el vector prototipo $\tilde{x}$ y con $\hat{\Sigma}_X^{-1}$ (estimación de la matriz de precisión) correspondientes, se calcula la distancia de Mahalanobis

$$MD(F(p), \tilde{x}) = \sqrt{(F(p) - \tilde{x})^T \, \hat{\Sigma}_X^{-1} \, (F(p) - \tilde{x})}. \tag{3.3}$$

El resultado final es el mapa G_{MDM}, una representación que captura diferencias de textura en las zonas más complejas, es decir, en el borde de las lesiones y su vecindad inmediata, tanto hacia el interior como hacia el exterior. Precisamente es en esta zona donde se concentran los principales desafíos de la segmentación automática, debido al solapamiento de intensidades y al efecto del volumen parcial.

3.4. Mahalanobis Distance Loss

La función de pérdida propuesta se construye utilizando el MDM, descrito previamente en el Algoritmo 5. Para obtener el MDM, es necesario generar primero el mapa de características F, el cual contiene N canales derivados tanto de la segmentación G como de la secuencia FLAIR. Desde el dominio Ω de G se extraen dos características espaciales para cada píxel, correspondientes a las coordenadas i y j del vector de coordenadas $p = (i, j)$, asociadas a las distancias euclidianas empleadas en los mapas tradicionales DTM (Ecuación 2.28) y SDF (Ecuación 2.29). Adicionalmente, desde la imagen FLAIR se incorporan características de intensidad y característica de textura radiómicas (Mayerhoefer et al., 2020), tales como el gradiente, así como también un conjunto de características derivadas de las matrices de co-ocurrencia de niveles de gris (GLCM) (Haralick et al., 1973) y de las matrices run-length (GLRLM) (Galloway, 1975).

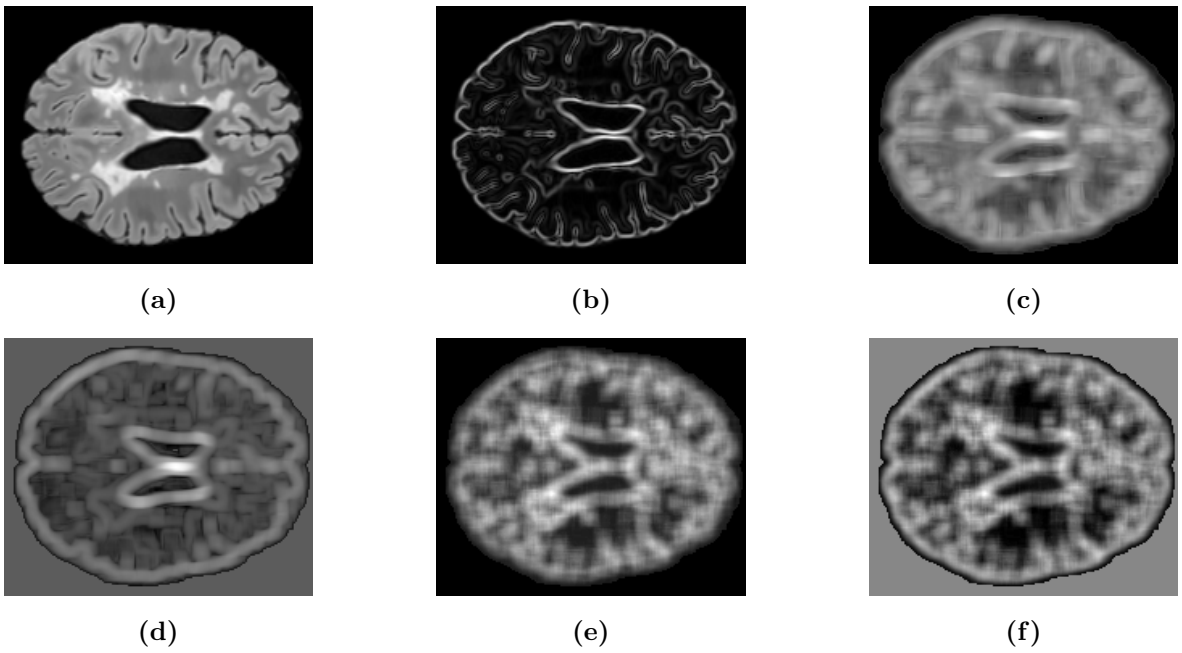
Las características de textura se calculan a partir de ventanas cuadradas centradas en cada vóxel. En el caso del gradiente, se utilizan vecindarios de tamaño 3×3, mientras que para las 13 características GLCM y las 11 características GLRLM se utilizan vecindarios de 9×9. Se seleccionó este mayor tamaño debido a que, de manera empírica, se observaron problemas en la estimación de dichas características cuando se utilizaban parches de menor tamaño, particularmente en zonas homogéneas o con bajo contraste, donde las matrices de co-ocurrencia y run-length no se definían adecuadamente. Esto se debe a que cuando el número

de niveles de gris presentes en una ventana es reducido o con muy baja variabilidad, las distribuciones de frecuencias asociadas a la GLCM y GLRLM tienden a concentrarse en un número muy pequeño de entradas, lo que produce matrices casi vacías o con filas/columnas nulas. Esto condujo a problemas como divisiones por cero, varianzas nulas o índices no definidos en las fórmulas como contraste, homogeneidad y entropía.

La alta dimensionalidad de la matriz de características X construida para cada RoI puede generar problemas en la estimación de la matriz de precisión, sobretodo cuando sus características están fuertemente correlacionadas. En estas situaciones, la matriz de covarianza tiende a ser casi singular, lo que dificulta o incluso imposibilita el calcular su inversa. Para mitigar este problema, se aplicó un análisis de componentes principales (PCA) sobre las características radiómicas GLCM y GLRLM, con el objetivo de reducir su dimensionalidad. Dicho análisis dejó en evidencia que las dos primeras componentes principales explican alrededor del 90 % de la varianza total, lo que indica que muchas de las características originales son redundantes o presentan un alto grado de dependencia lineal. Por esta razón, con el fin de mejorar la estabilidad numérica en $\hat{\Sigma}_X^{-1}$, se conservaron únicamente las dos primeras componentes principales de cada conjunto.

Análisis empíricos mostraron que incluir componentes adicionales no mejoraba el desempeño de la segmentación y, por el contrario, introducía inestabilidad en la estimación de la matriz de precisión. Esto, a su vez, incrementaba el costo computacional, ya que hacía necesario ajustar con mayor precisión los hiperparámetros del algoritmo Graphical Lasso, tales como el parámetro de regularización, las tolerancias de convergencia y el número máximo de iteraciones.

En la Figura 3.3 se muestran las características que componen el mapa de características F, omitiendo las correspondientes a las coordenadas espaciales.



(a) **(b)** **(c)**

(d) **(e)** **(f)**

Figura 3.3: Visualización de características de textura. (a) Canal FLAIR, (b) Magnitud del gradiente, (c, d) Primera y segunda componente principal de la GLCM y (e, f) Primera y segunda componente principal de la GLRLM.

La idea central de la función de pérdida propuesta, es utilizar el MDM para asignar penalizaciones diferenciadas a los falsos positivos (FP) y falsos negativos (FN), incorporando información de textura directamente en la estructura de la función de pérdida. Mediante este mecanismo se busca aproximar el razonamiento empleado por los médicos especialistas, quienes utilizan información local de textura para complementar el análisis basado en intensidades, especialmente en zonas donde las distribuciones presentan solapamiento debido al efecto del volumen parcial.

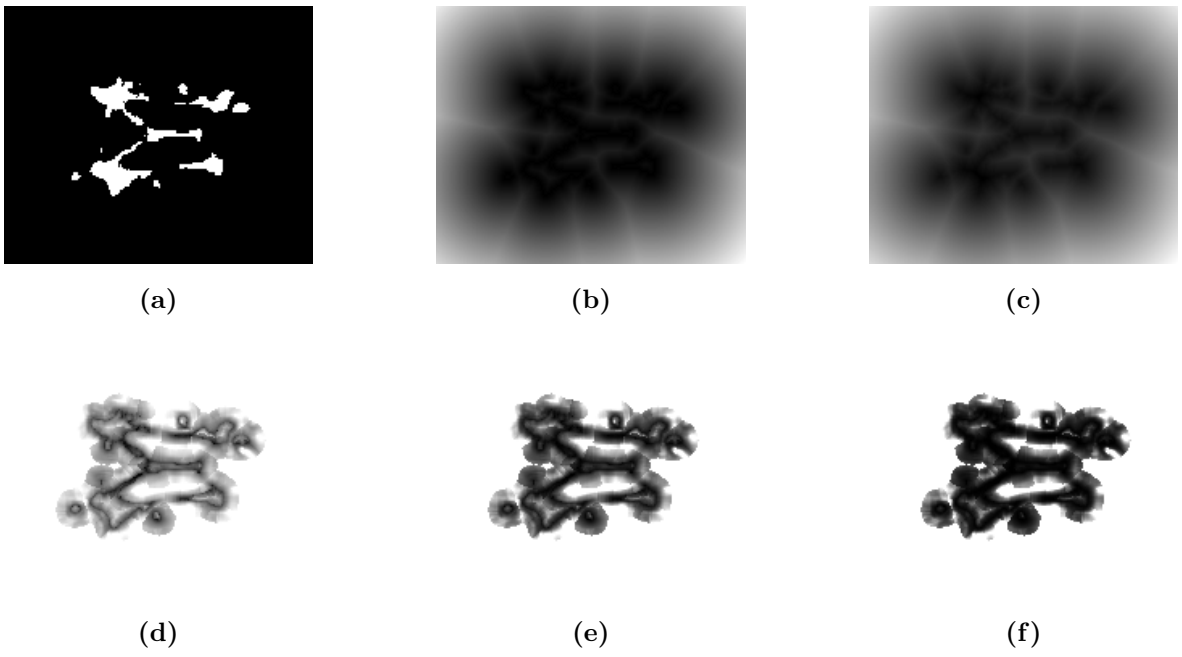
La función de pérdida propuesta, denominada Mahalanobis Distance Loss (MDL), se define como:

$$L_{\mathrm{MDL}}(G, s_\theta) = \epsilon \cdot L_{\mathrm{GDice}} + (1-\epsilon) \cdot \frac{1}{|\Omega|} \sum_{p \in \Omega} G_{\mathrm{MDM}}(p)^\lambda \left[(1 - G(p))\, s_\theta(p) + G(p)\, (1 - s_\theta(p)) \right], \tag{3.4}$$

donde $G_{\mathrm{MDM}}(p)$ es el valor del mapa MDM en el vóxel p, λ es el parámetro que controla la intensidad relativa de la penalización, y ϵ decrece de manera lineal desde 1.0 hasta 0.01 a lo largo de las épocas de entrenamiento. De este modo, al inicio del entrenamiento predomina

el término L_{GDice} que permite estabilizar la segmentación global, mientras que en épocas posteriores aumenta el peso relativo de la penalización basada en distancia textural a los bordes.

