UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA

FACULTAD DE INGENIERÍA

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

FF-TRANS-UNET: MODELO DE SEGMENTACIÓN DE IMÁGENES MÉDICAS CON FUNCIÓN DE PÉRDIDA FOCAL DIFUSA

POR:

ADRIÁN TALAMANTES ROMÁN

TESIS PRESENTADA COMO REQUISITO PARA OBTENER EL GRADO DE

MAESTRO EN INGENIERÍA EN COMPUTACIÓN

CHIHUAHUA, CHIH., MÉXICO MARZO 2024

FF-TRANS-UNET: Modelo de Segmentación de Imágenes Médicas con Función de Pérdida Focal Difusa. Tesis presentado por Adrián Talamantes Román como requisito parcial para obtener el grado de Maestro en Ingeniería en Computación, ha sido aprobado y aceptado por:

M.I. Fabián Vinicio Hernández Martínez
Director de la Facultad de Ingeniería

Dr. Fernando Martínez Reyes
Secretario de Investigación y Posgrado

M.S.I. Karina Rocío Requena Yáñez
Coordinadora Académica

Graciela Ramirez A.

Dra. Graciela María de Jesús Ramírez Alonso
Directora de Tesis

Marzo 2024

Fecha

COMITÉ
Dra. Graciela María de Jesús Ramírez Alonso
Dr. Juan A. Ramírez Quintana
Dr. Jesús Roberto López Santillán
M.A. Olanda Prieto Ordaz

UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA

09 de abril de 2024.

ING. ADRIAN TALAMANTES ROMAN
Presente. -

En atención a su solicitud relativa al trabajo de tesis para obtener el grado de Maestría en Ingeniería en Computación, nos es grato transcribirle el tema aprobado por esta Dirección, propuesto y dirigido por la directora **Dra. Graciela María de Jesús Ramírez Alonso** para que lo desarrolle como Tesis, con el título **"FF-TRANS-UNET: Modelo de Segmentación de Imágenes Médicas con Función de Pérdida Focal Difusa"**.

Índice de Contenido

FACULTAD DE INGENIERÍA
Circuito No. 1, Campus Universitario II
Tel. (614) 442-9500
Chihuahua, Chih., México

UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA

4. Resultados

4.1. Modelos que utilizan la base de datos CHAOS

4.2. Resultados con modelos implementados

4.3. Resultados con el modelo propuesto: FF-TransUnet

5. Conclusiones y recomendaciones

6. Anexos

A T E N T A M E N T E

"naturam subiecit aliis"

EL DIRECTOR

M.I. FABIÁN VINICIO HERNÁNDEZ MARTÍNEZ

FACULTAD DE INGENIERÍA U.A.CH.

SECRETARIO DE INVESTIGACIÓN Y POSGRADO

DR. FERNANDO MARTÍNEZ REYES

DIRECCIÓN

Resumen

En la actualidad, el diagnóstico asistido por computadora desempeña un papel fundamental en la identificación y análisis de lesiones u órganos internos afectados por diversas causas. La mayoría de estas técnicas se basan en modelos de segmentación semántica que permiten estudiar tomografías computarizadas y resonancias magnéticas.

En este trabajo, se propone el desarrollo de un modelo de segmentación de imágenes médicas en órganos abdominales utilizando la base de datos CHAOS que utiliza una arquitectura encoder-decoder que incorpora modelos de atención y una función de pérdida focal con un enfoque difuso para ajustar la función de pérdida.

Los resultados muestran que se logra un desempeño competitivo en comparación con modelos del estado del arte basados en arquitecturas neuronales *UNet, conexiones residuales* y *transformers*.

Finalmente, se analizan los resultados mediante pruebas estadísticas, utilizando como variables, los resultados obtenidos con el coeficiente DICE de segmentación y el número de parámetros de cada modelo.

Estas pruebas nos ayudan a decidir si un modelo es significativamente diferente a otro, y, por lo tanto, poder tomar una decisión a la hora de decidir utilizar alguno de los modelos presentados.

El enfoque difuso en la función de pérdida focal proporciona una metodología sofisticada y efectiva para adaptar de manera inteligente el proceso de entrenamiento para responder a las variaciones en los tamaños de las clases y la evolución de la pérdida.

Palabras clave: CHAOS, segmentación órganos abdominales, *transformers*, lógica difusa.

Índice de figuras

Índice de tablas

Índice general

CAPÍTULO 1. Introducción

Uno de los campos de la inteligencia artificial (AI), es el aprendizaje máquina (ML), donde se encuentra el área del aprendizaje profundo (DL) [1]. Este último puede aplicarse a una cantidad de áreas diferentes, desde la medicina analizando diferentes tipos de imágenes médicas o señales obtenidas de electroencefalogramas [2], hasta la industria de los videojuegos que buscan incorporar sistemas de AI para hacerlos más entretenidos o desafiantes [3].

El aprendizaje profundo tiene varios problemas fundamentales en el reconocimiento de imágenes, en específico de imágenes médicas, que son: clasificación de imágenes, detección de objetos, segmentación de instancias y segmentación semántica [4]. La finalidad de la segmentación de imágenes médicas es localizar regiones con significado, es decir, generar una imagen que identifique diferentes regiones sin superponerse una sobre otra; en específico en la región abdominal, separar en primera instancia los órganos abdominales y como siguiente punto, identificar anomalías o tumores en alguno de los órganos segmentados.

Las imágenes de Resonancia magnética (RM) son una de las técnicas diagnósticas más importantes en la detección de lesiones o anomalías hepáticas. Los estudios que utilizan técnicas con RM tienen resultados prometedores para la detección y diagnóstico de tumores hepáticos [5]. Otra técnica que se utiliza para realizar una estimación preoperatoria es la Tomografía computarizada (TC) para la detección de enfermedades hepáticas [6].

El procedimiento para llevar a cabo el análisis de las RM y TC es tardado y puede tomar días en tener un resultado de estos análisis a los médicos especialistas. El tiempo que se requiere para este procedimiento puede ser decisivo para detectar a tiempo un tumor o alguna lesión hepática en el hígado u otro órgano abdominal. De aquí se reconoce la necesidad de emplear métodos de aprendizaje automático, que ayuden a los encargados de realizar las interpretaciones de las RM y TC. Algunos modelos computacionales que han reportado destacables resultados en tareas de segmentación en imágenes médicas, se presentan a continuación.

Dentro de los modelos para analizar, y más específicamente, segmentar imágenes médicas, se encuentra el modelo U-Net, que tiene una arquitectura de red neuronal convolucional [7]. Es ideal para la segmentación, por el hecho de que las imágenes de salida son del mismo tamaño que las que se proporcionan como entrada. El modelo U-Net tiene, como bloque de entrada un encoder, el cual se compone de varias capas convolucionales que reducen la dimensionalidad en ancho y alto de la entrada (generalmente usando conexiones residuales). El bloque de salida decoder, se encarga de ajustar la dimensión a un tamaño igual al de entrada. Este tipo de modelos cuenta con un número de parámetros entrenables en promedio de 34 millones.

El modelo PRDNet usa una red de segmentación semántica, con una modificación de la red neuronal ResNet [8]. Las 5 capas del *encoder* reducen la imagen de entrada a dimensiones menores, la imagen de salida producida en el *decoder*, se alimenta de las capas 4 y 5 del encoder, utilizando convoluciones dilatadas. Finalmente, se fusionan todas las imágenes de salida de cada capa para generar una imagen de dimensiones iguales a las originales, segmentando los órganos abdominales de resonancias magnéticas (RM - T1 DUAL) de la base de datos "Combined (CT-MR) Healthy Abdominal Organ Segmentation Challenge data" (CHAOS). Dicha base de datos se conforma de 40 pacientes, de los cuales 20 cuentan con información del *groundtruth*. Dichos pacientes son utilizados para entrenar, validar y probar el modelo considerando una partición de 65 %, 10 % y 25 %, respectivamente. Los autores reportan una eficiencia con el coeficiente DICE de 90,2 %. La red se basa en una estructura similar a una U-Net, de manera que tiene un número de parámetros aproximado de 34 millones.

Por otro lado, la RMS-UNet propone una función objetivo para el tratamiento de píxeles del *foreground* y del *background*, usando una normalización para reducir la pérdida de información en las imágenes de la base de datos CHAOS enfocándose únicamente en la segmentación del hígado. La red neuronal utiliza convolución dilatada basada en el modelo U-Net para entrenar la segmentación de extremo a extremo [9]. Para este modelo se usaron las tomografías de únicamente 10 pacientes, de manera que 70 % se usaron en entrenamiento, 15 % para validar y el resto para pruebas. Los resultados que obtuvo la red RMS-UNet para el coeficiente DICE son de 95,49 %. El número de parámetros entrenables aproximado que tiene es de 34 millones, esto se debe a que

cuenta con una estructura de red U-Net.

En la red MiDeepSeg se propone un método de segmentación interactiva, donde la gran diferencia con los modelos anteriores es que el modelo se alimenta mediante la interacción del usuario, y no solo con una base de datos [10]. El usuario debe realizar una cantidad mínima de clics como entrada, logrando una alta eficiencia para reconocer objetos que nunca se han usado para el entrenamiento de la red. La forma de lograrlo es utilizando puntos del margen interior que el usuario proporciona a la red, esta entrada logra una buena segmentación con una red neuronal convolucional (*CNN*), por último, se fusiona la información inicial con la de la segmentación, usando unos pocos clics adicionales, con la finalidad de refinar la segmentación final. Para los resultados obtenidos de esta red con la base de datos CHAOS, se usaron los datos de 2 pacientes en cada modalidad, tanto de las tomografías computarizadas como de las resonancias magnéticas (T1 y T2) que cuentan con *groundtruth*, solo para la etapa de validación, para segmentar en un caso los riñones y en otro caso el bazo. MiDeepSeg cuenta con un modelo U-Net que le ayuda en la segmentación, a diferencia del resto de métodos, al ser interactivo, cuenta con una estructura diferente a la hora de tomar las entradas para la red, por lo que su número aproximado de parámetros entrenables es de 34 millones.

En la red DCACNet se propone un método de agregación de contexto dual y una red de deconvolución guiada por atención, donde se capturan características contextuales de diferentes escalas en las imágenes que se van a segmentar [11]. La red está conformada por 3 partes: el *encoder*, el módulo de atención DCAM y el *decoder*. El número de parámetros entrenables de esta red, es aproximado a los 34 millones, por contar con una red U-Net como base. En el modelo se utilizaron las imágenes de resonancia magnética (T1-DUAL, en fase), de donde se usaron únicamente los 20 pacientes que cuentan con un *groundtruth*, dividiendo los datos en proporción de 60 % para el entrenamiento, 20 % para la validación y el restante 20 % para las pruebas. La media de la segmentación para los 4 órganos abdominales fue de 91,03 %, teniendo un mayor desempeño en el hígado con un 94,08 %.

La FFANet se basa en agregar características al modelo, usando una red VoVNet como la

columna vertebral del modelo [12], la cual consiste en módulos de *One-Shot Aggregation* los que en lugar de tener las *skip-connections* entre cada capa de convolución, tiene las *skip-connections* de cada capa conectadas hasta el final [13], como muestra la Figura 1.1. Para la capa final, se incluye un módulo de atención, en donde se considera la relevancia de cada espacio y canal [14]. Al contar con el modelo U-Net y un módulo de atención, el número de parámetros entrenables que reporta el modelo FFANet es de 37,24 millones. Para los datos, se usaron las imágenes que tuvieran *groundtruth* de las resonancias magnéticas, los 20 pacientes, de las cuales se dividieron en 3 subconjuntos, guardando una proporción de entrenamiento, validación y prueba de: 65 %, 10 % y 25 % respectivamente obteniendo un coeficiente DICE de 90,9 %.

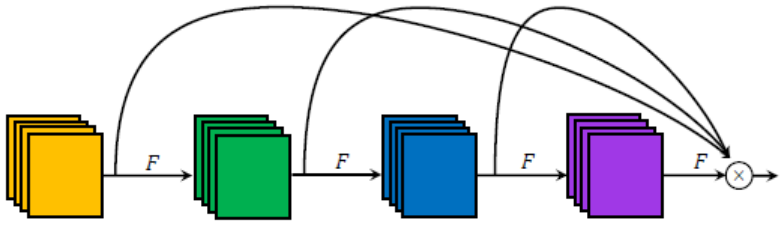


Figura 1.1: Estructura básica de la red VoVNet.

El concepto de transferencia de aprendizaje o *transfer learning* ayuda para utilizar lo que se aprendió en otros problemas, y llevar ese conocimiento a un nuevo problema[15]. Este concepto se puede combinar con los modelos de atención, que están inspirados en el comportamiento humano, donde el cerebro se centra en una parte de un texto, imagen o cualquier percepción [16] y le da menos importancia a otras. Con esto, un modelo de atención en una red neuronal permite asignar diferentes pesos a cada estado en la red, por lo que al realizar la suma ponderada de cada estado permite que la salida se componga de diferentes estados y no únicamente del estado anterior [17]. Esto funciona de manera que en un modelo para segmentación, la atención se utiliza para fusionar características de diferentes niveles de un codificador, desde el actual hasta los niveles posteriores. Al realizar esto, el decodificador realiza un *upsampling* con la información de cada nivel de atención, usando los estímulos relevantes para la segmentación [18].