De manera más específica, el parámetro λ regula cómo se escala la contribución de cada vóxel según su distancia textural $G_{\text{MDM}}(p)$ al borde de la clase opuesta. Cuando $\lambda > 1$, los errores cometidos en vóxeles que presentan gran distancia textural reciben una penalización más alta. En cambio, si $0 < \lambda < 1$, la penalización se comprime para los vóxeles texturalmente distantes, es decir, se les asigna un peso menor, y en consecuencia, se enfatiza la contribución relativa de aquellos vóxeles más cercanos al borde. Esta cercanía no solo se expresa desde el punto de vista textural a través de $G_{\text{MDM}}(p)$, sino también espacial, dado que dicho mapa incorpora explícitamente como características a las coordenadas de cada vóxel en su construcción. Las Figuras 3.4d–f muestran ejemplos del mapa G^{λ}_{MDM} para distintos valores de λ. En contraste con los mapas DTM (Figura 3.4b) y SDF (Figura 3.4c), el MDM ofrece un rango más amplio de separación entre vóxeles cercanos al borde (en distancia euclidiana) gracias a la incorporación de información de textura.



(a) **(b)** **(c)**

(d) **(e)** **(f)**

Figura 3.4: Comparación de mapas de distancia: (a) Máscara de segmentación G, (b) G_{DTM}, (c) G_{SDF} y (d–f) mapas G^{λ}_{MDM} para diferentes valores de λ: (d) $\lambda = 0.5$, (e) $\lambda = 1.5$ y (f) $\lambda = 2.5$.

En la Figura 3.5(a) se presenta un esquema general del Algoritmo 5 utilizado para generar el mapa de distancia de Mahalanobis. La Figura 3.5(b) muestra el flujo completo de procesamiento empleado en la segmentación de lesiones de esclerosis múltiple, el cual está compuesto por la red convolucional empleada, la función de pérdida propuesta y el mapa de distancia de Mahalanobis.

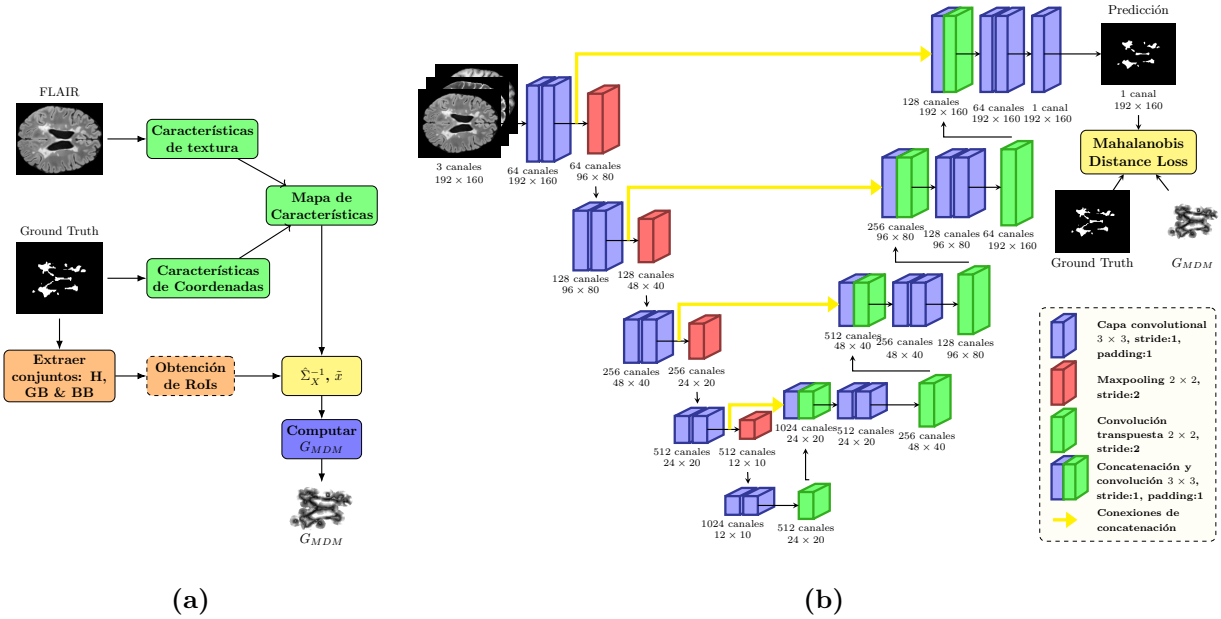


Figura 3.5: (a) Esquema general del Algoritmo 5 para el mapa de distancia de Mahalanobis y (b) flujo completo de procesamiento de segmentación de lesiones de esclerosis múltiple.

Capítulo 4

Resultados

4.1. Introducción

En este capítulo se presentan y analizan los resultados obtenidos al aplicar la función de pérdida propuesta, Mahalanobis Distance Loss, junto con un conjunto de funciones de pérdida ampliamente utilizadas en la literatura para la segmentación de imágenes médicas.

El análisis se organiza de la siguiente manera. En primer lugar, en la Sección 4.2 se presenta un resumen de las principales funciones de pérdida utilizadas en la tarea de segmentación, con el fin de facilitar la interpretación de los resultados posteriores. A continuación, se describen los conjuntos de datos utilizados y las métricas de evaluación empleadas para cuantificar el grado de concordancia entre las segmentaciones automáticas y las segmentaciones ground truth. Posteriormente, se detallan los aspectos más relevantes del proceso de entrenamiento, incluyendo la estrategia de selección del hiperparámetro λ en Mahalanobis Distance Loss. Finalmente, se reportan los resultados cuantitativos y cualitativos obtenidos en los conjuntos de datos ISBI-MS y MSSEG2016, junto con una discusión comparativa que permite analizar las fortalezas y limitaciones de las funciones de pérdida.

4.2. Resumen de funciones de pérdidas para la segmentación de imágenes

Con el fin de contextualizar los resultados presentados en este capítulo, en esta sección se resumen las principales funciones de pérdida utilizadas en la segmentación de imágenes médicas. Estas funciones fueron seleccionadas por su amplia adopción en la literatura, su capacidad para manejar distintos grados de desbalance de clases y por representar enfoques basados en probabilidad, solapamiento espacial y distancia a bordes. Es importante destacar que para la etapa de evaluación experimental se utilizó el subconjunto de funciones de pérdida más utilizadas en la literatura y en especial aquellas que en su construcción presenten términos basados en mapas de distancia a los bordes como DTM y SDF.

Si bien su formulación detallada fue expuesta previamente en la sección 2.5, se incluye aquí una síntesis orientada a facilitar la interpretación de los resultados experimentales.

A continuación, en la Tabla 4.1 se presenta una descripción condensada de cada función de pérdida considerada en los experimentos.

Tabla 4.1: Resumen de principales funciones de pérdida utilizadas en segmentación de imágenes

Función de pérdida	**Descripción**
Binary Cross-Entropy Loss (Yi-de et al., 2004)	Basada en probabilidad; mide la distancia entre dos distribuciones. Realiza clasificación vóxel a vóxel. Sensible al desbalance de clases.
Weighted BCE Loss (Xie and Tu, 2015)	Extiende BCE mediante pesos diferenciados para FP y FN. Adecuada para moderado desbalance de clases.
Focal Loss (Lin et al., 2020)	Focaliza ejemplos difíciles ajustando su contribución al gradiente. Adecuada para alto desbalance de clases. Requiere ajustar un parámetro.
Dice Loss (Milletari et al., 2016)	Métrica basada en el solapamiento entre regiones. No depende de TN. Adecuada para moderado desbalance de clases.
Generalized Dice Loss (Sudre et al., 2017)	Extiende Dice mediante pesos inversamente proporcionales a la frecuencia de cada clase. Adecuada para alto desbalance de clases.
Tversky Loss (Hashemi et al., 2018)	Controla explícitamente la asimetría entre FP y FN mediante parámetros α y β.
Asymmetric Similarity (F_β) Loss (Hashemi et al., 2019)	Ajusta la penalización de FN mediante un parámetro β. Adecuada para alto desbalance de clases, especialmente cuando los FN son más relevantes.
Focal Tversky Loss (Abraham and Khan, 2019)	Combina Tversky Loss con focalización para reforzar la atención en clases difíciles. Adecuada para alto desbalance de clases. Requiere ajustar tres parámetros.
Sensitivity–Specificity Loss (Brosch et al., 2015)	Combina sensibilidad y especificidad mediante ponderaciones por clase. Adecuada para moderado desbalance de clases.
Log-Cosh Dice Loss (Jadon, 2020)	Variante suave y más estable de Dice Loss, menos sensible a outliers.

Exponential Logarithmic Loss (Wong et al., 2018)	Combina Dice Loss y BCE Loss mediante transformaciones logarítmicas y exponenciales. Útil en estructuras pequeñas.
Hausdorff Distance Loss (Karimi and Salcudean, 2020)	Inspirada en la distancia de Hausdorff; incorpora distancias a bordes mediante DTM. Adecuada para alto desbalance de clases.
Boundary Loss (Kervadec et al., 2021)	Utiliza la distancia a bordes mediante SDF para penalizar discrepancias en las fronteras. Adecuada para alto desbalance de clases.
Active Boundary Loss (Wang et al., 2021)	Optimiza vectores de dirección en vóxeles de borde utilizando DTM. Adecuada para moderado desbalance de bordes. Puede presentar entrenamiento inestable en lesiones pequeñas.
Boundary-Sensitive Loss (Du et al., 2023)	Penaliza FP y FN tanto en interior como en bordes de la lesión. Adecuada para moderado desbalance de clases. Puede ser inestable en lesiones pequeñas.
Conditional Boundary Loss (Wu et al., 2023)	Modela relaciones entre pares de vóxeles de la misma y diferentes clases. Puede ser inestable en lesiones pequeñas.

4.3. Conjuntos de datos

En este trabajo fueron utilizados conjuntos de datos de acceso público ampliamente reconocidos en la evaluación de métodos de segmentación de lesiones cerebrales: ISBI-MS y MSSEG2016. Estos conjuntos de datos provienen de desafíos (challenges) internacionales en los que distintas instituciones contribuyeron con imágenes multimodales de resonancia magnética, obtenidas con protocolos de adquisición heterogéneos.

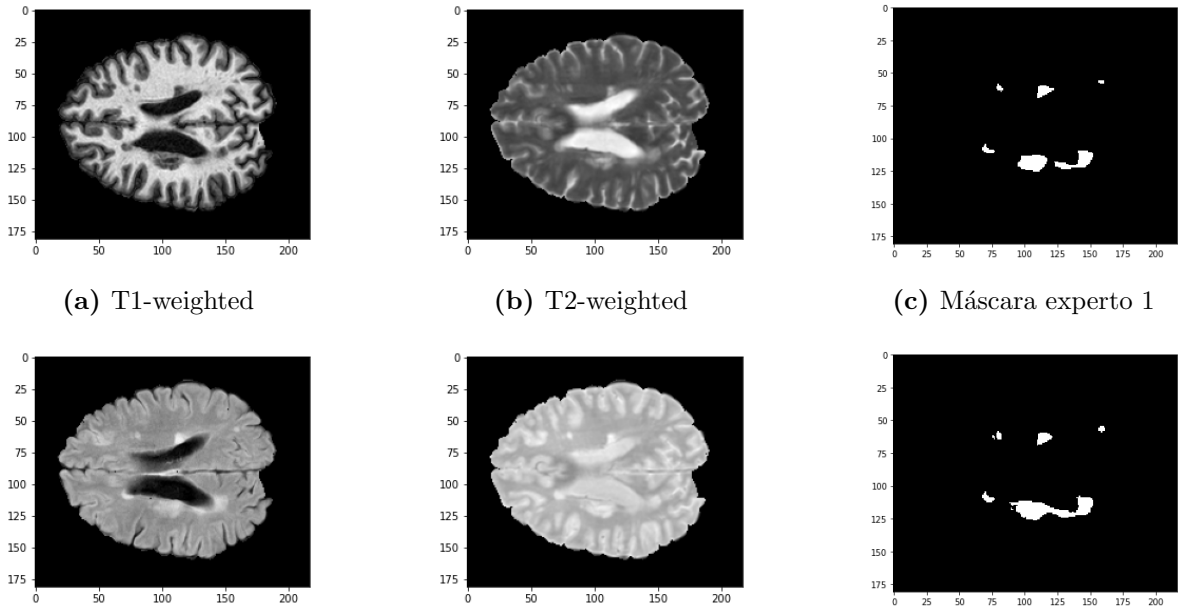
A continuación se realiza una descripción de cada conjunto de datos.

4.3.1. ISBI-MS

ISBI-MS (donde MS refiere a Multiple Sclerosis) es un conjunto de datos de libre acceso preparado para el desafío *Longitudinal MS Lesion Segmentation Challenge*, organizado en el marco del International Symposium on Biomedical Imaging (ISBI) 2015 (Carass et al., 2017). El conjunto de datos está compuesto por MRI de 19 pacientes con esclerosis múltiple divididos en dos conjuntos, 5 pacientes en el conjunto de entrenamiento y 14 en el conjunto de test. Para cada paciente se cuenta entre 4-6 volúmenes multimodales de imágenes de resonancia magnética nuclear, adquiridas en lapsos de tiempo de aproximadamente 1 año entre cada adquisición, utilizando el mismo scanner Philips 3T Achieva (Philips Medical Systems, Best Netherlands).

El conjunto de datos fue liberado tanto en su estado original como preprocesado. Es

importante destacar que la segmentación realizada por expertos se llevó a cabo sobre el conjunto de datos preprocesados. El preprocesamiento consistió en los siguientes pasos: Bias field correction (corrección del sesgo del campo magnético en diferentes tejidos y zonas), skull stripping (extracción del cerebro, cerebelo y tronco encefálico), segundo Bias field correction, y registración no rígida al atlas estandar MNI152 de $1\,\mathrm{mm}^3$, la cual es una plantilla anatómica derivada de 152 adultos jóvenes mediante registración no lineal, descrita en (Fonov et al., 2011). En la figura 4.1 se muestra un ejemplo de las 4 secuencias y las dos máscaras de segmentación del paciente 1 en el tiempo 1.



(a) T1-weighted **(b)** T2-weighted **(c)** Máscara experto 1

(d) FLAIR **(e)** PD-weighted **(f)** Máscara experto 2

Figura 4.1: Ejemplo del conjunto de datos ISBI-MS (Carass et al., 2017). Se observan las distintas secuencias de MRI y las segmentaciones manuales de expertos.

Las secuencias corresponden a T1-weighted, T2-weighted, PD-weighted y FLAIR, todas con dimensiones de $181 \times 217 \times 181$ vóxeles y con resolución espacial isotrópica de $1\,\mathrm{mm}^3$. En el conjunto de entrenamiento, además de las cuatro secuencias, se incluyen dos máscaras binarias que corresponden a las segmentaciones realizadas por dos radiólogos expertos.

4.3.2. MSSEG2016

El conjunto de datos MSSEG2016 forma parte del desafío internacional MICCAI 2016 *MS Lesion Segmentation Challenge* (Commowick et al., 2018). Este dataset esta compuesto por imágenes de resonancia magnética multimodales correspondientes a 53 pacientes diagnosticados con esclerosis múltiple. Las imágenes fueron adquiridas en cuatro centros clínicos, utilizando distintos resonadores y protocolos de adquisición.

Cada adquisición incluye 5 secuencias: T1-weighted, T2-weighted, PD-weighted, Gadolinium-enhanced T1-weighted y FLAIR. Para el conjunto de entrenamiento se dispone de las máscaras de segmentación realizadas por 7 expertos, junto con una máscara de consenso, que corresponde a la fusión de estas mediante el algoritmo STAPLE (Warfield et al., 2004).

El conjunto se encuentra disponible tanto en su versión sin preprocesamiento como en su versión preprocesada. El preprocesamiento proporcionado por los organizadores del desafío incluye los siguientes paso en orden: skull stripping, bias field correction y registración rígida de todas las secuencias a la secuencia FLAIR.

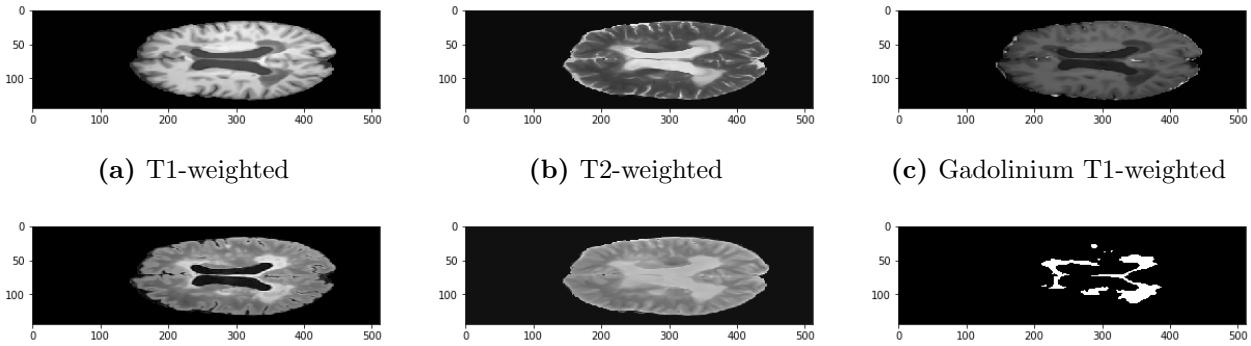
La Tabla 4.2 resume las principales características de adquisición de cada uno de los cuatro centros. Por su parte, en la figura 4.2 se ilustran las distintas secuencias y la máscara de segmentación de consenso correspondientes a un caso representativo del conjunto de datos.

Tabla 4.2: Detalles de adquisición de conjunto de datos MSSEG2016.

Centro	Modelo de scanner	Casos de entrenamiento	Casos de test	Matriz	Resolución [mm]
1	Siemens Verio 3T	5	10	$512 \times 512 \times 224$	$0.47 \times 0.47 \times 0.90$
3	General Electrics Discovery 3T	0	8	$336 \times 336 \times 261$	$0.74 \times 0.74 \times 0.70$
7	Siemens Aera 1.5T	5	10	$256 \times 224 \times 128$	$1.03 \times 1.03 \times 1.25$
8	Philips Ingenia 3T	5	10	$512 \times 512 \times 144$	$0.50 \times 0.50 \times 1.10$

4.4. Métricas de evaluación

En la evaluación de métodos de segmentación en imágenes médicas es habitual emplear un conjunto estándar de métricas cuantitativas que permiten medir el grado de similitud entre la segmentación de referencia o ground truth y la segmentación automática generada por



(a) T1-weighted **(b)** T2-weighted **(c)** Gadolinium T1-weighted

(d) FLAIR **(e)** PD-weighted **(f)** Máscara de consenso

Figura 4.2: Ejemplo del conjunto de datos MSSEG2016 (Commowick et al., 2018). Se observan las distintas secuencias de MRI y la segmentación de consenso, todas en resolución original.

una red neuronal convolucional. Estas métricas de evaluación se agrupan principalmente en dos categorías, basadas en solapamiento espacial y basadas en distancia espacial (Nai et al., 2021; Terven et al., 2025; Yeghiazaryan and Voiculescu, 2018).

Utilizando la misma notación que en las secciones 2.1 y 2.5, sea $I : \Omega \to \mathbb{R}^D$ la imagen con dominio espacial Ω, $G : I \to \{0, 1\}$ la segmentación binaria realizada por el experto o ground truth, y $P = 1_{[s_\theta > 0.5]}$ es la predicción, donde $s_\theta : I \to [0, 1]$ la salida softmax de la CNN.

La mayoría de las métricas presentadas a continuación pueden ser calculadas utilizando los resultados de la clasificación binaria presentes en la matriz de confusión o también conocida como matriz de contingencia, correspondiente a la clasificación vóxel a vóxel (Tabla 2.2). Para facilitar la lectura, esta matriz se muestra nuevamente a continuación:

Matriz de confusión. (Tabla 2.2)

		Predictor	
		0 (-)	1 (+)
Experto	0 (-)	TN	FP
	1 (+)	FN	TP

4.4.1. Métricas basadas en solapamiento espacial

Sensibilidad

La Sensibilidad, también denominada Recall o Tasa de Verdaderos Positivos (TPR), cuantifica la proporción de ejemplos positivos que son correctamente identificados por el modelo. En otras palabras, mide la capacidad del modelo para detectar todas las instancias de la clase de interés y evitar errores del tipo II, es decir, falsos negativos. Su definición es:

$$\begin{aligned} \mathrm{TPR}(G,P) &= \frac{\sum_{p\in\Omega} G(p)\cdot P(p)}{\sum_{p\in\Omega} G(p)\cdot P(p) + \sum_{p\in\Omega} G(p)\cdot(1-P(p))} \\ &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}. \end{aligned} \tag{4.1}$$

Especificidad

La Especificidad, también llamada Tasa de Verdaderos Negativos (TNR), cuantifica la proporción de ejemplos negativos que el modelo identifica correctamente. Es decir, evalúa la capacidad del modelo para evitar falsos positivos (FP o error tipo I) al clasificar instancias que pertenecen a la clase negativa. Su definición es la siguiente:

$$\begin{aligned} \mathrm{TNR}(G,P) &= \frac{\sum_{p\in\Omega} (1-G(p))\cdot(1-P(p))}{\sum_{p\in\Omega} (1-G(p))\cdot(1-P(p)) + \sum_{p\in\Omega} P(p)\cdot(1-G(p))} \\ &= \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}. \end{aligned} \tag{4.2}$$

Precisión

La Precisión, o Valor Predictivo Positivo (Positive Predictive Value, PPV), mide la proporción de las predicciones positivas del modelo que son correctas respecto a todos los ejemplos clasificados como positivos. Por tanto, cuantifica la fiabilidad de las clasificaciones positivas del modelo. Se define como:

$$\begin{aligned} \mathrm{PPV}(G,P) &= \frac{\sum_{p\in\Omega} G(p)\cdot P(p)}{\sum_{p\in\Omega} G(p)\cdot P(p) + \sum_{p\in\Omega} P(p)\cdot(1-G(p))} \\ &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}. \end{aligned} \tag{4.3}$$

Coeficiente Dice

El coeficiente Dice, también denominado coeficiente de similitud Dice, F_1-score o coeficiente de Sørensen-Dice, cuantifica el grado de solapamiento o intersección entre dos conjuntos: el ground truth G y la segmentación predicha P (Dice, 1945). Este coeficiente deriva como un caso especial del índice de Tversky (con $\alpha = \beta = 0.5$). Su formulación, expresada en términos de los componentes de la matriz de confusión para la evaluación binaria, es la siguiente:

$$\begin{aligned} \mathrm{Dice}(G,P) &= \frac{2\cdot\sum_{p\in\Omega} G(p)\cdot P(p)}{2\cdot\sum_{p\in\Omega} G(p)\cdot P(p) + \sum_{p\in\Omega} P(p)\cdot(1-G(p)) + \sum_{p\in\Omega} G(p)\cdot(1-P(p))} \\ &= \frac{2\cdot\mathrm{TP}}{2\cdot\mathrm{TP}+\mathrm{FN}+\mathrm{FP}}. \end{aligned} \tag{4.4}$$

Un valor $\mathrm{Dice}(G,P) = 0$ indica una nula intersección espacial entre los conjuntos, mientras que $\mathrm{Dice}(G,P) = 1$ indica una coincidencia exacta entre la segmentación automática y el ground truth.