Un *transformer* es una herramienta que puede utilizarse en tareas donde se tienen pocos

datos, es decir, poco entrenamiento previo. Uniendo los *transformers* con una red *UNet*, se tiene un modelo novedoso *TransUNet*, que ha sido usado en la perspectiva de secuencia-secuencia, aprovechando la información de toda la red de manera detallada [19]. Este tipo de redes, heredan méritos tanto de las CNN como de los *transformers* para capturar detalles locales (como el color de una lesión en la piel o su textura), así como el contexto de rango largo (como la forma de una lesión o el tamaño físico). La red U-Net es débil modelando dependencias de largo alcance, por lo que modelando regiones grandes es inferior. Para abordar esto, la *TransUNet* presenta un *encoder* fuerte incorporando varias capas de *transformers* durante la extracción de características.

El desbalanceo de datos es un problema con la segmentación de imágenes médicas, donde algunas clases pueden estar sobre representadas en el conjunto de entrenamiento. La función de pérdida focal es una de las numerosas estrategias que se han propuesto para abordar este problema. Esta función de pérdida, presentada por Lin et al. en 2017 [20], tiene la ventaja de centrarse en los píxeles difíciles de clasificar, lo que ayuda a concentrar el aprendizaje del modelo en las áreas más pertinentes y difíciles de segmentar.

La función de pérdida focal combina la entropía cruzada, con un término de ponderación que penaliza más a los píxeles difíciles de clasificar. De esta manera, el modelo se enfoca en los casos más complejos y, al mismo tiempo, reduce la influencia de los píxeles bien clasificados. Se ha demostrado que esta táctica es efectiva para abordar el desbalance de clases en problemas de detección de objetos y segmentación.

La función de pérdida focal es utilizada por algunos modelos de segmentación para mejorar su desempeño en bases de datos desbalanceadas. Por ejemplo, en el campo de la detección de objetos, Retinanet, una red de detección basada en un enfoque de una sola etapa, aborda el problema de clasificación desbalanceada en las detecciones mediante el uso de la función de pérdida focal [21].

Aunque la función focal tiene ventajas en el manejo del desequilibrio de clases, tiene un problema: utiliza un valor fijo para el término de ponderación durante todo el entrenamiento. Es-

to puede resultar en imágenes o datos que afectan demasiado el proceso de aprendizaje. Es aquí donde surge la importancia de los modelos que, dependiendo de los datos disponibles, permitan actualizar dinámicamente los parámetros de la función de pérdida focal y su relevancia para la tarea de segmentación.

En este sentido, el modelo *FF-TransUNet* propuesto en este trabajo de investigación es muy útil porque incorpora la función de pérdida focal y proporciona una actualización automática durante el entrenamiento basado en un enfoque difuso del parámetro alfa de la función de pérdida focal. Al permitir que el modelo ajuste este parámetro dinámicamente, se logra una adaptación más efectiva y eficiente a las características específicas de la base de datos. Esto mejora el rendimiento del modelo en situaciones en las que hay desequilibrio de clases y variabilidad en los datos.

El modelo *FF-TransUNet* es especialmente prometedor para la segmentación precisa de imágenes médicas, ofreciendo una herramienta más robusta y flexible para el diagnóstico asistido por computadora en el campo de la medicina. Esto se debe a la capacidad de actualización automática del parámetro alfa de la función de pérdida focal, así como a la combinación de la arquitectura TransUNet que captura detalles locales y contextuales.

Utilizando los conceptos de CNN, modelos de segmentación, modelos transformers, la función de perdida focal, la base de datos CHAOS, y teoría de lógica difusa, en este trabajo se propone un modelo de segmentación de imágenes médicas con una arquitectura TransUNet que obtiene resultados competitivos comparados con el estado del arte. En la Tabla 1.1 se muestra un resumen de algunos modelos que han utilizado la base de datos CHAOS en los años recientes con un resumen del equipo utilizado, parámetros entrenables de sus modelos y la partición de datos utilizados. Cada uno de los trabajos mencionados utilizan tanto imágenes de RM como de TC, las cuales ejercen un papel fundamental en el diagnóstico temprano de enfermedades y masas que se encuentran en el cuerpo, ayudando a un oportuno tratamiento de enfermedades como el cáncer y anomalías hepáticas.

Gracias a su amplio campo de visión y contraste en el tejido, las TC y las RM, generan

imágenes que tienen individualidad, logrando un mejor entendimiento de lo que se observa y de donde se hará la revisión pertinente. Estas técnicas son sumamente importantes para el diagnóstico y tratamiento posterior de dichas lesiones o tumores [5] [6]; las cuales se pueden complementar con técnicas de segmentación de imágenes médicas con modelos de redes neuronales artificiales como son las redes neuronales convolucionales (CNN).

Tabla 1.1: En el siguiente resumen del estado del arte, se presenta un análisis de diversos trabajos de investigación en el campo de la segmentación de imágenes médicas. Este análisis incluye el porcentaje del coeficiente DICE reportado en cada investigación, la modalidad en la que se trabajaron los datos (utilizando exclusivamente la base de datos CHAOS), la partición de los mismos (en la que se consideraron únicamente 20 pacientes del *dataset*), el equipo de trabajo que se empleó y el número de parámetros aproximado en cada modelo.

Trabajos	Año	Autor	DICE	Modalidad	Train-Val-Test	Equipo	Parámetros entrenables
PRDNet: Medical image segmentation based on parallel residual and dilated network	2020	Haojie Guo	90,2 %	RM T1 Dual - Segmentación de 4 órganos abdominales	65 % – 10 % – 25 %	NA	+34M
RMS-Unet: Residual multi-scale Unet for liver and lesion segmentation	2022	Rayyan Azam Khan	95,49 %	TC - Segmentación del hígado	70 % – 15 % – 15 %	Python 3.7, Keras y Tensorflow, intel Xenon (2.6 GHz), 30 GB Ram y Nvidia K40 GPU	NA
MIDeepSeg: Minimally interactive segmentation of unseen objects from medical images using deep learning	2021	Xiangde Luo	96,93 %	RM (T1 y T2) y TC - Segmentación del bazo y riñones	2 pacientes	Ubuntu (16.04), Nvidia 1080Ti, Intel I7, 120 GB memory. Pytorch	NA
DCACNet: Dual Context aggregation and attention-guided cross deconvolution network for medical image segmentation	2021	Hongchun Lu	91,03 %	RM T1 Dual - Segmentación de 4 órganos abdominales	60 % – 20 % – 20 %	Nvidia Tesla V100	+34M
FFANet - Feature fusion attention network to medical image segmentation	2021	Jiankang Yu	90,90 %	RM - Segmentación de 4 órganos abdominales	65 % – 10 % – 25 %	PyTorch, Nvidia 1080Ti e Intel I7-8700 K CPU	37,24 M

Tabla 1.2: Resumen del estado del arte en el que se menciona el porcentaje del coeficiente DICE reportado en cada investigación, la base de datos usada, la partición de los mismos, el equipo de trabajo que se empleó y el número de parámetros aproximado en cada modelo.

Trabajos	Año	Autor	DICE	Base de datos	Train-Val-Test	Equipo	Parámetros entrenables
TransUNet: Transformers make strong encoders for medical image segmentation	2021	Jieneng Chen	89,71 %	ACDA (Automated cardiac diagnosis challenge)	70 % – 10 % – 20 %	Nvidia RTX2080Ti GPU	105,28M
EG-TransUNet: Enhanced and Guided U-Net with Transformer for Biomedical Image Segmentation	2022	Shaoming Pan	90,75 %	ISIC-2018 Challenge	80 % – 10 % – 10 %	Pytorch, Nvidia GeForce 3090 GPU, 12 GB Ram	+105,28M
DS-TransUnet: Dual Swin Transformer U-Net for Medical Image Segmentation	2022	Ailiang Lin	91,30 %	ISIC-2018 Challenge	70 % – 0 % – 30 %	Pytorch, Nvidia RTX3090 GPU	171,44M – 287,75M
Multi-Compound Transformer for Accurate Biomedical Image	2021	Yuanfeng Ji	84,22 % y 90,35 %	Pannuke dataset e ISIC-2018 Challenge	70 % – 0 % – 30 %	Pytorch, V100 GPU	7,642M - 23,79M

El objetivo principal de esta investigación es proponer un modelo de segmentación de imágenes médicas, tomadas de la base de datos CHAOS, basado en una arquitectura encoder-decoder incluyendo modelos de atención y una función de pérdida focal con un enfoque difuso para el ajuste de parámetros que logre resultados competitivos de acuerdo con el estado del arte.

A continuación se muestran los objetivos específicos que guiarán este trabajo:

1. Identificar literatura con respecto a las técnicas de extracción de características y los diversos modelos de segmentación que se utilizan en el estado del arte.
2. Identificar los diferentes métodos de segmentación que usaron en la base de datos CHAOS.
3. Comparar los métodos de segmentación de imágenes médicas que utilicen la base de datos CHAOS, que registren buenos resultados.

4. Desarrollar un método de segmentación de imágenes médicas que utilicen la base de datos CHAOS, usando una arquitectura encoder-decoder.

5. Incorporar modelos de atención a la red propuesta.

6. Utilizar un modelo difuso para actualizar automáticamente el parámetro alfa de la función de pérdida focal.

7. Comparar el desempeño del modelo propuesto con el estado del arte.

Los objetivos precisan las acciones concretas que se llevarán a cabo a medida que se va desarrollando el trabajo propuesto, a fin de contribuir a la optimización del sistema de segmentación de imágenes.

Al igual que en las investigaciones revisadas en el estado del arte, este proyecto de investigación busca una mejora utilizando redes de aprendizaje profundo, de manera que sea un modelo ligero y competitivo de acuerdo a las métricas que se utilizan para comparar la efectividad de la segmentación.

CAPÍTULO 2. Marco Teórico

En este capítulo se hará un repaso de los conceptos fundamentales en el documento, como son: la base de datos utilizada, inteligencia artificial, redes convolucionales, capas convolucionales, las funciones de activación usadas, capas de max pooling y *average pooling*, estructuras conocidas de modelos convolucionales como la *ResNet*, estructuras de modelos de segmentación como la *UNet*, modelos de atención usando *Transformers*, transferencia de conocimiento y métricas de evaluación de los modelos.

2.1. Base de datos CHAOS

La base de datos "*Combined (CT-MR) Healthy Abdominal Organ Segmentation Challenge data"* (CHAOS) consta de dos conjuntos de imágenes: imágenes de resonancia magnética (MR) y el otro de tomografía computarizada (CT). Estos conjuntos cuentan con una serie de imágenes DICOM [22]. Los conjuntos de datos se recopilan de forma retrospectiva y aleatoria del PACS (*Picture Archiving and Communication System*) del Hospital DEU. No hay conexión entre los conjuntos de datos obtenidos de las bases de datos de CT y MR (es decir, se obtienen de diferentes pacientes y no se registran).

Las imágenes de CT se obtuvieron de 40 pacientes potenciales donantes de hígado, que tienen un hígado sano (sin tumores, lesiones o cualquier otra enfermedad). Las imágenes se adquirieron del área del abdomen superior de los pacientes en la fase venosa portal después de la inyección del agente de contraste. La orientación y alineación del paciente es la misma para todos los conjuntos de datos.

La resolución de las imágenes es de tamaño 512×512, espaciado *x-y* entre 0,7 y 0,8 mm y con una distancia entre cortes (ISD) de 3 a 3,2 mm. En total, se proporcionarán 1367 cortes para entrenamiento y 1408 cortes para pruebas.

Las imágenes de MR incluyen 120 conjuntos de datos DICOM de dos secuencias de reso-

nancia magnética diferentes [T1-DUAL en fase (40 conjuntos de datos), fuera de fase (40 conjuntos de datos) y T2-SPIR (40 conjuntos de datos)], cada uno de los cuales se realiza de forma rutinaria para escanear el abdomen usando diferentes combinaciones de pulsos y gradientes de radiofrecuencia.

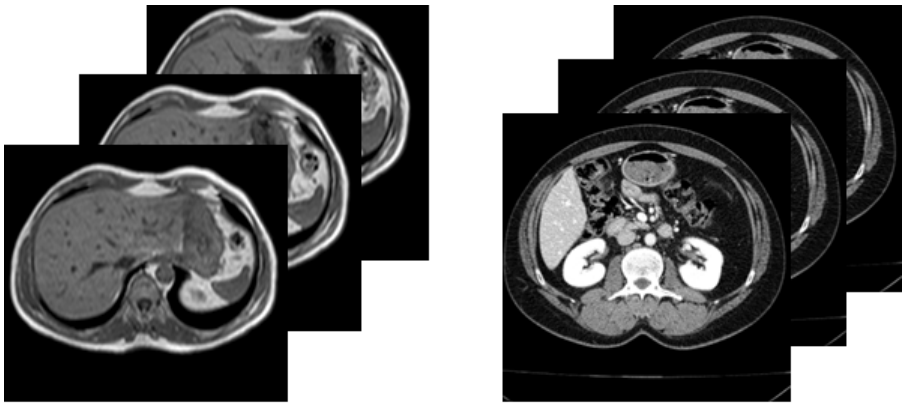


Figura 2.1: (a) son ejemplos de imágenes de resonancias magnéticas (RM) y (b) contiene ejemplos de imágenes de tomografías computarizadas.