4.4.2. Métricas basadas en distancia espacial entre superficies

Distancia de Hausdorff

La distancia de Hausdorff (HD) cuantifica el mayor error de segmentación considerando los bordes o fronteras entre dos objetos presentes en imágenes binarias (Nai et al., 2021). Es decir, esta métrica evalúa el peor caso de discrepancia entre las superficies predicha y la ground truth.

La distancia de Hausdorff unidireccional entre dos conjuntos de puntos u objetos X e Y, se define como la máxima distancia desde un punto en el borde de X hasta el punto más cercano en el borde de Y:

$$\mathrm{hd}(X, Y) = \max_{x \in X} \min_{y \in Y} \|x - y\|_2. \tag{4.5}$$

Dado que esta distancia no es simétrica, se define la distancia de Hausdorff bidireccional (o simplemente distancia de Hausdorff) como el máximo de las siguientes dos distancias de Hausdorff unidireccionales, donde $G = X$ e $P = Y$

$$\mathrm{hd}(X, Y) = \max_{x \in X}; \min_{y \in Y} \|x - y\|_2. \tag{4.6}$$

La HD es suceptible de ser afectada de manera adversa por la presencia de valores atípicos (outliers), debido a que utiliza la función máximo, la cual es un estadístico no robusto. Es por esta razón que en la literatura es común el uso de variantes robustas basadas en los percentiles 90 (HD90) y 95 (HD95) de las distancias de Hausdorff unidireccionales.

Distancia simétrica promedio de superficie

La métrica Distancia Simétrica Promedio de Superficie (Average Symmetric Surface Distance, ASSD) mide la discrepancia media entre los contornos de ambos objetos. Se calcula promediando la suma de las distancias mínimas desde los bordes del ground truth hacia el borde predicho y viceversa (Yeghiazaryan and Voiculescu, 2018). Se define como:

$$\mathrm{ASSD}(G, P) = \frac{\sum_{x \in \mathrm{GB}} d(x, \mathrm{PB}) + \sum_{y \in \mathrm{PB}} d(y, \mathrm{GB})}{|\mathrm{GB}| + |\mathrm{PB}|}, \tag{4.7}$$

donde GB y PB representan los bordes de G y P respectivamente. Las distancias mínimas se definen como $d(x, \mathrm{PB}) = \min_{y \in \mathrm{PB}} \|x - y\|_2$ y $d(y, \mathrm{GB}) = \min_{x \in \mathrm{GB}} \|x - y\|_2$.

4.4.3. Otras métricas utilizadas en segmentación de imágenes

Diferencia relativa de volumen

La métrica Diferencia Relativa de Volumen (Relative Volume Difference, RVD) cuantifica la disparidad en volumen entre la segmentación predicha binarizada P y el ground truth G,

normalizada por el tamaño del objeto real (Yeghiazaryan and Voiculescu, 2018). Se define formalmente como el valor absoluto de la diferencia relativa:

$$\mathrm{RVD}(G, P) = \frac{\big||P| - |G|\big|}{|G|}, \tag{4.8}$$

donde $|\cdot|$ representa el volumen (o cardinalidad) del conjunto de vóxeles.

Area bajo la curva Precision-Recall

El Área bajo la Curva Precision-Recall (AUCPR) es una métrica utilizada para evaluar clasificadores binarios en presencia de alto desbalance de clases.

A diferencia de el área bajo la curva ROC (AUCROC), AUCPR no considera los TN, lo cual evita una sobreestimación del rendimiento del modelo cuando existe un alto desbalance de clases donde la clase negativa es la mayoritaria (Boyd et al., 2013). Por tanto, la AUCPR puede ser más adecuada utilizar en tareas de segmentación donde el objetivo es detectar objetos pequeños.

Para obtener la AUCPR, primero se construye la curva Precision-Recall utilizando los pares de puntos (Precision,Recall), generados utilizando diferentes umbrales sobre la salida softmax del modelo. En este trabajo, la AUCPR fue obtenida utilizando las funciones auc[1] y precision_recall_curve[2] de la biblioteca scikit-learn (Pedregosa et al., 2011).

4.5. Entrenamiento

Para evaluar la función de pérdida propuesta, se utilizó una arquitectura convolucional basada en U-Net (Ronneberger et al., 2015), la cual constituye actualmente uno de los marcos más utilizados en la segmentación de imágenes médicas (Azad et al., 2024). En particular, se empleó un autoencoder convolucional con un camino de decodificación que utiliza convoluciones transpuestas en lugar de interpolación (ver Figura 3.5(b)), ya que esta elección permitió mejorar la reconstrucción de los mapas de características durante la etapa de upsampling.

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.auc.html`

[2] `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_curve.html#sklearn.metrics.precision_recall_curve`

El desarrollo e implementación de la red neuronal se realizó utilizando la biblioteca PyTorch (Ansel et al., 2024). Los experimentos se ejecutaron en un entorno distribuido compuesto por cuatro tarjetas NVIDIA GeForce GTX 1080Ti (11 GB VRAM cada una), instaladas en cuatro servidores distintos. Esta infraestructura requirió la configuración de un sistema de entrenamiento distribuido (Cruz et al., 2023), lo que permitió llevar a cabo un número suficiente de ejecuciones experimentales (20 corridas por cada función de pérdida evaluada), requisito fundamental para analizar la variabilidad del desempeño.

El entrenamiento se llevó a cabo siguiendo las estrategias reportadas en la literatura para funciones de pérdida basadas en borde (Kervadec et al., 2021; Wang et al., 2021). Los modelos se entrenaron por un máximo de 200 épocas, aplicando early stopping con una paciencia de 30 épocas si la métrica Dice no presentaba mejoras en el conjunto de validación. Este procedimiento fue crucial para prevenir el sobreajuste y la degradación de los resultados en las últimas épocas, donde el término basado en borde alcanza un mayor dominio sobre la pérdida total. Aunque se intentó limitar este dominio del término basado en bordes en las etapas finales del entrenamiento (por ejemplo, limitando el valor mínimo de ε, tal como se recomienda en (Kervadec et al., 2021; Wang et al., 2021)), en todos los casos el uso de early stopping produjo un mejor resultado en las métricas basadas en solapamiento espacial y métricas basadas en distancia entre superficies (bordes).

El proceso de optimización se realizó mediante el algoritmo Adam, con tasa de aprendizaje igual a 10^{-4} y un tamaño de batch de 16, cuyos parámetros fueron determinados de manera empírica en pruebas preliminares. Para cada conjunto de datos, únicamente se consideraron para entrenamiento aquellas slices que contenían lesiones, de este modo evitando introducir un exceso de muestras negativas que pudieran sesgar el aprendizaje y aumentar el costo computacional.

Los datos, es decir, las slices seleccionadas para todos los volumenes se realizó asignando un 60 % para entrenamiento, un 20 % para validación y un 20 % para test. Dado que las imágenes corresponden a volúmenes 3D, y para evitar la correlación entre slices adyacentes o cercanas de un mismo paciente, la partición se realizó a nivel de paciente y no a nivel de slice. Es decir, cada conjunto de datos contiene slices provenientes de pacientes diferentes (equivalente a un fold de group cross-validation). Esto asegura una evaluación más realista, donde las slices pertenecen a volúmenes de los conjuntos de validación y test de pacientes no vistos durante el entrenamiento.

4.5.1. Selección del parámetro λ

El ajuste del parámetro λ de la función de pérdida Mahalanobis Distance Loss se realizó de manera empírica evaluando los valores del conjunto $\{0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$. El objetivo fue identificar el valor que optimizara simultáneamente las métricas basadas en distancia espacial y solapamiento espacial, comparando la segmentación predicha P con la segmentación ground truth G. En esta selección se buscó no solo mejorar los resultados de métricas como Dice y ASSD, sino que también mitigar posibles desequilibrios entre falsos negativos y falsos positivos, especialmente relevantes en las métricas de sensibilidad (recall) y precisión (PPV), que reflejan de manera directa este comportamiento.

Los resultados obtenidos, resumidos en la Tabla 4.3, muestran que el valor $\lambda = 1.5$ alcanzó el mejor rendimiento general tanto en el conjunto de datos ISBI-MS como en MSSEG2016. Para este último conjunto, además, se observó que valores superiores a $\lambda = 2.5$ producían inestabilidades numéricas durante el entrenamiento, lo que acotó de manera natural el rango de búsqueda. De este modo, se seleccionó $\lambda = 1.5$ como el valor óptimo de Mahalanobis Distance Loss para ambos conjuntos de datos.