Esta base de datos no incluye ningún tumor o lesión en los bordes de los órganos de interés anotados (es decir, hígado, riñones, bazo). Los conjuntos de datos se adquieren mediante una resonancia magnética Philips de 1,5 T, que produce imágenes DICOM de 12 bits con una resolución de 256×256. Los ISD (*inter-slice distance*) varían entre 5,5 y 9 mm (promedio de 7,84 mm), el espacio *x-y* está entre 1,36 y 1,89 mm. En total, se proporcionarán 1594 cortes para entrenamiento y 1537 cortes se utilizarán para las pruebas [22].

2.2. Campos de la inteligencia artificial

2.2.1. *Aprendizaje máquina*

El aprendizaje máquina o *machine learning* (ML) es un campo dentro de la inteligencia artificial (AI) donde se entrena a las máquinas para reconocer patrones basados en datos y poder realizar tareas de clasificación o predicción. Para lograr este entrenamiento se requieren de datos

para tener conclusiones a partir de los mismos, es decir, aprender. Estos algoritmos se ajustan a medida que se procesa la información y conoce el entorno en que está aprendiendo [23].

El ML se conforma de distintos tipos de aprendizaje. El aprendizaje supervisado es cuando un algoritmo aprende a partir de los datos etiquetados y categorizados que se proporcionan por los seres humanos. Más adelante, las máquinas son capaces de generalizar y clasificar automáticamente, sin intervención humana, mediante el etiquetado previo. Con este aprendizaje, se extraen reglas que se van actualizando y ayudan a tomar decisiones [24].

Para el aprendizaje no supervisado en ML, no se requiere que los datos cuenten con el etiquetado, ya que el objetivo es encontrar relaciones entre los datos, agrupándolos mediante conclusiones que se obtienen de los datos no estructurados. De esta manera, la máquina es capaz de detectar similitudes, diferencias o anomalías en los datos [25].

2.2.2. *Aprendizaje profundo*

El aprendizaje profundo (DL, por sus siglas en inglés *Deep learning*) es un subconjunto de ML en el que la máquina es capaz de razonar y sacar sus propias conclusiones, aprendiendo por sí misma con redes profundas multicapa con grandes cantidades de datos.

La diferencia clave reside en que los humanos no tienen que enseñar al programa, como es, por ejemplo, un gato o un perro, simplemente con darle los suficientes datos, será capaz de resolver el problema por sí mismo. Estos métodos han mejorado el estado del arte en el reconocimiento de voz, reconocimiento de objetos visuales, detección de objetos y muchos otros dominios [26].

DL descubre estructuras entre los grandes conjuntos de datos mediante algoritmos de propagación hacia atrás, indicando como la máquina debe cambiar sus parámetros internos para calcular la representación en cada capa a partir de la capa anterior [27].

2.2.3. *Visión por computadora*

La visión por computadora es una disciplina dentro de la robótica e informática, en el cual sus avances han permitido emplear soluciones inteligentes, dinámicas y versátiles [28]. Se hace referencia a un grupo de tecnología que permite a los equipos de cómputo captar imágenes, procesarlas y generar información, es decir, realizar un análisis de las mismas.

En esta disciplina, se aborda el conjunto de la teoría, el diseño e implementación de algoritmos para procesar datos visuales (imágenes o video), para reconocer y detectar objetos, además de analizar y construir la forma y diseño espacial. De aquí se extienden 4 problemas principales de la visión por computadora: clasificación de imágenes, detección de objetos, segmentación de instancias y segmentación semántica

La visión por computadora tiene como objetivo comprender escenas del mundo real a través del análisis de imágenes digitales. Este campo se emplea en una gran cantidad de aplicaciones en la vida real como: los vehículos autónomos, el reconocimiento facial, cuidado de la salud, seguimiento deportivo en tiempo real, en la agricultura, líneas de producción, modelado 3D, captura de movimiento, entre otros [28].

Segmentación de imágenes médicas

La segmentación de imágenes es un problema general en el campo de la visión artificial. Consiste en dividir una imagen digital en regiones o grupos de píxeles. Estas regiones se clasifican, es decir, a cada píxel se le asigna una categoría de la imagen que se está procesando. El reconocimiento de los elementos de las imágenes es una tarea de gran complejidad, que regularmente muestra mejores resultados al ser realizada por los humanos. Sin embargo, esta tarea requiere gran precisión, que se obtiene de mejor manera por medios computacionales. Por lo que, una buena segmentación dará lugar a buenos resultados dependiendo la aplicación en la que se aplique [29].

2.2.4. *Transferencia de conocimiento*

La transferencia de conocimiento o el aprendizaje por transferencia (TL) es un problema de investigación en *Machine Learning* (ML) que se enfoca en almacenar el conocimiento obtenido al resolver un problema y aplicarlo a un problema diferente con características similares [15].

Dentro de las ANN, esto puede ser benéfico, de manera que se puede aprovechar el conocimiento de haber entrenado redes robustas en equipos de cómputo con muchos recursos, solamente utilizando los pesos que se usaron para el entrenamiento y aprovecharlos para la validación o prueba de los datos que nos interesa clasificar o segmentar.

La transferencia de aprendizaje es una técnica efectiva para entrenar redes neuronales grandes cuando tienes pocos datos de entrenamiento, evitando el problema de *overfitting*. Se ha comprobado que, especialmente en el caso de las redes neuronales convolucionales profundas (CNN), transferir conocimientos previos, incluso de tareas diferentes, supera el uso de pesos de red aleatorios. Al utilizar conjuntos de datos previamente entrenados, como ImageNet, los resultados de la transferencia demuestran un mejor rendimiento en comparación con otros enfoques estándar en diversos conjuntos de datos [30].

2.3. Redes neuronales

En esta sección, se describen las redes neuronales convolucionales, redes neuronales residuales, capas de *pooling*, los modelos *UNet*, *ResNet-UNet*, mecanismos de atención, el modelo *Trans-UNet*, teoría de lógica difusa, métricas para evaluar los modelos de segmentación, funciones de perdida y pruebas estadísticas.

2.3.1. *Capas de convolución*

En una capa convolucional, múltiples filtros se deslizan sobre la capa de los datos de entrada. La salida de esta capa, será una suma de una multiplicación elemento por elemento de los filtros y la capa de los datos [31]. El resultado será un elemento de la siguiente capa (capa de salida). La Figura 2.2 muestra el primer paso de una capa convolucional, aplicando un filtro de 3×3.

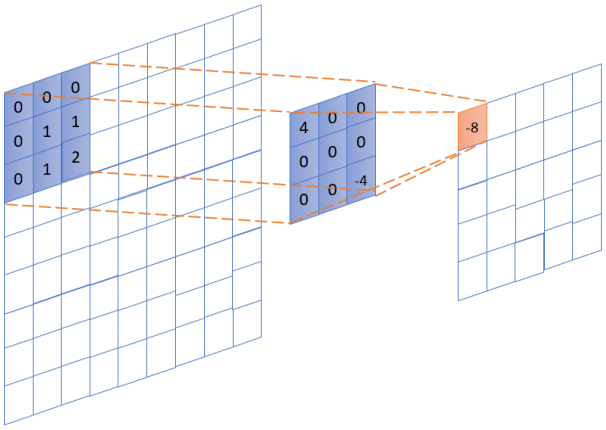


Figura 2.2: La capa de convolución desliza el filtro sobre una entrada, y la salida es la suma de un elemento por la multiplicación de la matriz del elemento del filtro.

Una operación de convolución se define como:

$$V = \| \sum_{i=1}^{q} (\sum_{j=1}^{q} f_{ij} d_{ij}) \| \tag{2.1}$$

Donde f_{ij} es el coeficiente del kernel de convolución en la posición i, j (en el kernel), d_{ij} es el valor del pixel que corresponde con f_{ij}, q es la dimensión del kernel, asumiendo que el kernel es un cuadrado (si $q = 3$, el kernel es de 3×3) y V el valor del pixel de salida [32].

Las operaciones realizadas para calcular el primer elemento de la Figura 1 son:

$$(4 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 1) + (0 \times 1) + (0 \times 0) + (0 \times 1) + (-4 \times 2) = -8 \tag{2.2}$$

Cada operación convolucional se especifica por el paso que da el filtro sobre la entrada (stride), el tamaño del filtro y el relleno de ceros (padding). En el caso de la Figura 2.2, se tiene un $stride = 1$, filtro de tamaño 3×3 y se rellena con ceros la orilla de la capa de entrada. El rellenado de ceros agrega filas y columnas a la matriz de entrada original para controlar el tamaño del mapa de características de salida [33].

2.3.2. *Capa de agrupación máxima*

Una capa de Max-pooling reduce la dimensionalidad de la entrada por un factor constante. Esto no solo es para reducir la carga computacional, sino también realizar la selección de características [33]. Las imágenes de entrada se organizan de forma que no se superponen y solo se extrae un valor de salida. Las opciones más comunes son *Max-pooling* (el valor máximo) y el Avg-pooling (el promedio). El Max-pooling es favorable, ya que introduce una pequeña invarianza en la distorsión, además conduce a una convergencia más rápida y una mejor generalización [34]. Un ejemplo de la primera operación de una capa de *Max-pooling* se muestra en la Figura 2.3.

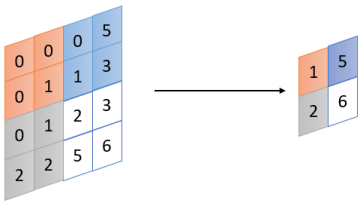


Figura 2.3: Max-pooling con filtro de 2x2 y un stride= 2.

2.3.3. *Redes neuronales convolucionales*

En el campo del aprendizaje profundo (DL) destacan las redes neuronales convolucionales (CNN), conocidas en inglés como *Convolutional Neural Networks*. Estas redes han revolucionado la visión por computadora y la percepción de imágenes gracias a su capacidad para aprender automáticamente características de las imágenes. Las CNN utiliza un enfoque de aprendizaje supervisado, donde se utilizan etiquetas o datos de entrenamiento para ajustar los pesos de la red y aprender a realizar tareas específicas [35].

La arquitectura VGG [36] se destacó por su simplicidad y profundidad, lo que la convirtió en un referente en el campo de las CNN. Esta arquitectura demostró que las redes más profundas podían aprender características más abstractas y complejas de las imágenes, mejorando significativamente el rendimiento en tareas de clasificación de imágenes.

Otra innovación importante en el campo de las CNN es la introducción de redes totalmente convolucionales (FCN), que permiten la segmentación semántica de imágenes. A diferencia de las redes tradicionales, las FCN utilizan capas convolucionales en lugar de capas completamente conectadas en la etapa de decodificación, lo que las hace adecuadas para tareas de segmentación de imágenes al retener información en el espacio [37].

El modelo U-Net es otro avance relevante en este contexto. Este modelo está diseñado específicamente para tareas de segmentación de imágenes biomédicas y presenta una arquitectura en forma de U que captura detalles a nivel de píxeles al tiempo que preserva la información espacial [7].

Modelo VGG

Una red neuronal VGG es una arquitectura de red neuronal convolucional (CNN) utilizada para tareas de visión por computadora, como el reconocimiento de imágenes [36]. La red neuronal VGG fue desarrollada por un equipo de investigadores en el Visual Geometry Group (VGG) en la Universidad de Oxford. La arquitectura más conocida, consta de 16 capas convolucionales y de agrupación, seguidas por varias capas totalmente conectadas.

Las diferencias entre la arquitectura VGG con otras CNN es que utilizan filtros de tamaño muy pequeño (3×3) en todas las capas convolucionales, lo que permite a la red aprender características más detalladas de las imágenes. Además, la VGG es capaz de aprender patrones complejos en las imágenes gracias a su profundidad, que puede alcanzar hasta 19 capas. Antes del modelo VGG, los modelos utilizaban capas de convolución de dimensión 5 o incluso 11 [38] [39].

La arquitectura de este modelo se muestra en la Figura 2.4. Teniendo 2 capas de convolución, seguida de 1 de *max-pooling*, este bloque se repite, y luego vienen 3 capas de convolución, seguidas de otra capa de *max-pooling*, este bloque se repite 3 veces, con lo que se tienen 13 capas de convolución. Al final tenemos 2 capas totalmente conectadas y una capa con la función *softmax* [40].

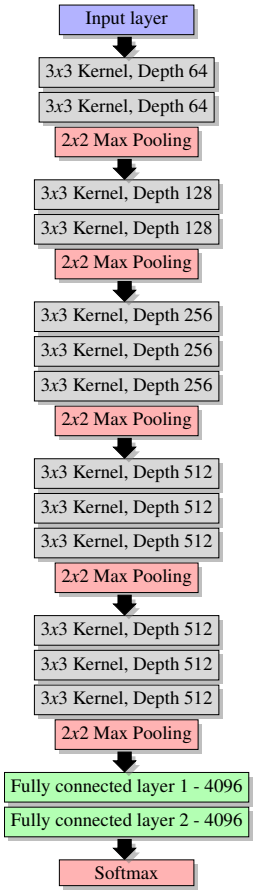


Figura 2.4: Arquitectura del modelo VGG.

Como se puede observar, la red VGG cuenca con una estructura fácil de comprender con pocas capas convolucionales. La red neuronal VGG se ha utilizado en la tarea de segmentación semántica mediante la adición de una capa de decodificación después de las capas convolucionales. La capa de decodificación toma las características aprendidas por la red convolucional y las utiliza para producir una máscara de segmentación de píxeles de la imagen de entrada [41].

Una forma común de realizar la segmentación semántica con la arquitectura VGG es utilizar una estructura de red *Fully Convolutional Network* (FCN) o el modelo *UNet*. Eliminando las capas totalmente conectadas y agregando capas de deconvolución.

Modelo *Fully Convolutional Network* (FCN)

La estructura de red FCN (*Fully Convolutional Network*) es una arquitectura de red neuronal convolucional que se utiliza para tareas de segmentación semántica. Comienza con una etapa de codificación en la que se tienen capas convolucionales (por ejemplo, VGG-16) en donde se extraen las características de la imagen de entrada. Al final de la última capa de convolución, se agrega una capa de conexión para fusionar las características extraídas en la etapa de codificación con la información de alta resolución de las capas de deconvolución [42] [43].

Para la decodificación se tiene una serie de capas de convolución transpuesta (deconvolución) que aumentan la resolución de las características extraídas en la etapa de codificación. Cada capa de convolución transpuesta se utiliza para deshacer la reducción de tamaño realizada por las capas de pooling en la etapa de codificación. Por último, la capa de salida es una capa de convolución que produce una matriz de píxeles que representa la máscara de segmentación. La máscara de segmentación tiene la misma resolución que la imagen de entrada y asigna una etiqueta a cada píxel de la imagen [44] [**?**].

La ventaja de utilizar un modelo FCN es que puede proporcionar una etiqueta semántica precisa para cada píxel en la imagen, lo que puede ser útil en aplicaciones como la detección de objetos y la identificación de regiones de interés. Además, al utilizar una arquitectura de red neuronal convolucional, el modelo FCN puede aprender características de alto nivel de la imagen de entrada, lo que puede mejorar la precisión de la segmentación semántica [37] [45].

Modelo de redes neuronales residuales

Las redes neuronales residuales (*ResNet*) son un tipo de redes neuronales artificiales (ANN). Este tipo de redes utilizan saltos de conexiones (*skip connections*), para saltar sobre algunas capas, como se muestra en la Figura 2.5. Estos saltos se dan sobre capas que no tienen operaciones lineales como las ReLU o de normalización [46].

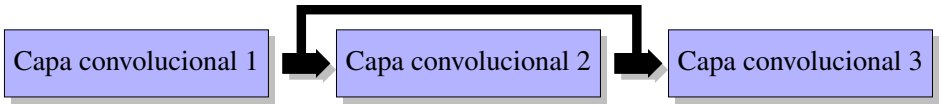


Figura 2.5: Forma simple de una ResNet. La capa 2 es salteada desde la capa 1.

Una de las principales razones para agregar las skip connections es para evitar el problema de la desaparición del gradiente (*vanishing gradients*) [47]. La desaparición del gradiente sucede al momento de tener un gran número de capas, lo que reduce el gradiente calculado, por lo que se tendrá un mayor error de entrenamiento. Al tener *skip connections* se evita que el gradiente se desvanezca, y la red puede aprender del espacio de características [48].

Modelo *UNet*

El modelo U-Net es una arquitectura de red neuronal convolucional (CNN) que se desarrolló para la segmentación de imágenes biomédicas [7]. Esta red es efectiva para problemas en los que la salida es de tamaño similar a la entrada. Esto hace que sean especialmente buenas en el procesamiento o generación de imágenes con buena resolución.

Para una arquitectura simple, se tienen 3 bloques: *Encoder, Decoder,* y *BottleNeck.* En el bloque *Encoder*, se realiza un proceso convolucional: se incrementarán los canales y se decrementarán las dimensiones espaciales del vector de entrada. Para el bloque *Decoder*, se usa un *UpSampling* para ajustar las dimensiones de los vectores de entrada, y que estos sean concatenables con las imágenes de la misma dimensionalidad. El bloque *Bottleneck* usa un procesamiento extra que implicará estimar características más profundas en el vector. De manera simple, un modelo U-Net se puede ejemplificar de la siguiente manera en la Figura 2.6.

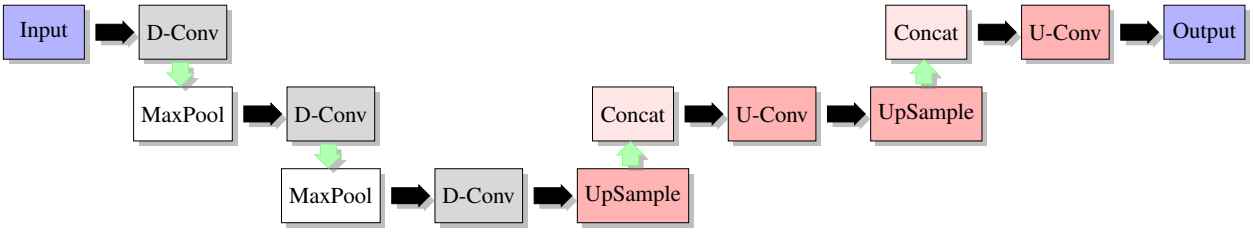


Figura 2.6: Ejemplo de la arquitectura de una U-Net.

Modelo *ResNet-UNet*

El modelo ResNet-UNet consta de una arquitectura *encoder-decoder* en donde se incluyen saltos de conexiones que pueden aumentar la profundidad y mejorar la precisión de las CNN profundas evitando problemas de *overfitting* [49].

Al usar únicamente una arquitectura de U-Net, las predicciones pueden carecer de detalles finos, para resolver este problema se pueden agregar las *skip connections* con un *backbone* de la ResNet. La diferencia con las *skip connections* de la ResNet, es que estos saltos se hacen con las matrices de igual dimensión. Esto permite que los detalles finos de la imagen de entrada estén en la parte superior de la U-Net, con la entrada asignada casi directamente a la salida [50].

2.4. Mecanismos de atención

Los mecanismos de atención, están inspirados en el comportamiento humano, donde el cerebro se centra en una parte de un texto, imagen o cualquier percepción [16] y le da menos importancia a otras. Con esto, un modelo de atención en una red neuronal permite asignar diferentes pesos a cada estado en la red, por lo que al realizar la suma ponderada de cada estado permite que la salida se componga de diferentes estados y no únicamente del estado anterior [17]. En una red, esto permite que la atención se utilice para fusionar características y con la información total obtenida obtener un resultado parecido al que obtendría un humano. Este tipo de modelos se han utilizado en una variedad de tareas, como procesamiento de lenguaje natural, subtítulos de imágenes y traducción automática.

La entrada para el módulo de atención consta de traducir la entrada (palabras, texto, imágenes) a números mediante incrustaciones (*embedding*), y aplicarle una codificación posicional, que nos resultará en un vector o vectores con información posicional de los datos (información con contexto) [51].

El módulo de atención normalmente consta de tres partes: una *query*, una *key* y un *value*, como se puede ver en la Figura 2.7. La *query* se usa para determinar en qué partes de la entrada

debe enfocarse el modelo, la *key* se usa para determinar qué partes de la entrada coinciden con la *query* y el *value* es la información que el modelo debe extraer de la entrada.

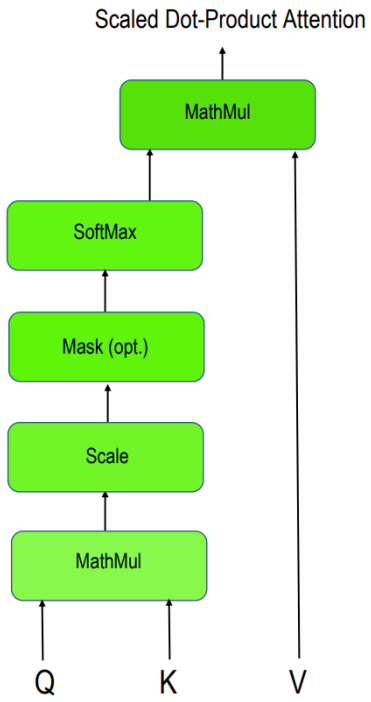


Figura 2.7: Atención de producto punto escalado.

Hay varios tipos de mecanismos de atención, como la atención de producto punto, la atención aditiva y la atención de múltiples cabezas (*multi-head attention*). La atención de producto escalar es la forma más básica de atención, donde el producto escalar del *query* y la *key* se usa para calcular las puntuaciones de atención. La atención aditiva utiliza una red neuronal de avance para calcular las puntuaciones de atención. La atención de múltiples cabezas, como se ve en la Figura 2.8, es una extensión de la atención de producto punto en la que se utilizan varios conjuntos de *query*, *key* y *value* para calcular múltiples puntuaciones de atención, que luego se concatenan y se utilizan para ponderar los valores [52].

Uno de los módulos de atención más populares se llama *transformer self-attention*, se basa en un mecanismo de auto atención que se utiliza para sopesar la importancia de las diferentes partes de la entrada antes de pasarla a la siguiente capa, esto permite que el modelo se centre en

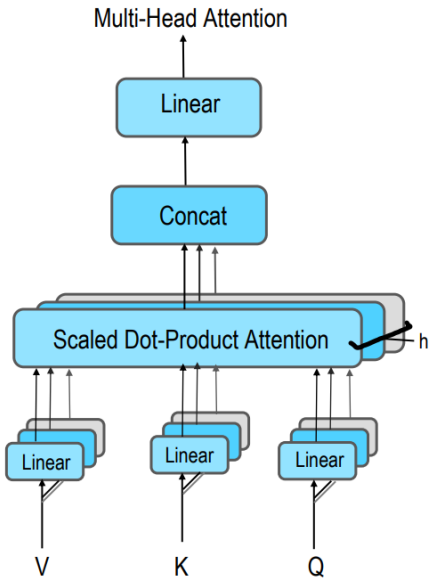


Figura 2.8: Atención de múltiples cabezas (*multi-head attention*).

partes específicas de la entrada, en lugar de utilizar toda la entrada por igual. Se utiliza en modelos basados en *transformer* como BERT, GPT-2, T5 y se ha demostrado que es eficaz en varias tareas de *Natural Language Processing* (NLP).

2.4.1. *Transformer self-attention (TSA)*

Transformer self-attention es un tipo de arquitectura de red neuronal que se presentó en el documento *Attention is all you need* por los investigadores de Google en 2017 [51]. Se utiliza principalmente para tareas de procesamiento de lenguaje natural, como traducción de idiomas, resumen de texto y respuesta a preguntas.

La arquitectura del *transformer* se compone de dos partes clave: el codificador y el decodificador. En el proceso, el codificador recibe una secuencia de entrada y crea una serie de representaciones intermedias, conocidas como *query*, *key* y *value*. Estas representaciones son esenciales para el mecanismo de atención, ya que permiten evaluar y asignar importancia a las diferentes partes de la secuencia de entrada. Luego, el decodificador utiliza esta información ponderada para generar la secuencia de salida deseada.

Una de las ventajas clave de la arquitectura del *transformer* es su capacidad para manejar secuencias de entrada de diferentes longitudes. Además, los modelos de *transformer* se pueden entrenar en paralelo, lo que permite tiempos de entrenamiento más rápidos.

Un enfoque para usar el *transformer* en la segmentación de imágenes es usar un conjunto de capas convolucionales para extraer características de la imagen, luego usar un codificador del *transformer* para procesar las características y generar los mapas de atención, que luego se pasan a una red decodificadora para generar al final la máscara de segmentación [18].

Otro enfoque es usar el *transformer* directamente en los píxeles de la imagen [19]. Se realiza aplanando la imagen en una secuencia 1D de píxeles y luego aplicando la arquitectura del *transformer* sobre ella. Este enfoque es más costoso desde el punto de vista computacional, pero permite utilizar la capacidad total del *transformer* para capturar las dependencias globales entre píxeles.

En ambos enfoques, la diferencia clave entre usar el *transformer* para NLP y la segmentación de imágenes es la forma de los datos de entrada, mientras que las tareas de NLP toman como entrada, secuencias de palabras 1D, las tareas de segmentación de imágenes toman como entrada, matrices de píxeles 2D. Cabe señalar que el uso de *transformer* en la segmentación de imágenes es un área activa de investigación y se están proponiendo varios enfoques y arquitecturas para mejorar el rendimiento de estos modelos [53].

La estructura de un *transformer* con mecanismo de atención (*self-attention*) se compone de varias capas, cada una de las cuales tiene una estructura similar. Cada capa se compone de dos sub capas: una capa de atención multi-cabeza y una capa de alimentación hacia adelante (*Feed-forward*). Además, se utilizan técnicas de normalización y regularización para mejorar el rendimiento del modelo. En la entrada, la secuencia que se proporciona al *transformer* se concatena con un *positional embedding* el cual proporciona información para que la red pueda comprender la posición relativa de cada secuencia.

La capa *multi-head attention* se utiliza para calcular la matriz de atención ponderada. Cada cabeza realiza la operación de atención, y el resultado de cada cabeza de atención, se concatena y aplica una transformación lineal para obtener la salida de la capa. Para la capa *Feed-forward*, toma como entrada, la salida de *multi-head attention*, y se le aplica una función de activación no lineal, como la función *ReLU*. Al final de las capas de *multi-head attention* y *Feed-forward*, se coloca una capa de normalización para normalizar la salida de cada capa y mejorar la estabilidad del modelo. Un ejemplo de la estructura mencionada se ilustra en la Figura 2.9.