Tabla 4.3: Análisis de sensibilidad del hiperparámetro λ

	ISBI-MS						
λ	**HD95(sd)↓**	**ASSD(sd)↓**	**Sensibilidad(sd)↑**	**Precisión(sd)↑**	**Dice(sd)↑**	**rvd(sd)↓**	**auc-pr(sd)↑**
0.5	31.9470(9.6748)	3.5236(1.2628)	0.7127(0.0398)	0.7281(0.0730)	0.7132(0.0346)	0.1541(0.0930)	0.7205(0.0317)
1.0	31.3346(11.8639)	3.5310(1.6014)	0.7101(0.0308)	0.7446(0.0709)	0.7199(0.0291)	0.1647(0.0479)	0.7274(0.0285)
1.5	27.0210(11.2736)	3.1685(1.6361)	0.7051(0.0361)	0.7591(0.0684)	0.7247(0.0261)	0.1549(0.0558)	0.7322(0.0274)
2.0	25.3589(12.0826)	2.9046(1.3958)	0.6879(0.0423)	0.7834(0.0748)	0.7257(0.0309)	0.1697(0.0681)	0.7357(0.0326)
2.5	24.8438(8.6735)	2.8080(1.1443)	0.6384(0.0983)	0.8365(0.0573)	0.7138(0.0690)	0.2387(0.1364)	0.7376(0.0420)
3.0	29.0668(11.1076)	3.4869(1.8727)	0.5957(0.1008)	0.8668(0.0592)	0.6957(0.0756)	0.3092(0.1368)	0.7314(0.0465)
	MSSEG2016						
λ	**HD95(sd)↓**	**ASSD(sd)↓**	**Sensibilidad(sd)↑**	**Precisión(sd)↑**	**Dice(sd)↑**	**rvd(sd)↓**	**auc-pr(sd)↑**
0.5	17.4068(4.0942)	2.5211(0.4648)	0.6742(0.0532)	0.6946(0.0508)	0.6652(0.0251)	0.3002(0.0851)	0.6851(0.0200)
1.0	19.3930(4.5113)	2.5342(0.4190)	0.6866(0.0380)	0.6967(0.0556)	0.6743(0.0197)	0.3071(0.0762)	0.6923(0.0159)
1.5	17.8207(5.5012)	2.4039(0.4239)	0.6868(0.0412)	0.7039(0.0508)	0.6794(0.0184)	0.2802(0.0656)	0.6960(0.0153)
2.0	17.4090(5.0380)	2.5485(0.5108)	0.6569(0.0298)	0.7071(0.0437)	0.6643(0.0173)	0.2780(0.0628)	0.6827(0.0151)
2.5	18.5212(4.7203)	2.9840(0.7522)	0.6278(0.0486)	0.7248(0.0468)	0.6506(0.0269)	0.3301(0.0646)	0.6771(0.0198)

Con el hiperparámetro λ seleccionado para ambos conjuntos de datos, se procede en la siguiente sección con la evaluación comparativa del método propuesto frente al estado del arte.

4.5.2. Resultados

Las Tablas 4.4 y 4.5 presentan los resultados obtenidos para las distintas funciones de pérdida evaluadas en los conjuntos de datos ISBI-MS y MSSEG2016. En el caso de Conditional Boundary Loss, se utilizó el valor recomendado por los autores en (Wu et al., 2023), fijando $\beta = 0.5$. Para Boundary-Sensitive Loss, el ajuste del parámetro correspondió a $\alpha = 0.7$ para ISBI-MS y $\alpha = 0.6$ para MSSEG2016.

En ambos conjuntos de datos, la función de pérdida propuesta, Mahalanobis Distance Loss (Ecuación (3.4)), obtuvo de manera sistemática los mejores resultados en las métricas de distancia de borde (HD95 y ASSD), así como en métricas de solapamiento espacial como Precisión, Dice y AUC-PR. En relación con la métrica diferencia relativa de volumen (RVD), la Boundary Loss alcanzó el mejor desempeño en ISBI-MS, mientras que la función propuesta obtuvo el mejor resultado en MSSEG2016. La única métrica en la que la función de pérdida propuesta nunca obtuvo el mejor resultado fue Sensibilidad (recall), en ISBI-MS obtuvo el segundo mejor resultado superada por la Generalized Dice Loss, y en MSSEG2016 obtuvo el tercer lugar por detrás de Conditional Boundary Loss y de Hausdorff Distance Loss, donde esta última obtuvo el mejor resultado.

En general, los resultados indican que las funciones de pérdida que hacen uso de mapas de distancia, como Hausdorff Distance Loss, Boundary Loss y Mahalanobis Distance Loss, tienden a obtener un mejor rendimiento de manera consistente en un amplio rango de tipos de métricas. Por el contrario, aquellas funciones orientadas a penalizar exclusivamente los vóxeles de borde mal clasificados, tales como Active Boundary Loss y Conditional Boundary Loss, muestran un desempeño inferior en este contexto. Esto se puede explicar por la inherente complejidad de la segmentación de lesiones de esclerosis múltiple, donde fenómenos como el efecto de volumen parcial, la superposición de intensidades entre tejidos sanos y lesionados, y la variabilidad morfológica dificultan la definición de los bordes de las lesiones. Es importante señalar que estas dificultades son menos importantes en los conjuntos de datos en los cuales dichas funciones de pérdida basadas en bordes fueron originalmente diseñadas y utilizadas.

Las Figuras 4.3(d) y 4.4(d) ilustran los resultados obtenidos mediante la función de pérdida propuesta. Como referencia, las Figuras 4.3(a) y 4.4(a) muestran las segmentaciones producidas con Generalized Dice Loss, mientras que las Figuras 4.3(b)-(c) y 4.4(b)-(c) presentan los resultados de las dos funciones de pérdida con mejor desempeño según las métricas Dice

Tabla 4.4: Comparación cuantitativa de los resultados de segmentación en el conjunto de datos ISBI-MS

Función de pérdida	HD95(sd)↓	ASSD(sd)↓	Sensibilidad(sd)↑	Precisión(sd)↑	Dice(sd)↑	RVD(sd)↓	AUC-PR(sd)↑
L_{GD}	31.22(11.52)	4.04(2.33)	**0.721**(0.038)	0.708(0.087)	0.706(0.050)	0.201(0.181)	0.715(0.041)
L_{BL}	30.29(09.75)	3.45(1.42)	0.699(0.032)	0.745(0.064)	0.716(0.032)	**0.144**(0.049)	0.722(0.034)
L_{HD}	29.88(11.49)	3.61(2.05)	0.700(0.034)	0.748(0.082)	0.715(0.037)	0.172(0.069)	0.724(0.036)
L_{ABL}	31.94(10.67)	3.76(1.48)	0.702(0.045)	0.742(0.057)	0.715(0.032)	0.155(0.056)	0.722(0.031)
L_{CBL}	28.30(09.93)	3.47(1.57)	0.701(0.039)	0.740(0.079)	0.711(0.037)	0.180(0.101)	0.721(0.033)
L_{BS}	35.94(11.41)	4.13(1.93)	0.704(0.031)	0.735(0.078)	0.711(0.039)	0.176(0.099)	0.719(0.035)
$L_{MDL}^{\lambda=1.5}$ **(Propuesta)**	**27.02**(11.27)	**3.17**(1.64)	0.705(0.036)	**0.759**(0.068)	**0.725**(0.026)	0.155(0.056)	**0.732**(0.027)

Nota. Los valores representan la media seguida de la desviación estándar (sd), calculados sobre 20 corridas experimentales. El mejor resultado de cada métrica se destaca en negrita.

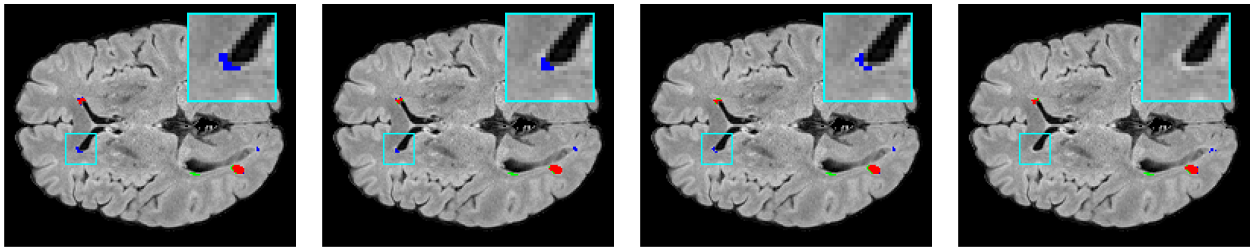
Tabla 4.5: Comparación cuantitativa de los resultados de segmentación en el conjunto de datos MSSEG2016

Función de pérdida	HD95(sd)↓	ASSD(sd)↓	Sensibilidad(sd)↑	Precisión(sd)↑	Dice(sd)↑	RVD(sd)↓	AUC-PR(sd)↑
L_{GD}	19.28(5.03)	2.81(0.89)	0.684(0.049)	0.670(0.071)	0.655(0.038)	0.350(0.223)	0.678(0.024)
L_{BL}	19.26(4.71)	2.61(0.47)	0.687(0.037)	0.688(0.054)	0.671(0.019)	0.290(0.089)	0.688(0.015)
L_{HD}	19.16(5.21)	2.78(0.72)	**0.705**(0.048)	0.662(0.063)	0.662(0.029)	0.366(0.173)	0.684(0.020)
L_{ABL}	18.87(5.25)	2.63(0.55)	0.673(0.066)	0.693(0.056)	0.660(0.035)	0.328(0.081)	0.683(0.024)
L_{CBL}	19.44(3.98)	2.85(0.60)	0.695(0.040)	0.661(0.041)	0.657(0.024)	0.343(0.125)	0.679(0.016)
L_{BS}	20.04(5.56)	2.81(0.71)	0.677(0.054)	0.682(0.056)	0.661(0.025)	0.319(0.071)	0.681(0.024)
$L_{MDL}^{\lambda=1.5}$ **(Propuesta)**	**17.82**(5.50)	**2.40**(0.42)	0.687(0.041)	**0.704**(0.051)	**0.679**(0.018)	**0.280**(0.066)	**0.696**(0.015)

Nota. Los valores representan la media seguida de la desviación estándar (sd), calculados sobre 20 corridas experimentales. El mejor resultado de cada métrica se destaca en negrita.

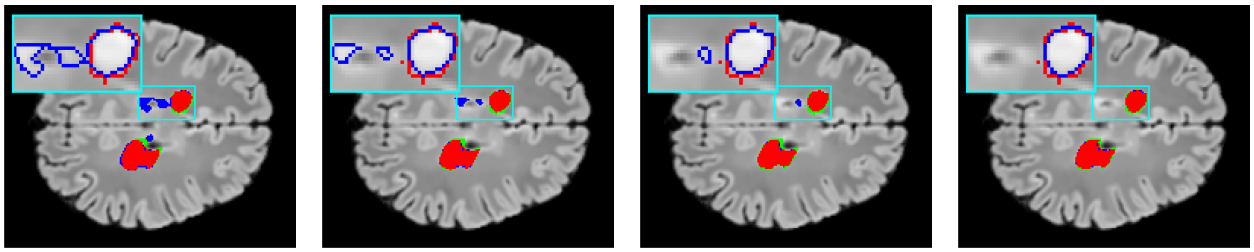
y ASSD: Hausdorff Distance Loss y Boundary Loss respectivamente.

En estas figuras, los vóxeles verdaderos positivos aparecen en rojo, los falsos negativos en verde y los falsos positivos en azul. En ambos conjuntos de datos se observa que Mahalanobis Distance Loss reduce de manera significativa la cantidad de falsos positivos, tanto en términos del número de vóxeles como en la presencia de "lesiones falsas", es decir, regiones marcadas como lesión pero ausentes en el ground truth. Asimismo, en lo concerniente a las métricas de distancia a superficie, la función de pérdida propuesta disminuye los errores no solo en las zonas cercanas a los bordes de las lesiones, sino también en regiones más alejadas, incluyendo falsos positivos remotos. Esto evidencia que el uso del MDM, al integrar información de textura y espacial, mejora tanto la precisión en los bordes como la coherencia global de la segmentación final.



(a) L_{GD} **(b)** L_{HD} **(c)** L_{BL} **(d)** L_{MDL}

Figura 4.3: Comparación cualitativa de los resultados de segmentación en el conjunto de datos ISBI-MS: (**a**) Generalized Dice Loss, (**b**) Hausdorff Distance Loss, (**c**) Boundary Loss y (**d**) Mahalanobis Distance Loss con $\lambda = 1.5$.



(a) L_{GD} **(b)** L_{HD} **(c)** L_{BL} **(d)** L_{MDL}

Figura 4.4: Comparación cualitativa de los resultados de segmentación en el conjunto de datos MSSEG2016: (**a**) Generalized Dice Loss, (**b**) Hausdorff Distance Loss, (**c**) Boundary Loss y (**d**) Mahalanobis Distance Loss con $\lambda = 1.5$.

4.6. Discusión

El análisis comparativo realizado en este trabajo evidencia el impacto que tiene la selección de la función de pérdida en la segmentación de lesiones de esclerosis múltiple en imágenes de resonancia magnética. A través de la evaluación de un conjunto amplio de funciones de pérdida en los conjuntos de datos ISBI-MS y MSSEG2016, se observó que aquellas que integran información de distancia espacial, textural, además del componente de solapamiento espacial (Generalized Dice Loss), como Mahalanobis Distance Loss, obtienen un rendimiento superior tanto en métricas basadas en bordes como en métricas de solapamiento. En particular, la función de pérdida propuesta logra un equilibrio adecuado entre la calidad del

solapamiento global y una mejor definición de los bordes, además es importante destacar una reducción de falsos positivos y una estimación más precisa del volumen de la clase lesión.

Resultados relacionados con la función de pérdida propuesta fueron reportados previamente en (Ulloa-Poblete et al., 2025); sin embargo, la presente tesis amplía dicho trabajo mediante una formulación metodológica más detallada y un análisis más profundo y sistemático de los resultados, enfatizando su interpretación en términos de métricas de borde, reducción de falsos positivos y relevancia clínica.

Estos beneficios, sin embargo, implican un compromiso entre precisión y sensibilidad. Las pérdidas basadas en distancia tienden a obtener un rendimiento levemente inferior en términos de sensibilidad (recall), lo que puede resultar en un aumento en los falsos negativos. Asimismo, el rendimiento limitado que presentan las funciones de pérdida que en su término basado en bordes se centran de manera exclusiva en penalizar solo los vóxeles en los bordes, evidencian las dificultades inherentes a la segmentación de lesiones de esclerosis múltiple, donde el efecto del volumen parcial, la superposición de intensidades entre tejidos y la morfología irregular de las lesiones dificultan la definición de los bordes.

Los resultados obtenidos también destacan la importancia del ajuste empírico del parámetro λ en Mahalanobis Distance Loss para obtener el mejor rendimiento. La consistencia observada en distintos conjuntos de datos demuestra la robustez y capacidad de generalización del método propuesto. Estos buenos resultados pueden sugerir que existe espacio para abordar nuevas variantes de funciones de pérdida basadas en distancia más generales, que incluyan información espacial y de contexto local como textura, que incrementen la sensibilidad sin sacrificar la precisión. De este modo, Mahalanobis Distance Loss se posiciona como una alternativa prometedora para avanzar en la segmentación automática de lesiones de esclerosis múltiple y para fortalecer su aplicación en entornos clínicos reales.

Capítulo 5

Conclusiones Generales y Trabajo Futuro

5.1. Conclusiones

En este trabajo se presentó una nueva función de pérdida, denominada Mahalanobis Distance Loss, orientada a la segmentación automática de lesiones de esclerosis múltiple en imágenes de resonancia magnética cerebral. A diferencia de las funciones de pérdida tradicionales basadas exclusivamente en distancias euclidianas, la propuesta integra características radiómicas locales junto con las coordenadas espaciales de cada vóxel. Mientras que los mapas de distancia euclidiana asumen independencia entre las características y se basan únicamente en información geométrica, el mapa de distancia de Mahalanobis incorpora dependencias estadísticas entre características, permitiendo de este modo capturar variaciones sutiles de textura alrededor a los bordes de las lesiones. De este modo, la función de pérdida propuesta extiende los conceptos clásicos presentes en las funciones de pérdida que utilizan los mapas de distancia DTM y SDF, incorporando un componente textural que modela de manera más precisa la transición entre tejido sano y lesionado.

Los resultados obtenidos demuestran que la función de pérdida propuesta supera de manera consistente a las funciones de pérdida ampliamente utilizadas en la literatura, tanto en métricas basadas en distancia entre bordes (HD95 y ASSD), en métricas basadas en solapamiento espacial (Precisión y Dice), como también en la métrica AUC-PR. Donde esta

última es de especial importancia ya que valida el desempeño del modelo en escenarios de alto desbalance de clases, como es el caso de las lesiones de EM. Incluso, en los casos donde no alcanzó el mejor desempeño en la totalidad de las métricas evaluadas (Recall/Sensibilidad), su rendimiento se mantuvo estable y competitivo, lo cual evidencia robustez frente a variaciones en los conjuntos de datos.

Un aspecto relevante es que el costo computacional de Mahalanobis Distance Loss es comparable al de la Boundary Loss, ya que el mapa de distancia de Mahalanobis se calcula una única vez antes del entrenamiento, de este modo, evitando el alto costo de métodos que requieren actualizar mapas en cada iteración. Además, a diferencia de variantes como Active Boundary Loss, Conditional Boundary Loss y Boundary-Sensitive Loss, que están definidas únicamente para imágenes 2D (slices), la función de pérdida propuesta es aplicable tanto en imágenes 2D como también en volúmenes 3D.

En conjunto, los resultados demuestran que incorporar información textural local dentro del cálculo de penalización contribuye a separar regiones espaciales difíciles típicas de las imágenes de esclerosis múltiple, especialmente en zonas con solapamiento de intensidades debido al efecto del volumen parcial.

5.2. Contribuciones

Las principales contribuciones de este trabajo de tesis se pueden resumir en los siguientes ítems:

1. Propuesta de Mahalanobis Distance Loss, una nueva función de pérdida basada en el mapa de distancia de Mahalanobis, la cual integra información textural y espacial para mejorar la segmentación de lesiones de esclerosis múltiple

2. Extensión del concepto de mapas de distancia mediante la incorporación de características de textura radiómicas locales, superando las limitaciones de los mapas tradicionales basados en distancia euclidiana como DTM y SDF.

3. Demostración empírica sobre dos conjuntos de datos de esclerosis múltiple de público acceso (ISBI-MS y MSSEG2016), donde la función de pérdida propuesta obtiene resultados superiores o comparables a los métodos actuales del estado del arte en múltiples

métricas de evaluación.

4. Formulación de la función de pérdida compatible con imágenes 2D y 3D, a diferencia de otras funciones de pérdida basadas en bordes que solo operan en 2D.

5. Implementación eficiente, ya que el mapa de distancia de Mahalanobis se calcula una única vez antes del entrenamiento, manteniendo un costo computacional similar al de métodos clásicos como Boundary Loss.

6. Análisis del papel del parámetro λ, mostrando estabilidad en distintos conjuntos de datos y validando la capacidad de generalización del método.

5.3. Trabajo futuro

Como continuación de este trabajo, se plantean diversas líneas de investigación. En primer lugar, se pretende incorporar más secuencias de MRI al cálculo del mapa de distancia de Mahalanobis, especialmente la secuencia T1-weighted, con el objetivo de enriquecer la información de textura de lesiones crónicas (antiguas). También, se buscará mejorar la métrica de Sensibilidad (Recall) mediante estrategias orientadas a reducir los falsos negativos sin sacrificar la precisión, es decir, mitigando el impacto en el aumento de los falsos positivos.

Otra posible línea de trabajo contempla acelerar el cálculo del mapa de distancia de Mahalanobis mediante la integración de estrategias computacionales de paralelización de cómputos, lo que permitiría procesar volúmenes 3D de manera más eficiente, sin la necesidad de precalcular los mapas de distancias. También se evaluará la función de pérdida propuesta en otros tipos de lesiones cerebrales, tales como tumores o hiperintensidades de origen vascular, así como en imágenes médicas de distintos órganos presentes en estudios de MRI abdominal.

Finalmente, se explorará la integración de la función de pérdida propuesta en arquitecturas más avanzadas, correspondientes a extensiones de la red U-Net, incluyendo modelos basados en atención (Attention U-Net) y arquitecturas de tipo Transformer.

Apéndices

Apéndice A

Selección de parámetros y resultados complementarios

A.1. Conjuntos de datos

Tabla A.1: Volúmenes del conjunto de datos ISBI-MS

Partición	Volúmenes
1	1-1, 1-2, 1-3, 1-4
2	2-1, 2-2, 2-3, 2-4
3	3-1, 3-2, 3-3, 3-4, 3-5
4	4-1, 4-2, 4-3, 4-4
5	5-1, 5-2, 5-3, 5-4

Tabla A.2: Conjuntos de entrenamiento, validación y prueba del conjunto de datos ISBI-MS

Entrenamiento			Validación			Prueba		
Particiones	#Volúmenes	#Cortes	Particiones	#Volúmenes	#Cortes	Particiones	#Volúmenes	#Cortes
2,3,4	13	1058	5	4	347	1	4	175

Tabla A.3: Volúmenes por partición del conjunto de datos MSSEG2016

Partición	Volúmenes
1	01016SACH, 07010NABO, 08027SYBR
2	01038PAGU, 07001MOEL, 08037ROGU
3	01039VITE, 07043SEME, 08029IVDI
4	01040VANE, 07003SATH, 08031SEVE
5	01042GULE, 07040DORE, 08002CHJE

Tabla A.4: Conjuntos de entrenamiento, validación y prueba del conjunto de datos MSSEG2016

Entrenamiento			Validación			Prueba		
Particiones	#Volúmenes	#Cortes	Particiones	#Volúmenes	#Cortes	Particiones	#Volúmenes	#Cortes
1,2,4	9	742	5	3	195	3	3	157

A.2. Selección de tamaño de batch

La selección de parámetros apunta a determinar la configuración de parámetros que generan los mejores resultados. Para esto se utilizan distintos tipos de métricas, de modo de considerar y evaluar un mayor rango de características del modelo. Para las métricas de solapamiento espacial (subsección 4.4.1) se busca maximizar la amplitud, en cambio para las métricas del tipo distancia entre superficies (subsección 4.4.2) y la métrica Diferencia de volumen relativo (subsección 4.4.3) se busca minimizar la amplitud.