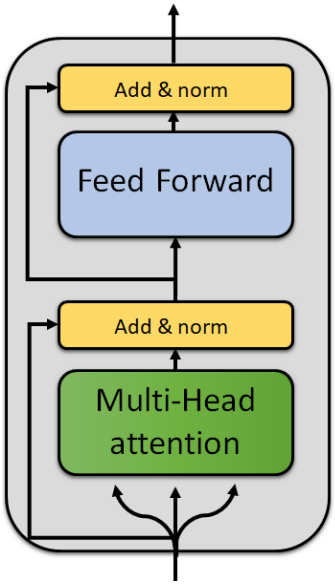


Figura 2.9: Estructura básica del *transformer self-attention*.

El *transformer* se entrena utilizando un proceso llamado entrenamiento por pares, en el que el modelo aprende a predecir la probabilidad de una etiqueta dada la posición en la imagen, y con el contexto de su vecindario. El TSA genera vectores de atención para cada palabra (o píxel) de la entrada para representar cuanto se relaciona con cada palabra (o píxel) de la misma oración.

2.4.2. *Transformer cross-attention (TCA)*

En un *transformer cross-attention* (TCA) la entrada se divide en 2 secuencias: una secuencia de consulta y una secuencia de contexto. En la capa de atención se da la importancia de cada elemento en la secuencia de consulta, en función de la relación con la secuencia de contexto. Por lo general, la secuencia de consulta es una secuencia de entrada que se está procesando, mientras que la secuencia de contexto es una secuencia de referencia que se utiliza para ayudar en el procesamiento de la secuencia de consulta [54] [55].

Al realizar el cálculo de *cross-attention* se obtiene una matriz que representa la importancia relativa de cada elemento en la secuencia de consulta en función a la relación con cada elemento de la secuencia de contexto. Esto permite modelar la relación entre diferentes partes de una oración en caso de procesamiento de lenguaje natural (por ejemplo, una palabra en una oración con el contexto de la misma oración), o en diferentes partes de una imagen respecto al resto de la imagen.

El *transformer cross-attention* cuenta con dos módulos similares que el TSA, pero antes se le agrega un módulo de atención con múltiples cabezas enmascarado. Por este último módulo pasa la información de la imagen o texto de entrada con su información posicional, para obtener el *Query*, lo cual es el contexto de este vector, mientras que los valores de *key* y *value*, es la información más relevante del vector que se obtiene de un TSA 2.10.

2.4.3. *Deformable self-attention (DSA)*

El módulo de atención *Deformable self-attention* atienden a un pequeño conjunto de puntos de muestreo clave alrededor de un punto de referencia. Los módulos TSA tienen una complejidad computacional cuadrática, mientras que el DSA puede reducir la complejidad a $nlog(n)$. Esto ayuda a que el rendimiento del módulo sea mejor, con 10 veces menos épocas de entrenamiento. El módulo DSA cuenta con 8 cabezas de atención [56].

La convolución deformable funciona de formas eficaz y eficiente en el reconocimiento de

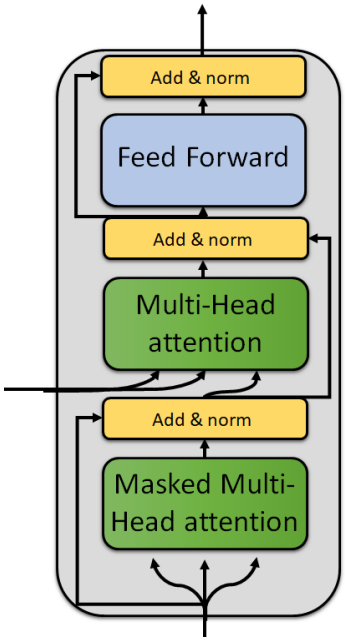


Figura 2.10: Estructura básica del *transformer cross-attention*.

imágenes que el *transformer self-attention*. El DSA se inspira directamente de las convoluciones deformables y se enfoca especialmente en un pequeño conjunto fijo de puntos de muestreo predichos a partir de la característica de los elementos del *query*. El DSA aprovecha la arquitectura de un *transformer encoder-decoder* para transformar los mapas de características de entrada en características de un conjunto de *query*, se le agrega una red FFN de 3 capas y una capa lineal para el decodificador.

En el codificador del DSA, el *query* y *key* son los píxeles en los mapas de características. Estos mapas se obtienen de una red *ResNet* (con *positional embeddings* codificados). Mientras, para el decodificador, la entrada incluye los mapas obtenidos por el codificador y la matriz *query* representan los *positional embeddings* aprendibles.

2.4.4. *Modelo Trans-UNet*

Los modelos de *Convolutional Neural Networks* (CNN) han sido ampliamente utilizados en el procesamiento de imágenes debido a su capacidad para capturar patrones locales y carac-

terísticas específicas en los datos visuales. Sin embargo, estas redes tienen ciertas limitaciones, especialmente cuando se trata de modelar relaciones de largo alcance entre elementos en una imagen. Esta debilidad se vuelve evidente en aplicaciones médicas, donde las estructuras objetivo pueden variar significativamente en términos de textura, forma y tamaño entre pacientes.

Para abordar estas limitaciones, se han propuesto modelos que incorporan mecanismos de atención basados en CNN. Estos mecanismos de atención permiten a la red sopesar la importancia de diferentes partes de la entrada, lo que resulta en una mejor captura de relaciones de largo alcance y una representación más rica de las características.

Por otra parte, los modelos fundamentados en la arquitectura Transformer han emergido como herramientas altamente competentes en la captura de relaciones y contextos globales en secuencias de datos, como es el caso del texto. Es importante destacar que estos modelos han trascendido su ámbito original y, en investigaciones recientes, se ha demostrado que los Transformers no solo pueden igualar, sino también superar, a otros enfoques en tareas de reconocimiento de imágenes [19].

El modelo *TransUNet*, da mecanismos de auto-atención desde la perspectiva secuencia-secuencia, mientras añade la arquitectura de una red UNet, para aprovechar la información espacial detallada de alta resolución y aprovecha las ventajas de los *transformers* para el análisis del contexto global [57].

2.4.5. *Modelo MCTrans*

Los *transformers* usados para procesar imágenes, aprenden de la interacción no local de diferentes *tokens* en cada parche, desafortunadamente estas técnicas pierden tanto lo aprendido entre diferentes escalas de píxeles como la correspondencia semántica entre etiquetas, todo esto fundamental en la segmentación de imágenes médicas.

El modelo *MCTrans* incorpora, además de las capas de convolución, los *transformers*:

Transformer Self-Attention (TSA) y *Transformer Cross-Atenttion* (TCA). El módulo TSA tiene como objetivo lograr un modelado contextual en un nivel de pixel a escala cruzada mediante los mecanismos de autoatención, dando características más completas para diferentes escalas de pixel. Mientras, el módulo TCA ayuda al modelo a aprender automáticamente la correspondencia semántica de diferentes categorías, introduciendo un *Proxy embedding*, ayudando a interactuar con las representaciones de características del módulo [58]. La arquitectura del modelo se pueden ver en la Figura 2.11.

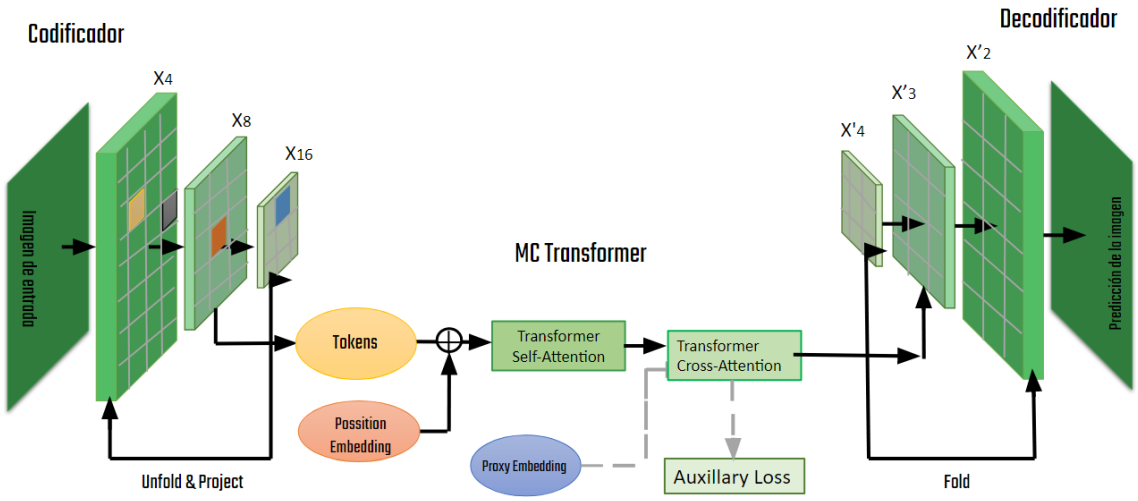


Figura 2.11: Diagrama a bloques del modelo MCTrans.

2.5. Teoría de lógica difusa

La teoría de lógica difusa o teoría *fuzzy logic* es una rama de la inteligencia artificial que trata con la lógica difusa, una forma de lógica que permite el razonamiento con conceptos imprecisos o vagos. La teoría difusa se basa en la idea de que los valores pueden tener un grado de pertenencia a un conjunto o categoría en lugar de ser solo verdaderos o falsos. Por ejemplo, si preguntamos ”¿Una persona es alta?”, la respuesta podría ser ”sí, es alta, pero no muy alta”. La teoría difusa permite capturar este tipo de información y manejarla en cálculos y decisiones [59].

Los conjuntos difusos son una parte fundamental de la teoría difusa. A diferencia de los

conjuntos tradicionales, en los que cada elemento pertenece o no a un conjunto, en los conjuntos difusos cada elemento tiene un grado de pertenencia, que puede variar entre 0 y 1, indicando el grado de similitud del elemento con el conjunto.

Los conjuntos difusos se definen mediante funciones de membresía que asignan a cada valor un grado de pertenencia a un conjunto. Por ejemplo, considere el conjunto difuso de ”personas altas”. En lugar de tener una definición precisa, como ”personas con una altura mayor a $1,80$ metros”, el conjunto difuso de ”personas altas”podría tener una función de membresía que asigne a cada persona un grado de pertenencia al conjunto. Si alguien tiene una altura de $1,85$ metros, su grado de pertenencia al conjunto ”personas altas”podría ser del 0,9, mientras que alguien con una altura de $1,70$ metros podría tener un grado de pertenencia de 0,2.

La función de membresía es una curva que describe la relación entre la variable de entrada (en el ejemplo anterior, la altura de una persona) y el grado de pertenencia al conjunto difuso. Existen diferentes tipos de funciones de membresía, como la función triangular, la función trapezoidal o la función gaussiana, que se ajustan a diferentes situaciones y contextos [60].

2.5.1. *Funciones de membresía*

Función de membresía triangular

La función de membresía triangular es una de las funciones de membresía más comunes en la teoría difusa. Su forma se asemeja a un triángulo y se utiliza para definir conjuntos difusos que tienen un valor mínimo, un valor máximo y un valor medio [61] [62].

La función de membresía triangular se define matemáticamente como:

$$\mu_A(x) = \begin{cases} 0 & \text{si } x \leq a \\ \frac{x-a}{m-a} & \text{si } a < x \leq m \\ \frac{b-x}{b-m} & \text{si } m < x \leq b \\ 0 & \text{si } x > b \end{cases} \tag{2.3}$$

donde a, m y b son los parámetros que definen la posición de los puntos mínimos, medios y máximos de la función, respectivamente.

La Figura triangular formada por la función de membresía tiene una base que se extiende desde a hasta b y una altura que alcanza el valor 1 en m, como muestra la Figura 2.12. El grado de pertenencia de un elemento x a un conjunto difuso definido por una función de membresía triangular se determina por la posición de x con respecto a los puntos a, m y b [63] [64] [65] [66].

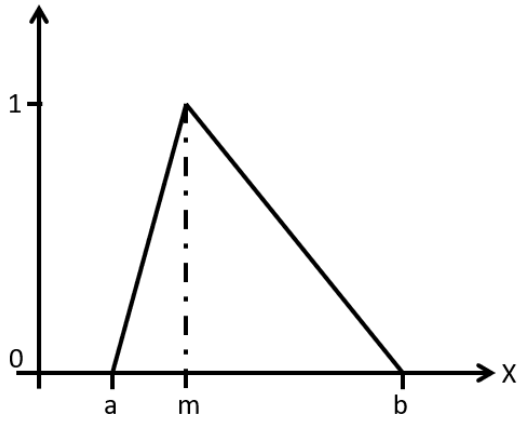


Figura 2.12: Función de membresía triangular.

Función de membresía trapezoidal

La función de membresía trapezoidal es otra función de membresía común en la teoría difusa, y se utiliza para definir conjuntos difusos que tienen un valor mínimo, un valor máximo y un valor medio, pero con una base más ancha que la función triangular [67] [61].

La función de membresía trapezoidal se define matemáticamente como:

$$\mu_A(x) = \begin{cases} 0 & \text{si } x \leq a \\ \frac{x-a}{b-a} & \text{si } a < x \leq b \\ 1 & \text{si } b < x \leq c \\ \frac{d-x}{d-c} & \text{si } c < x \leq d \\ 0 & \text{si } x > d \end{cases} \tag{2.4}$$

donde a, b, c, y d son los parámetros que definen la posición de los puntos mínimos, medio y máximos de la función, y los puntos de transición entre los valores mínimo y máximo.

La Figura trapezoidal formada por la función de membresía tiene una base que se extiende desde a hasta d, y una altura que alcanza el valor 1 en el rango b y c [63] [64]. El grado de pertenencia de un elemento x a un conjunto difuso definido por una función de membresía trapezoidal se determina por la posición de x con respecto a los puntos a, b, c y d, como se muestra en la Figura 2.13 [65].

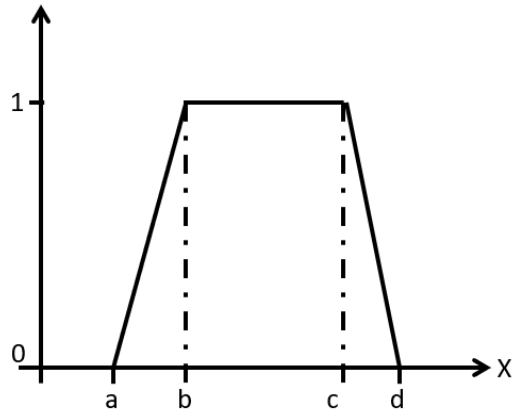


Figura 2.13: Función de membresía trapezoidal.

Función de membresía Gaussiana

La función de membresía gaussiana es una función de forma de campana que se utiliza para definir conjuntos difusos que tienen un valor medio y una dispersión en torno a ese valor medio.

La función de membresía gaussiana se define matemáticamente como:

$$\mu_A(x) = e^{-\frac{(x-m)^2}{\sigma^2}} \tag{2.5}$$

donde m es el valor medio y σ es la desviación estándar, que controla la anchura de la campana.

La forma de la función de membresía gaussiana es una campana simétrica alrededor del valor medio m [63]. El grado de pertenencia de un elemento x a un conjunto difuso definido por una función de membresía gaussiana se determina por la distancia entre x y el valor medio m en unidades de desviación estándar σ, la Figura 2.14 muestra un ejemplo de la función de membresía Gaussiana [65].

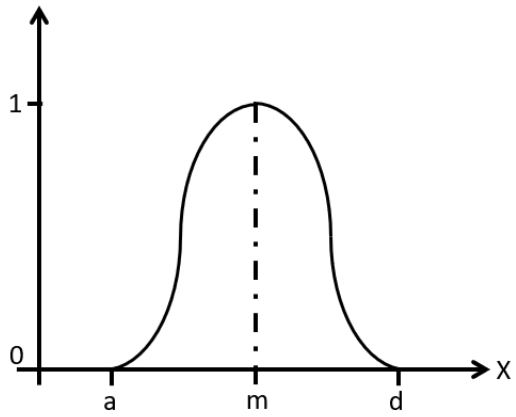


Figura 2.14: Función de membresía Gaussiana.

Función de membresía sigmoidal

La función de membresía sigmoidal es una función de forma S que se utiliza para definir conjuntos difusos que tienen un valor de transición suave entre los valores de pertenencia 0 y 1.

La función de membresía sigmoidal se define matemáticamente como:

$$\mu_A(x) = \frac{1}{1+e^{-b(x-m)}} \tag{2.6}$$

donde *m* es el punto de inflexión de la curva, y *b* controla la pendiente de la curva.

La forma de la función de membresía sigmoidal es una curva S que se extiende desde 0 hasta 1, y su transición es suave alrededor del punto de inflexión *m* [64]. El grado de pertenencia de un elemento *x* a un conjunto difuso definido por una función de membresía sigmoidal se determina por la posición de *x* con respecto al punto de inflexión *m*, como se muestra en la Figura 2.15 [67].

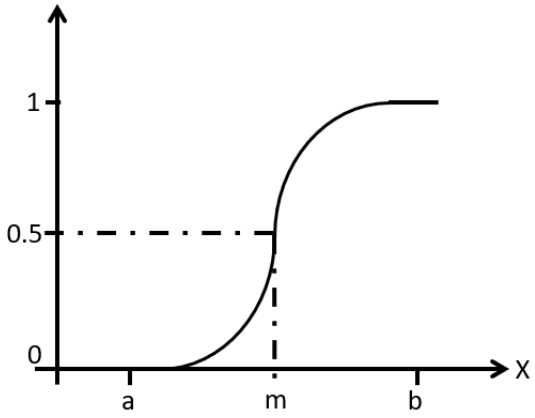


Figura 2.15: Función de membresía sigmoidal.

2.5.2. *Sistema de control difuso*

Los sistemas de control difuso (también conocidos como sistemas de control fuzzy) son sistemas de control basados en la teoría de conjuntos difusos o fuzzy. A diferencia de los sistemas de control convencionales, que utilizan reglas y cálculos precisos para controlar un proceso, los sistemas de control difuso utilizan la lógica difusa para modelar el conocimiento y la experiencia del experto humano y controlar el proceso en consecuencia.

Los sistemas de control difuso constan de tres partes principales: el sistema de entrada, el sistema de inferencia y el sistema de salida, como se muestra en la Figura 2.16. El sistema de entrada convierte las entradas del proceso en valores de pertenencia a conjuntos difusos utilizando funciones de membresía. El sistema de inferencia aplica reglas difusas basadas en la lógica difusa para determinar el grado de activación de cada regla y, por lo tanto, la salida del sistema. El sistema de salida convierte los resultados difusos en una salida precisa utilizando funciones de membresía

y técnicas de defuzzyficación.

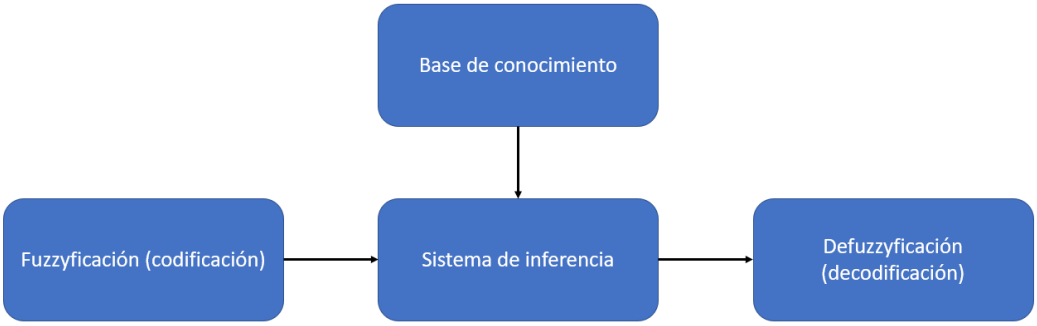


Figura 2.16: Estructura general de un sistema de control difuso.

Los sistemas de control difuso se utilizan en una amplia variedad de aplicaciones, como el control de la velocidad de un motor, la temperatura de un horno, el nivel de agua en un tanque, la calidad de la señal de audio en un sistema de comunicación, entre otros. Además, los sistemas de control difuso se han utilizado en combinación con otras técnicas de inteligencia artificial, como las redes neuronales, para mejorar su precisión y capacidad de control.

Entrada del sistema (codificación)

La entrada del sistema en un sistema de control difuso es una variable que se mide y se convierte en un valor de pertenencia a un conjunto difuso utilizando una función de membresía. La codificación de la entrada es una parte crítica del diseño del sistema de control difuso, ya que la precisión y la calidad de la codificación de la entrada afectarán directamente la precisión y la calidad de la salida del sistema.

En la codificación de la entrada, se selecciona una función de membresía apropiada para representar la variable. Las funciones de membresía más comunes incluyen la triangular, la trapezoidal, la gaussiana y la sigmoidal, que se mencionaron en el apartado 2,5,1. La selección de la función de membresía adecuada dependerá de la naturaleza de la variable de entrada y de los objetivos del sistema de control.

Además, la selección de los límites de los conjuntos difusos es otra parte importante de la codificación de la entrada. Estos límites determinan el rango de valores que se consideran para cada variable de entrada y deben elegirse cuidadosamente para garantizar que el sistema de control difuso sea efectivo.

Sistema de inferencia

El sistema de inferencia en un sistema de control difuso es el componente que utiliza las reglas difusas y el conocimiento experto para determinar la salida del sistema de control. El sistema de inferencia se compone de dos partes: el motor de inferencia y la base de reglas.

La base de reglas se compone de un conjunto de reglas que expresan el conocimiento del experto en términos de variables de entrada y salida. Cada regla tiene la forma ”Si A es X y B es Y, entonces C es Z”, donde A y B son variables de entrada, C es una variable de salida, y X, Y y Z son etiquetas de conjuntos difusos asociadas con A, B y C, respectivamente.

El motor de inferencia es el componente que utiliza las reglas de la base de reglas para determinar la salida del sistema de control. El motor de inferencia utiliza la lógica difusa para evaluar el grado de activación de cada regla, en función del grado de pertenencia de las variables de entrada a los conjuntos difusos asociados con cada antecedente de regla. Luego, combina las salidas de todas las reglas para producir una salida final para el sistema.

Existen diferentes métodos para la combinación de las salidas de las reglas, como el método de Mamdani y el método de Takagi-Sugeno-Kang (TSK).

Método de Mamdani

El método de Mamdani utiliza reglas lingüísticas para describir la relación entre las entradas difusas y la salida difusa del sistema. Estas reglas lingüísticas se expresan en términos de *IF-THEN*, donde la premisa es una conjunción de proposiciones lingüísticas sobre las entradas y la conclusión es una proposición lingüística sobre la salida.

Por ejemplo, una regla *IF-THEN* podría ser: *IF* la entrada *A* es baja Y la entrada *B* es alta, *THEN* la salida *C* es media. Cada premisa y conclusión en la regla se describe utilizando términos lingüísticos y funciones de membresía difusas.

El método de Mamdani utiliza la regla *IF-THEN* para determinar el grado de pertenencia de la salida difusa a cada una de las etiquetas lingüísticas que se utilizan para describir la salida. El grado de pertenencia se calcula a partir de las funciones de membresía de las entradas y de la premisa de la regla utilizando la operación de mínimo.

Una vez que se han determinado los grados de pertenencia para cada etiqueta lingüística de la salida, se utiliza una técnica de defuzzificación para calcular el valor numérico de la salida difusa. La técnica de defuzzificación más común utilizada en el método de Mamdani es el centroide [68].

El modelo de Mamdani puede expresarse en forma de pasos, de la siguiente manera:

- Definición de variables lingüísticas: Se definen las variables de entrada y salida del sistema, y se establecen las etiquetas lingüísticas que se utilizarán para describir las variables. Por ejemplo, para una variable de entrada ”*temperatura*”, se pueden definir etiquetas lingüísticas como ”*baja*”, ”*media*”, y ”*alta*”.
- Creación de funciones de membresía: Se definen las funciones de membresía difusas que representan la relación entre las etiquetas lingüísticas y los valores numéricos de las variables de entrada y salida. Por ejemplo, para la etiqueta lingüística ”*baja*”, de la variable de entrada ”*temperatura*”, se puede utilizar una función de membresía triangular que asigna un grado de pertenencia máximo a los valores numéricos más bajos de la variable de entrada, y un grado de pertenencia cero a los valores más altos.
- Creación de reglas lingüísticas: Se definen las reglas lingüísticas *IF-THEN* que describen la relación entre las entradas y la salida del sistema. Por ejemplo, una regla podría ser: ”*IF*

temperatura es baja *THEN* velocidad es baja". Las reglas lingüísticas se expresan utilizando las etiquetas lingüísticas y las funciones de membresía correspondientes.

- Evaluación de las reglas: Se evalúan todas las reglas lingüísticas utilizando la operación de mínimo para determinar el grado de pertenencia de la salida a cada etiqueta lingüística.
- Agregación de las salidas: Se agregan todos los grados de pertenencia de la salida a cada etiqueta lingüística para obtener una salida difusa global.
- Defuzzificación: Se utiliza una técnica de defuzzificación, como el centroide, para obtener un valor numérico de salida a partir de la salida difusa global.

El método de Mamdani es relativamente fácil de implementar y entender, y se puede utilizar para una amplia variedad de problemas de control difuso. Sin embargo, también tiene algunas limitaciones, como la necesidad de un gran número de reglas para obtener una buena precisión del sistema, y la dificultad de ajustar los parámetros de las reglas de manera precisa.

Método de Takagi-Sugeno-Kang

El método de Takagi-Sugeno-Kang (TSK) es un enfoque alternativo al método de Mamdani para la implementación de sistemas de control difuso. A diferencia del método de Mamdani, que produce una salida difusa para cada regla, el método TSK produce una salida numérica precisa para cada regla [69] [70].

El modelo TSK consta de tres partes principales:

- Base de reglas: Al igual que en el modelo Mamdani, se definen las variables de entrada y salida del sistema, y se establecen las etiquetas lingüísticas que se utilizarán para describir las variables. Luego, se definen las reglas lingüísticas *IF-THEN* que describen la relación entre las entradas y la salida del sistema. Sin embargo, en el modelo TSK, la parte "*THEN*" de las reglas especifica una función lineal que produce una salida numérica precisa en lugar de una salida difusa.

- Evaluación de las reglas: Se evalúan todas las reglas lingüísticas y se determina el grado de pertenencia de las variables de entrada a cada etiqueta lingüística. Luego, se utilizan estos grados de pertenencia para calcular los coeficientes de las funciones lineales especificadas en la parte ”*THEN”* de las reglas.

- Defuzzificación: Se utiliza una técnica de defuzzificación, como el promedio ponderado, para obtener un valor numérico de salida a partir de las salidas precisas producidas por cada regla.