Tabla A.5: Selección de tamaño de batch en conjunto de datos ISBI-MS

batch size	**HD95(sd)↓**	**ASSD(sd)↓**	**Sensitivity(sd)↑**	**Precision(sd)↑**	**Dice(sd)↑**	**rvd(sd)↓**	**auc-pr(sd)↑**
10	16.0465(1.5634)	2.1237(0.2297)	0.6440(0.0254)	0.7963(0.0202)	0.6982(0.0170)	0.2469(0.0417)	0.7206(0.0125)
16	14.6489(1.2063)	2.0348(0.2513)	0.6532(0.0349)	0.7974(0.0237)	0.7037(0.0206)	0.2417(0.0431)	0.7257(0.0145)
24	15.0182(1.1756)	2.0301(0.2104)	0.6476(0.0198)	0.7923(0.0170)	0.6980(0.0150)	0.2370(0.0267)	0.7204(0.0111)

Tabla A.6: Selección de tamaño de batch en conjunto de datos MSSEG2016

batch size	HD95(sd)↓	ASSD(sd)↓	Sensitivity(sd)↑	Precision(sd)↑	Dice(sd)↑	rvd(sd)↓	auc-pr(sd)↑
10	18.0897(0.9583)	3.1345(0.1695)	0.6387(0.0253)	0.7374(0.0213)	0.6586(0.0121)	0.3555(0.0520)	0.6891(0.0088)
16	18.6895(1.8779)	3.3338(0.2184)	0.6610(0.0244)	0.7180(0.0229)	0.6618(0.0094)	0.3877(0.0804)	0.6905(0.0064)
24	19.823(1.5624)	3.3873(0.3058)	0.6664(0.0222)	0.7092(0.0264)	0.6601(0.0079)	0.3988(0.0957)	0.6888(0.0057)

A.3. Sistema distribuido de entrenamiento

Con el objetivo de entrenar el modelo la cantidad de veces necesarias para obtener una buena estimación del desempeño promedio en las diversas métricas, se implementó y utilizó un Sistema Distribuido de Entrenamiento.

Este sistema se diseñó para gestionar de manera eficiente múltiples nodos de procesamiento, aprovechando recursos tanto internos como externos a la Universidad para la ejecución de corridas experimentales en paralelo del entrenamiento del modelo de segmentación. La arquitectura de este sistema fue sometida a una evaluación formal para garantizar la monitorización, el rendimiento y la consistencia de los resultados, validando su robustez y escalabilidad para la experimentación (Ver Figura A.1).

Los detalles de la arquitectura, incluyendo el modelo de comunicación mediante API RESTful y la evaluación de sus drivers de calidad, se describen en detalle en la publicación Cruz et al. (2023).

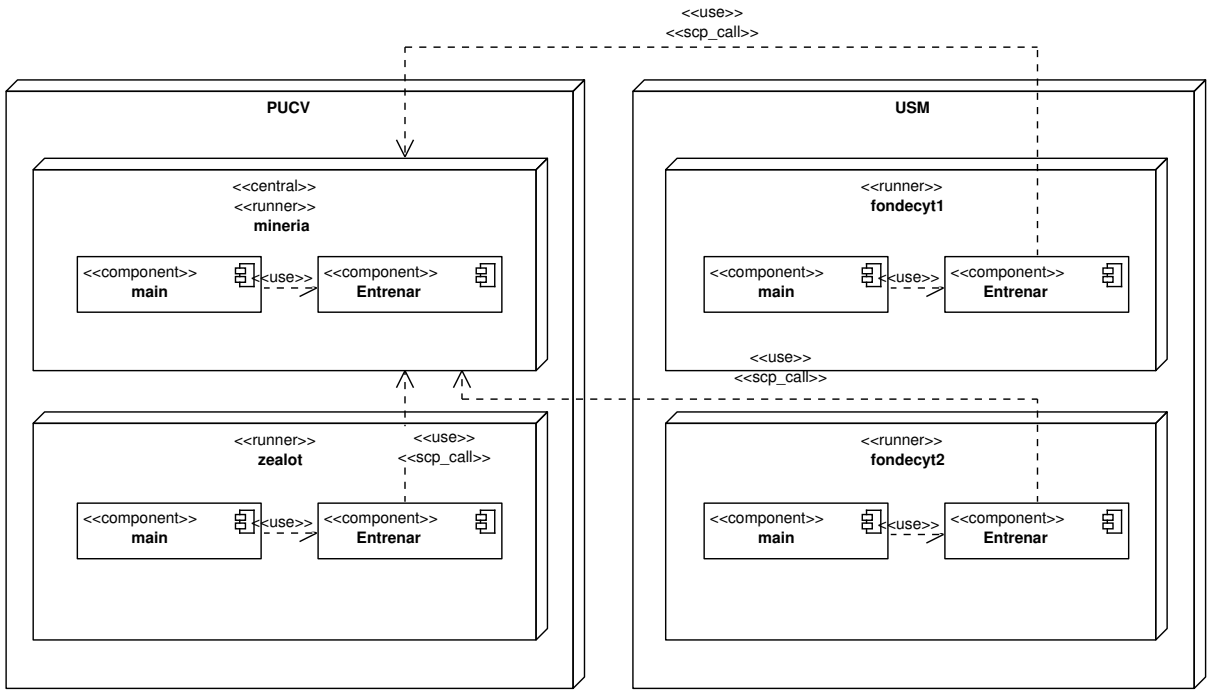


Figura A.1: Sistema Distribuido de Entrenamiento.

Apéndice B

Lista de Publicaciones

Artículos de revistas

Ulloa-Poblete, G., Veloz, A., Sánchez, S. and Allende, H. (2025). *MRI Boundary-Aware Segmentation of Multiple Sclerosis Lesions Using a Novel Mahalanobis Distance Map. Applied Sciences, 15*(19), 10629. https://doi.org/10.3390/app151910629

Ulloa-Poblete, G., Veloz, A., Allende-Cid, H. and Allende, H. (2023). *Edges-enhanced Convolutional Neural Network for Multiple Sclerosis Lesions Segmentation. Computación y Sistemas, 27*(1). https://doi.org/10.13053/CyS-27-1-4535

Artículos de conferencias

Cruz, P., Ulloa, G., San Martin, D. and Veloz, A. *Software Architecture Evaluation of a Machine Learning Enabled System: A Case Study*, 2023 42nd IEEE International Conference of the Chilean Computer Science Society (SCCC), Concepcion, Chile, 2023, pp. 1-8, doi: 10.1109/SCCC59417.2023.10315755.

Ulloa, G., Veloz, A., Allende-Cid, H., Monge, R. and Allende, H. (2022). Efficient Methodology Based on Convolutional Neural Networks with Augmented Penalization on Hard-to-Classify Boundary Voxels on the Task of Brain Lesion Segmentation. In: Vergara-Villegas, O.O., Cruz-Sánchez, V.G., Sossa-Azuela, J.H., Carrasco-Ochoa, J.A., Martínez-Trinidad,

J.F., Olvera-López, J.A. (eds) Pattern Recognition. MCPR 2022. Lecture Notes in Computer Science, vol 13264. Springer, Cham. https://doi.org/10.1007/978-3-031-07750-0_31

Ulloa, G., Veloz, A., Allende-Cid, H. and Allende, H. (2020). *Improving Multiple Sclerosis Lesion Boundaries Segmentation by Convolutional Neural Networks with Focal Learning*. In: Campilho, A., Karray, F., Wang, Z. (eds) Image Analysis and Recognition. ICIAR 2020. Lecture Notes in Computer Science(), vol 12132. Springer, Cham. https://doi.org/10.1007/978-3-030-50516-5_16

Ulloa, G., Naranjo, R., Allende-Cid, H., Chabert, S. and Allende, H. (2019). *Circular Non-uniform Sampling Patch Inputs for CNN Applied to Multiple Sclerosis Lesion Segmentation*. In: Vera-Rodriguez, R., Fierrez, J., Morales, A. (eds) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2018. Lecture Notes in Computer Science(), vol 11401. Springer, Cham. https://doi.org/10.1007/978-3-030-13469-3_78

Bibliografía

Abraham, N. and Khan, N. M. (2019). A novel focal Tversky loss function with improved attention U-Net for lesion segmentation. In *16th IEEE International Symposium on Biomedical Imaging, ISBI 2019, Venice, Italy, April 8-11, 2019*, pages 683–687. IEEE.

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. (1999). *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 3rd edition.

Ansel, J., Yang, E., and He (2024). Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS '24, page 929–947, New York, NY, USA. Association for Computing Machinery.

Azad, R., Aghdam, E. K., Rauland, A., Jia, Y., Avval, A. H., Bozorgpour, A., Karimijafarbigloo, S., Cohen, J. P., Adeli, E., and Merhof, D. (2024). Medical image segmentation review: The success of U-Net. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10076–10095.

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2015). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:2481–2495.

Balafar, M., Ramli, A., Saripan, M., and Mashohor, S. (2010). Review of brain MRI image segmentation methods. *Artificial Intelligence Review*, 33(3):261–274.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Information science and statistics. Springer, 1st ed. 2006. corr. 2nd printing edition.

Boyd, K., Eng, K. H., and Page, C. D. (2013). Area under the precision-recall curve: Point estimates and confidence intervals. In Blockeel, H., Kersting, K., Nijssen, S., and Železný, F., editors, *Machine Learning and Knowledge Discovery in Databases*, pages 451–466, Berlin, Heidelberg. Springer Berlin Heidelberg.

Brosch, T., Yoo, Y., Tang, L. Y. W., Li, D. K. B., Traboulsee, A., and Tam, R. (2015). Deep convolutional encoder networks for multiple sclerosis lesion segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 3–11, Cham. Springer International Publishing.

Bushberg, J. T., Seibert, A. J., Leidholdt, E. M., and Boone, J. M. (2012). *The essential physics of medical imaging; 3rd ed.* Lippincott Williams & Wilkins, Philadelphia, PA.

Carass, A. et al. (2017). Longitudinal multiple sclerosis lesion segmentation: Resource and challenge. *NeuroImage*, 148:77 – 102.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., and Ronneberger, O. (2016). 3d U-Net: Learning dense volumetric segmentation from sparse annotation. In Ourselin, S., Joskowicz, L., Sabuncu, M. R., Unal, G., and Wells, W., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pages 424–432, Cham. Springer International Publishing.

Commowick, O., Istace, A., and et al., M. (2018). Objective evaluation of multiple sclerosis lesion segmentation using a data management and processing infrastructure. *Scientific Reports*, 8.

Cruz, P., Ulloa, G., Martin, D. S., and Veloz, A. (2023). Software architecture evaluation of a machine learning enabled system: A case study. In *2023 42nd IEEE International Conference of the Chilean Computer Science Society (SCCC)*, pages 1–8.

Cui, Y., Jia, M., Lin, T., Song, Y., and Belongie, S. (2019). Class-balanced loss based on effective number of samples. In *2019 IEEE/CVF Conference on Computer Vision and*

Pattern Recognition (CVPR), pages 9260–9269, Los Alamitos, CA, USA. IEEE Computer Society.

Danelakis, A., Theoharis, T., and Verganelakis, D. A. (2018). Survey of automated multiple sclerosis lesion segmentation techniques on magnetic resonance imaging. *Computerized Medical Imaging and Graphics*, 70:83 – 100.

Dhanachandra, N., Manglem, K., and Chanu, Y. J. (2015). Image segmentation using K-means clustering algorithm and subtractive clustering algorithm. *Procedia Computer Science*, 54:764–771.

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

Dobson, R. and Giovannoni, G. (2019). Multiple sclerosis – a review. *European Journal of Neurology*, 26(1):27–40.

Dougherty, G. (2009). *Digital image processing for medical applications*. Cambridge.

Du, J., Guan, K., Liu, P., Li, Y., and Wang, T. (2023). Boundary-sensitive loss function with location constraint for hard region segmentation. *IEEE Journal of Biomedical and Health Informatics*, 27(2):992–1003.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159.

Dumoulin, V. and Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*.