Método de Tsukamoto

El método de Tsukamoto es otro enfoque para implementar sistemas de control difuso. A diferencia de los métodos de Mamdani y Takagi-Sugeno-Kang, que utilizan reglas lingüísticas para describir la relación entre las variables de entrada y salida, el método de Tsukamoto utiliza una función de membresía que se ajusta en función del error de control [71].

El modelo de Tsukamoto consta de tres partes principales:

- Base de reglas: Se definen las variables de entrada y salida del sistema, y se establecen las etiquetas lingüísticas que se utilizarán para describir las variables. Luego, se definen las reglas lingüísticas *IF-THEN* que describen la relación entre las entradas y la salida del sistema. Sin embargo, en el modelo de Tsukamoto, la parte ”*THEN”* de las reglas especifica una función de membresía que se ajusta en función del error de control.

- Evaluación de las reglas: Se evalúan todas las reglas lingüísticas y se determina el grado de pertenencia de las variables de entrada a cada etiqueta lingüística. Luego, se utiliza el grado de pertenencia para ajustar la función de membresía especificada en la parte ”*THEN”* de las reglas.

- Defuzzificación: Se utiliza una técnica de defuzzificación, como el promedio ponderado, para obtener un valor numérico de salida a partir de las funciones de membresía ajustadas.

Salida del sistema (decodificación)

La salida de un sistema de control difuso es una variable que se mapea desde un conjunto difuso a un valor numérico utilizando una técnica de defuzzificación. La decodificación de la salida es una parte crítica del diseño del sistema de control difuso, ya que determina cómo se traducen las salidas difusas del sistema en valores numéricos que pueden ser utilizados para controlar un sistema físico.

La decodificación de la salida implica el uso de una técnica de defuzzificación para calcular un valor numérico a partir de la salida difusa del sistema. Las técnicas de defuzzificación más comunes son el centroide, el método del máximo y el método del primer máximo. Estas técnicas calculan un valor numérico que representa la salida del sistema en función de la forma de la función de membresía de la salida difusa.

Método del centroide

La técnica de defuzzificación del centroide es la más común y se basa en el cálculo del centro de gravedad de la función de membresía de la salida difusa. El centroide es el punto en el que la masa de la función de membresía está equilibrada. El valor numérico de la salida del sistema se calcula como el centro de gravedad de la función de membresía.

Método del máximo

El método del máximo se basa en la selección del valor de salida con el grado de pertenencia máximo de la salida difusa. El valor numérico de la salida del sistema se calcula como el valor de la etiqueta de conjunto difuso correspondiente al grado de pertenencia máximo.

Método del primer máximo

El método del primer máximo se basa en la selección del primer valor de salida con el grado de pertenencia no nulo en la salida difusa. El valor numérico de la salida del sistema se calcula como el valor de la etiqueta de conjunto difuso correspondiente al primer grado de pertenencia no nulo.

2.6. Métricas de evaluación en modelos de segmentación

2.6.1. *Métrica de evaluación Intersection over Union (IoU)*

La métrica IoU (*Intersection over Union*) se utiliza en la segmentación de imágenes para evaluar la calidad de las máscaras de segmentación generadas por un modelo de redes neuronales.

IoU mide la similitud entre la máscara de segmentación generada por el modelo y la máscara de segmentación de referencia o *ground truth*, que es la máscara de segmentación correcta para la imagen. La métrica se calcula como el área de intersección entre la máscara generada por el modelo y la máscara de referencia, dividida por el área de unión entre ambas máscaras, como se ilustra en la Figura 2.17.

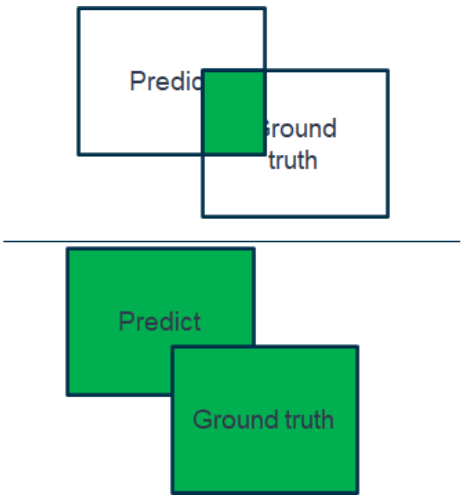


Figura 2.17: Interpretación gráfica de la métrica IoU.

La fórmula matemática del IoU (*Intersection over Union*) se calcula como la relación entre la intersección y la unión de dos conjuntos. En el caso de la segmentación de imágenes, estos dos conjuntos son las máscaras de segmentación generadas por un modelo de redes neuronales y la máscara de referencia. Su fórmula es la siguiente:

$$IoU = \frac{A_I}{A_U} \tag{2.7}$$

Donde: A_I es el área de intersección, es el área donde las dos máscaras de segmentación

se superponen, es decir, la zona donde la segmentación generada por el modelo y la máscara de referencia coinciden, y A_U es el área de unión, es el área total cubierta por ambas máscaras de segmentación.

Una puntuación de IoU cercana a 1 índica que la máscara generada por el modelo se superpone estrechamente con la máscara de referencia, lo que significa que la segmentación es precisa. Por el contrario, una puntuación de IoU cercana a 0 indica que hay una gran discrepancia entre la máscara generada y la máscara de referencia, lo que indica que la segmentación es incorrecta.

IoU se utiliza comúnmente en la evaluación de modelos de detección en imágenes, en tareas de visión por computadora, incluyendo, detección de objetos, segmentación semántica y segmentación de instancias.

2.6.2. *Métrica de evaluación DICE*

La métrica DICE (Índice de Similitud de DICE) se utiliza en la segmentación de imágenes para evaluar la calidad de las máscaras de segmentación generadas por un modelo de redes neuronales.

El Índice de Similitud de DICE mide la similitud entre la máscara de segmentación generada por el modelo y la máscara de segmentación de referencia. La métrica se calcula como el doble de la intersección entre las dos máscaras, dividida por la suma de las áreas de las dos máscaras, como se ilustra en la Figura 2.18.

Definición 2.1 *El coeficiente DICE clásico se define como:*

$$DC := \frac{2|A \cap B|}{|A| + |B|} \tag{2.8}$$

donde A es el conjunto que representa la imagen ground-truth y B representa la segmentación que realizo el modelo.

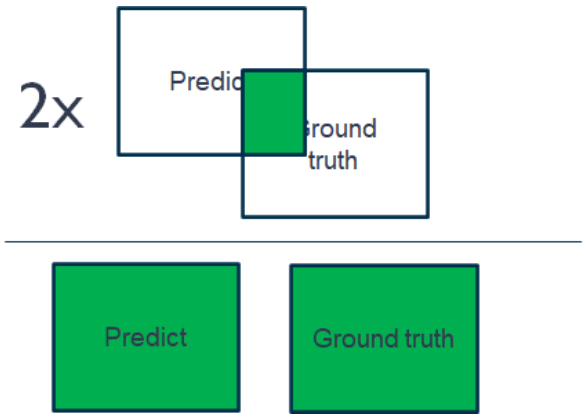


Figura 2.18: Interpretación gráfica de la métrica DICE.

Una puntuación de DICE cercana a 1 indica que la máscara generada por el modelo se superpone estrechamente con la máscara de referencia, lo que significa que la segmentación es precisa. Por el contrario, una puntuación de DICE cercana a 0 indica que hay una gran discrepancia entre la máscara generada y la máscara de referencia, lo que indica que la segmentación es incorrecta [72].

DICE es una métrica popular en la evaluación de modelos de segmentación de imágenes, y se utiliza comúnmente en tareas de visión por computadora, incluyendo segmentación semántica y segmentación de instancias.

2.7. Función de perdida

2.7.1. *Función de perdida Entropía cruzada binaria (BCE)*

La función de pérdida BCE (por sus siglas en inglés, *Binary Cross Entropy*) se utiliza comúnmente en la segmentación de imágenes binarias para medir la diferencia entre la máscara de segmentación generada por un modelo de redes neuronales y la máscara de referencia.

Definición 2.2 *La BCE se define de la siguiente manera:*

$$L_{BCE}(y,\tilde{y}) = -(ylog(\tilde{y}) + (1-y)log(1-\tilde{y})) \tag{2.9}$$

donde: y es la máscara de referencia, y $\tilde{y}$ es el valor predicho por la predicción del modelo.

BCE mide la entropía cruzada binaria entre la máscara generada por el modelo y la máscara de referencia. En la segmentación de imágenes binarias, la máscara de segmentación suele ser una imagen binaria que representa la presencia o ausencia de un objeto en cada píxel. BCE se define como una medida de la diferencia entre dos distribuciones de probabilidad para una variable aleatoria dada o un conjunto de eventos. Es ampliamente utilizada con el objetivo de clasificación, y también para la segmentación.

El objetivo de la función de pérdida BCE es minimizar la distancia entre la máscara de segmentación generada por el modelo y la máscara de referencia. Una baja puntuación de la función de pérdida BCE indica que la máscara generada por el modelo se ajusta bien a la máscara de referencia, lo que significa que la segmentación es precisa. Por el contrario, una alta puntuación de la función de pérdida BCE indica que hay una gran discrepancia entre la máscara generada y la máscara de referencia, lo que indica que la segmentación es incorrecta [73].

BCE es una de las funciones de pérdida más comunes en la segmentación de imágenes binarias y se utiliza en muchos modelos de segmentación de redes neuronales, como *UNet* y *FCN*.

2.7.2. *Función de perdida Entropía cruzada categórica (CCE)*

Para imágenes con más de dos clases o etiquetas en la segmentación de imágenes, la función de pérdida más comúnmente utilizada es la función de entropía cruzada categórica (CCE o *Cross-Entropy loss*, por sus siglas en inglés).

La función de pérdida de entropía cruzada categórica mide la diferencia entre la distribución de probabilidad de las etiquetas de segmentación generadas por el modelo y la distribución de probabilidad de las etiquetas de segmentación de referencia.

Definición 2.3 *La CCE se define de la siguiente manera:*

$$L_{CCE}(y,\tilde{y}) = -\sum(ylog(\tilde{y})) \tag{2.10}$$

donde: y es la máscara de referencia codificada como una distribución de probabilidad, donde cada píxel se codifica como un vector de probabilidad que representa la probabilidad de que ese píxel pertenezca a cada una de las clases, y ỹ es el valor predicho por la predicción del modelo codificado de una forma similar que y [74].

El objetivo de la función de pérdida CCE es minimizar la distancia entre la distribución de probabilidad de las etiquetas de segmentación generadas por el modelo y la distribución de probabilidad de las etiquetas de segmentación de referencia. Una baja puntuación de la función de pérdida CCE indica que la máscara generada por el modelo se ajusta bien a la máscara de referencia, lo que significa que la segmentación es precisa. Por el contrario, una alta puntuación de la función de pérdida CCE indica que hay una gran discrepancia entre la máscara generada y la máscara de referencia, lo que apunta que la segmentación es incorrecta [75].

CCE es una función de pérdida común en la segmentación de imágenes con múltiples clases, y se utiliza en diferentes modelos de segmentación de redes neuronales, como *UNet* y *FCN*.

2.7.3. *Función de perdida focal*

La función de pérdida focal es una función de pérdida diseñada para abordar el problema del desequilibrio de clases en tareas de clasificación, como la segmentación de imágenes y la detección de objetos. Fue introducida por Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He y Piotr Dollár en 2017 en el artículo "*Focal Loss for Dense Object Detection*". La función de pérdida focal es una modificación de la función de pérdida de entropía cruzada binaria, que se utiliza comúnmente para tareas de clasificación.

El problema del desequilibrio de clases ocurre cuando hay una gran diferencia en el número de muestras para diferentes clases en un conjunto de datos. En la detección de objetos y la segmentación de imágenes, la mayoría de los píxeles o regiones de la imagen pueden pertenecer a la clase de fondo, mientras que solo una pequeña proporción de ellos puede pertenecer a las clases de objetos de interés. Este desequilibrio puede hacer que los modelos de aprendizaje profundo sean

propensos a centrarse en las clases dominantes y, en última instancia, a proporcionar un rendimiento deficiente en las clases minoritarias.

La función de pérdida focal aborda este problema al agregar un factor de modulación a la función de pérdida de entropía cruzada binaria. Este factor de modulación se basa en la probabilidad asignada por el modelo a la clase verdadera y se utiliza para reducir el peso de las muestras fácilmente clasificables, es decir, las clases con más muestras, lo que permite que el modelo se enfoque en las muestras difíciles de clasificar o las clases minoritarias [20].

Definición 2.4 *La función de pérdida focal se define como:*

$$FL(p_t) = -\alpha_t * (1 - p_t)^{\gamma} * log(p_t) \tag{2.11}$$

donde:

- p_t *es la probabilidad asignada por el modelo a la clase verdadera.*
- α_t *es un factor de equilibrio de clase que se puede ajustar para dar más importancia a las clases minoritarias.*
- γ *es un parámetro de enfoque que controla el grado de modulación de la función de pérdida (un valor comúnmente utilizado es 2).*

Al agregar el factor de modulación $(1 - p_t)^{\gamma}$, la función de pérdida focal reduce el peso de las muestras que se clasifican fácilmente, lo que permite que el modelo se enfoque en las muestras difíciles de clasificar y mejore su rendimiento en las clases minoritarias.