Fazekas, F., Barkhof, F., Filippi, M., Grossman, R., Li, D., McDonald, W., McFarland, H., Paty, D., Simon, J., Wolinsky, J., and Miller, D. (1999). The contribution of magnetic resonance imaging to the diagnosis of multiple sclerosis. *Neurology*, (53):448–456.

Fonov, V., Evans, A. C., Botteron, K., Almli, C. R., McKinstry, R. C., and Collins, D. L. (2011). Unbiased average age-appropriate atlases for pediatric studies. *NeuroImage*, 54(1):313–327.

Frery, A. C. and Perciano, T. (2013). *Introduction to Image Processing Using R: Learning by Examples*. Springer.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics (Oxford, England)*, 9:432–41.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202.

Galloway, M. M. (1975). Texture analysis using gray level run lengths. *Computer Graphics and Image Processing*, 4(2):172–179.

Gao, Y., Li, D., Lin, J., Thomas, A. M., Miao, J., Chen, D., Li, S., and Chu, C. (2022). Cerebral small vessel disease: Pathological mechanisms and potential therapeutic targets. *Frontiers in Aging Neuroscience*, 14.

Giorgio, A. and Stefano, N. D. (2018). Effective utilization of MRI in the diagnosis and management of multiple sclerosis. *Neurologic Clinics*, 36(1):27 – 34. Multiple Sclerosis.

Gonzalez, R. C. and Woods, R. E. (2001). *Digital image processing*. Prentice Hall, second edition.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

Haider, L. and Lassmann, H. (2024). Gray matter pathology in multiple sclerosis. *Nature Reviews Neurology*, 20(2):85–100.

Halko, N., Martinsson, P.-G., and Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288.

Haralick, R. M., Shanmugam, K., and Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621.

Hashemi, S. R., Mohseni Salehi, S. S., Erdogmus, D., Prabhu, S. P., Warfield, S. K., and Gholipour, A. (2019). Asymmetric loss functions and deep densely-connected networks for highly-imbalanced medical image segmentation: Application to multiple sclerosis lesion detection. *IEEE Access*, 7:1721–1735.

Hashemi, S. R., Salehi, S. S. M., Erdogmus, D., Prabhu, S. P., Warfield, S. K., and Gholipour, A. (2018). Tversky as a loss function for highly unbalanced image segmentation using 3d fully convolutional deep networks. *CoRR*, abs/1803.11078.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Hinton, G. (2012). Lecture 6.5 — rmsprop: normalize the gradient. *Coursera/Youtube: Neural Networks for Machine Learning*.

Jadon, S. (2020). A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7.

Jenkinson, M. (2005). Bet2 : Mr-based estimation of brain, skull and scalp surfaces. *Eleventh Annual Meeting of the Organization for Human Brain Mapping, 2005*.

Jenkinson, M., Bannister, P., Brady, M., and Smith, S. (2002). Improved optimization for the robust and accurate linear registration and motion correction of brain images. *NeuroImage*, 17(2):825 – 841.

Jenkinson, M., Beckmann, C. F., Behrens, T. E., Woolrich, M. W., and Smith, S. M. (2012). Fsl. *NeuroImage*, 62(2):782 – 790. 20 YEARS OF fMRI.

Jia, W., Sun, M., Lian, J., and Hou, S. (2022). Feature dimensionality reduction: a review. *Complex & Intelligent Systems*, 8:2663–2693.

John Hall, M. H. (2020). *Guyton and Hall Textbook of Medical Physiology*. Guyton Physiology. Elsevier, 14th edition edition.

Karimi, D. and Salcudean, S. E. (2020). Reducing the Hausdorff distance in medical image segmentation with convolutional neural networks. *IEEE Transactions on Medical Imaging*, 39(2):499–513.

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

Kervadec, H., Bouchtiba, J., Desrosiers, C., Granger, E., Dolz, J., and Ben Ayed, I. (2021). Boundary loss for highly unbalanced segmentation. *Medical Image Analysis*, 67:101851.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lin, T., Goyal, P., Girshick, R., He, K., and Dollár, P. (2020). Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327.

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.

Liu, S. and Deng, W. (2015). Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 730–734.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, Los Alamitos, CA, USA. IEEE Computer Society.

Ma, J., Wei, Z., Zhang, Y., Wang, Y., Lv, R., Zhu, C., Gaoxiang, C., Liu, J., Peng, C., Wang, L., Wang, Y., and Chen, J. (2020). How distance transform maps boost segmentation CNNs: An empirical study. In Arbel, T., Ben Ayed, I., de Bruijne, M., Descoteaux, M., Lombaert, H., and Pal, C., editors, *Proceedings of the Third Conference on Medical Imaging with Deep Learning*, volume 121 of *Proceedings of Machine Learning Research*, pages 479–492. PMLR.

Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, volume 30, page 3, Atlanta, Georgia, USA.

Mayerhoefer, M., Materka, A., Langs, G., Häggström, I., Szczypiński, P., Gibbs, P., and Cook, G. (2020). Introduction to radiomics. *Journal of nuclear medicine*, 4(61):488–495.

Milletari, F., Navab, N., and Ahmadi, S.-A. (2016). V-Net: Fully convolutional neural networks for volumetric medical image segmentation. *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571.

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., and Terzopoulos, D. (2022). Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3523–3542.

Naga Karthik, E., McGinnis, J., Wurm, R., Ruehling, S., Graf, R., Valosek, J., Benveniste, P.-L., Lauerer, M., Talbott, J., Bakshi, R., Tauhid, S., Shepherd, T., Berthele, A., Zimmer, C., Hemmer, B., Rueckert, D., Wiestler, B., Kirschke, J. S., Cohen-Adad, J., and Mühlau, M. (2025). Automatic segmentation of spinal cord lesions in ms: A robust tool for axial t2-weighted MRI scans. *Imaging Neuroscience*, 3:IMAG.a.45.

Nai, Y.-H., Teo, B. W., Tan, N. L., O'Doherty, S., Stephenson, M. C., Thian, Y. L., Chiong, E., and Reilhac, A. (2021). Comparison of metrics for the evaluation of medical segmentations using prostate MRI dataset. *Computers in Biology and Medicine*, 134:104497.

Nock, R. and Nielsen, F. (2004). Statistical region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1452–1458.

Noh, H., Hong, S., and Han, B. (2015). Learning deconvolution network for semantic segmentation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1520–1528, Los Alamitos, CA, USA. IEEE Computer Society.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Plath, N., Toussaint, M., and Nakajima, S. (2009). Multi-class image segmentation using conditional random fields and global classification. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 817–824. ACM.

Poggio, T., Rifkin, R., Mukherjee, S., and Niyogi, P. (2004). General conditions for predictivity in learning theory. *Nature*, 428:419–422.

Punn, N. S. and Agarwal, S. (2022). Modality specific U-Net variants for biomedical image segmentation: a survey. *Artificial Intelligence Review*, 55(7):5845–5889.

Richard L Drake, A Wayne Vogl, A. W. M. M. (2022). *Gray's Anatomy for Students*. Elsevier, 4 edition.

Romo-Sanchez M, Nelson F, S.-D. M. (2020). Magnetic resonance imaging in multiple sclerosis: a review of the basic principles and practical guidelines. *Archivos de Neurociencias*, 25(4):23–31.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham. Springer International Publishing.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*.

Smith, N. B. and Webb, A. (2011). *Introduction to medical imaging: Physics, Engineering and Clinical Applications*. Cambridge.

Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., and Jorge Cardoso, M. (2017). Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations. In Cardoso, M. J., Arbel, T., Carneiro, G., Syeda-Mahmood, T., Tavares, J. M. R., Moradi,

M., Bradley, A., Greenspan, H., Papa, J. P., Madabhushi, A., Nascimento, J. C., Cardoso, J. S., Belagiannis, V., and Lu, Z., editors, *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 240–248, Cham. Springer International Publishing.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA. PMLR.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, Los Alamitos, CA, USA. IEEE Computer Society.

Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B., and Liang, J. (2016). Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Transactions on Medical Imaging*, 35(5):1299–1312.

Taye, M. M. (2023). Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions. *Computation*, 11(3).

Terven, J., Cordova-Esparza, D. M., Romero-González, J. A., et al. (2025). A comprehensive survey of loss functions and metrics in deep learning. *Artificial Intelligence Review*, 58:195.

Theng, D. and Bhoyar, K. K. (2024). Feature selection techniques for machine learning: a survey of more than two decades of research. *Knowledge & Information Systems*, 66(3):1575–1637.

Tversky, A. (1977). Features of similarity. *Psychological Review*, 84(4):327–352.

Ulloa-Poblete, G., Veloz, A., Sánchez, S., and Allende, H. (2025). MRI boundary-aware segmentation of multiple sclerosis lesions using a novel Mahalanobis distance map. *Applied Sciences*, 15(19).

Vapnik, V. N. (1998). *Statistical learning theory*. Wiley-Interscience.

Varmuza, K. and Filzmoser, P. (2009). *Introduction to Multivariate Statistical Analysis in Chemometrics*. Academic Press.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. (1989). Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):328–339.

Wang, C., Zhang, Y., Cui, M., Liu, J., Ren, P., Yang, Y., Xie, X., Hua, X., Bao, H., and Xu, W. (2021). Active boundary loss for semantic segmentation. In *AAAI Conference on Artificial Intelligence*.

Warfield, S. K., Zou, K. H., and Wells, W. M. (2004). Simultaneous truth and performance level estimation (staple): An algorithm for the validation of image segmentation. *IEEE Transactions on Medical Imaging*, 23(7):903–921.

Wong, K. C. L., Moradi, M., Tang, H., and Syeda-Mahmood, T. (2018). 3d segmentation with exponential logarithmic loss for highly unbalanced object sizes. In Frangi, A. F., Schnabel, J. A., Davatzikos, C., Alberola-López, C., and Fichtinger, G., editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, pages 612–619, Cham. Springer International Publishing.

Wu, D., Guo, Z., Li, A., Yu, C., Gao, C., and Sang, N. (2023). Conditional boundary loss for semantic segmentation. *IEEE Transactions on Image Processing*, 32:3717–3731.

Xia, Q., Zheng, H., Zou, H., Luo, D., Tang, H., Li, L., and Jiang, B. (2025). A comprehensive review of deep learning for medical image segmentation. *Neurocomputing*, 613:128740.

Xie, S. and Tu, Z. (2015). Holistically-nested edge detection. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1395–1403.

Yeghiazaryan, V. and Voiculescu, I. (2018). Family of boundary overlap metrics for the evaluation of medical image segmentation. *Journal of medical imaging (Bellingham, Wash.)*.

Yi-de, M., Qing, L., and Zhi-bai, Q. (2004). Automated image segmentation using improved PCNN model based on cross-entropy. In *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.*, pages 743–746.

Yu, Y., Wang, C., Fu, Q., Kou, R., Huang, F., Yang, B., Yang, T., and Gao, M. (2023). Techniques and challenges of image segmentation: A review. *Electronics*, 12(5).

Zhang, Y., Brady, M., and Smith, S. (2001). Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *IEEE Transactions on Medical Imaging*, 20(1):45–57.

Zhou, T., Ruan, S., and Canu, S. (2019). A review: Deep learning for medical image segmentation using multi-modality fusion. *Array*, 3-4:100004.