2.8. Pruebas estadísticas

Estas pruebas pueden ayudar a determinar si las diferencias en el rendimiento de los modelos son estadísticamente significativas.

Algunos objetivos principales de las pruebas estadísticas en la comparación de modelos de IA incluyen:

- Evaluar la bondad de ajuste: Las pruebas estadísticas pueden ayudar a determinar qué tan bien un modelo se ajusta a los datos de entrenamiento y, en última instancia, a predecir resultados en nuevos datos.

- Evitar el sobreajuste: El sobreajuste ocurre cuando un modelo es demasiado complejo y se ajusta demasiado bien a los datos de entrenamiento, pero no generaliza bien a nuevos datos. Las pruebas estadísticas pueden ayudar a equilibrar la complejidad del modelo y su capacidad para ajustarse a los datos, lo que puede prevenir el sobreajuste.

- Comparar modelos: Las pruebas estadísticas permiten comparar diferentes modelos y seleccionar el que tenga el mejor rendimiento en términos de capacidad predictiva y complejidad.

- Validar hipótesis: Al comparar modelos, las pruebas estadísticas pueden ayudar a validar o refutar hipótesis sobre qué características del modelo o enfoques de aprendizaje son más efectivos para un problema específico.

2.8.1. *Prueba de Wilcoxon*

La prueba de Wilcoxon es una prueba estadística no paramétrica que se utiliza para comparar dos muestras relacionadas o pareadas, con el objetivo de determinar si hay diferencias significativas entre ellas. Existen dos versiones de esta prueba: la prueba de Wilcoxon para rangos con signo y la prueba de Wilcoxon para sumas de rangos (también conocido como prueba de Mann-Whitney U).

La prueba de Wilcoxon para sumas de rangos, es una prueba estadística no paramétrica que se utiliza para comparar dos muestras independientes y determinar si hay diferencias significativas entre sus distribuciones [76].

El test de Wilcoxon para sumas de rangos (Mann-Whitney U) sigue estos pasos:

- Recolectar las muestras: Se recogen dos muestras independientes, por ejemplo, las puntuaciones de dos grupos de estudiantes que han sido sometidos a diferentes métodos de enseñanza.

- Unir y asignar rangos: Se combinan ambas muestras en un solo conjunto y se les asigna un rango a cada observación, de menor a mayor. En caso de empate, se asigna el promedio de los rangos correspondientes.

- Calcular sumas de rangos: Se separan las observaciones por grupo y se suman los rangos asignados a cada grupo. Estas sumas son conocidas como $R1$ y $R2$.

- Calcular el estadístico de prueba U: El estadístico de prueba U se calcula para cada grupo utilizando las siguientes fórmulas:

$$U_1 = R_1 - (n_1 * (n_1 + 1))/2 \tag{2.12}$$

$$U_2 = R_2 - (n_2 * (n_2 + 1))/2 \tag{2.13}$$

  Donde n_1 y n_2 son los tamaños de las muestras 1 y 2, respectivamente.

- Seleccionar el menor valor de U: El menor valor de U (U_1 o U_2) es el estadístico de prueba que se utilizará para la comparación.

- Calcular el valor crítico o el valor p: Dependiendo del enfoque que prefieras, puedes comparar el valor de U con un valor crítico obtenido de una tabla de valores críticos de Mann-Whitney (para un tamaño de muestra y nivel de significancia dados) o calcular el valor p correspondiente. Si el valor de U es menor o igual al valor crítico, o si el valor p es menor o igual al nivel de significancia establecido (por ejemplo, 0,05), se rechaza la hipótesis nula (H_0) en favor de la hipótesis alternativa (H_1), lo que indica que hay una diferencia significativa entre las dos muestras independientes [77] [78].

2.8.2. *Prueba de Friedman*

La prueba de Friedman es una prueba no paramétrica que compara múltiples tratamientos relacionados y es útil cuando los datos no cumplen con los supuestos de normalidad y homogeneidad de varianzas. Se puede utilizar la prueba de Friedman para determinar si existe una diferencia significativa entre los modelos en términos de rendimiento y, si es así, realizar comparaciones post

hoc (como la prueba de Nemenyi) para determinar cuáles modelos son diferentes entre sí.

La prueba de Friedman se basa en rangos y compara las medianas de los grupos. La hipótesis nula para la prueba de Friedman es que no hay diferencias significativas entre los grupos o modelos en términos de la variable dependiente. Si el valor p obtenido de la prueba es menor que el nivel de significancia predefinido (generalmente 0.05), se rechaza la hipótesis nula y se concluye que hay diferencias significativas entre al menos dos de los grupos o modelos [79] [80] [81].

2.8.3. *Comparación post hoc*

Las comparaciones post hoc son análisis que se realizan después de una prueba estadística global para investigar las diferencias específicas entre los pares de grupos o modelos. Estas comparaciones se realizan utilizando pruebas estadísticas adicionales, como la prueba de Nemenyi, la prueba de Dunn o la prueba de Bonferroni, para ajustar el nivel de significancia y controlar la tasa de error de tipo I (falsos positivos) debido a las múltiples comparaciones realizadas [82].

2.8.4. *Criterio de Información de Akaike (AIC)*

El Criterio de Información de Akaike (AIC, por sus siglas en inglés) es una métrica que se utiliza para evaluar y comparar diferentes modelos estadísticos. Fue desarrollado por Hirotugu Akaike en 1974. AIC se basa en la teoría de la información y busca encontrar un equilibrio entre la complejidad del modelo y su capacidad para ajustarse a los datos. El objetivo principal es seleccionar el modelo que tenga la mejor capacidad de predicción en un conjunto de datos independiente.

AIC se calcula utilizando la siguiente fórmula:

$$AIC = 2k - 2ln(L) \tag{2.14}$$

Donde:

- k es el número de parámetros en el modelo (incluyendo el término constante si existe).
- L es la verosimilitud máxima del modelo, que es una medida de qué tan bien el modelo se ajusta a los datos.

AIC tiene en cuenta tanto la complejidad del modelo como su bondad de ajuste. Un modelo con menos parámetros (menor complejidad) tendrá un AIC más bajo, mientras que un modelo con mayor verosimilitud (mejor ajuste) también tendrá un AIC más bajo. Por lo tanto, al seleccionar modelos, buscamos aquellos con el AIC más bajo, ya que estos modelos tienen una mejor capacidad de predicción y menor complejidad [83] [84].

2.8.5. *Criterio de Información Bayesiano (BIC)*

El Criterio de Información Bayesiano (BIC, por sus siglas en inglés) es otra métrica que se utiliza para evaluar y comparar diferentes modelos estadísticos. Al igual que el AIC, el BIC busca encontrar un equilibrio entre la complejidad del modelo y su capacidad para ajustarse a los datos. Sin embargo, el BIC tiene una penalización más fuerte para modelos con un mayor número de parámetros en comparación con el AIC, lo que lo hace más adecuado para la selección de modelos en situaciones donde hay una gran cantidad de datos.

El BIC se calcula utilizando la siguiente fórmula:

$$BIC = k * ln(n) - 2 * ln(L) \tag{2.15}$$

Donde:

- k es el número de parámetros en el modelo (incluyendo el término constante si existe).
- n es el número de observaciones en el conjunto de datos.
- L es la verosimilitud máxima del modelo, que es una medida de qué tan bien el modelo se ajusta a los datos.

Al igual que con el AIC, buscamos modelos con el BIC más bajo, ya que estos modelos tienen una mejor capacidad de predicción y menor complejidad. El BIC es especialmente útil cuando se trabaja con conjuntos de datos grandes, ya que su penalización más fuerte para la complejidad del modelo puede ayudar a evitar el sobreajuste [83] [85].

CAPÍTULO 3. Metodología

En esta sección se habla del proceso que se tomó para el análisis de la base de datos CHAOS, utilizando diferentes modelos del estado del arte, comparándolos entre sí y adicionalmente con un modelo propuesto que segmente de manera automática las imágenes de tomografías computarizadas y resonancias magnéticas que se encuentran dentro de la misma base de datos. Este proceso se lleva a cabo por las siguientes secciones: pre-procesamiento y segmentación de imágenes.

En la primera, se realizó la conversión de imágenes médicas en formato *.dcm* a formato *.png* y un proceso de limpieza. En la segunda sección se implementan y comparan diferentes métodos del estado del arte que realicen la segmentación de las imágenes, incluyendo el modelo propuesto. Cada etapa es descrita y desarrollada más adelante en las siguientes secciones.

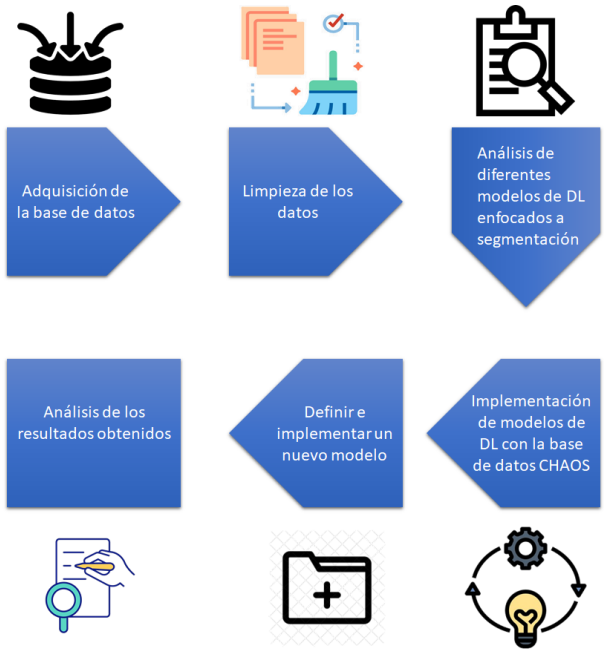


Figura 3.1: Diagrama a bloques del proceso para la implementación del modelo propuesto para segmentar imágenes médicas.

3.1. Pre-procesamiento

3.1.1. *Adquisición de la base de datos*

La base de datos que se utiliza es "*Combined (CT-MR) Healthy Abdominal Organ Segmentation Challenge data*", CHAOS por sus siglas en inglés, y fue descrita en la sección 2.1. La base de datos contiene las imágenes de TC o de RM de 40 pacientes potenciales donantes de hígado. La base de datos se obtuvo de https://chaos.grand-challenge.org/ [22]. El desafío publicado en 2019 da una base de datos particionada en 50 % para entrenamiento y 50 % para pruebas. La partición de entrenamiento cuenta con los *targets,* mientras que la partición para prueba, no cuenta con los *targets*.

La base de datos CHAOS cuenta con un total de 1367 imágenes en entrenamiento, en la carpeta de TC, mientras que en las imágenes de RM se tienen 1594 imágenes de entrenamiento.

3.1.2. *Limpieza de datos*

Las imágenes de la base de datos se proporcionan en formato *.dcm* (DICOM, *Digital Imaging and Communications in Medicine*), es un formato estándar de archivo utilizado en imágenes médicas para la adquisición, almacenamiento, transmisión y visualización de datos. El formato DICOM se utiliza principalmente para imágenes médicas como radiografías, tomografías computarizadas (CT), imágenes de resonancia magnética (MRI), imágenes de ultrasonido y otras modalidades de imágenes médicas. DICOM es un estándar abierto y es compatible con una amplia variedad de equipos de imagen médica y sistemas de información.

El formato DICOM incluye tanto los datos de la imagen como los datos del paciente y de la sesión de adquisición. Los datos del paciente incluyen información como el nombre, la fecha de nacimiento y el género, mientras que los datos de la sesión de adquisición incluyen información como el protocolo utilizado para adquirir la imagen y la fecha y hora de la adquisición [86]. Para fines de la investigación, la información que es de interés, es la información de los píxeles, sin importar la fecha y hora de adquisición, o el nombre y datos del paciente.

Para que los modelos puedan utilizar las imágenes, primero se convierten las imágenes de formato *.dcm* a *.png*. Esto con ayuda de un software especial de libre acceso *MicroDicom* [87]. Los datos deben ajustarse a la dimensión 256×256 píxeles 3.2, esto requerido por los modelos de segmentación para los que servirán como entrada.

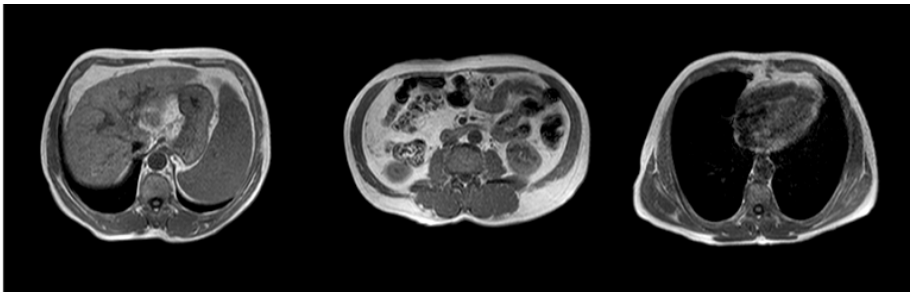


Figura 3.2: Ejemplos de imágenes de la base de datos CHAOS.

Las imágenes que proporciona la base de datos del *ground truth*, se encuentran en formato *.png*, con la consideración de que las imágenes de algunos pacientes se encuentran en dimensión de 256×256 y otros en 288×288. Aplicando un procedimiento de escalamiento (usando la función *resize* en Python) se puede llegar a que todas las imágenes contengan un *ground truth* de 256×256.

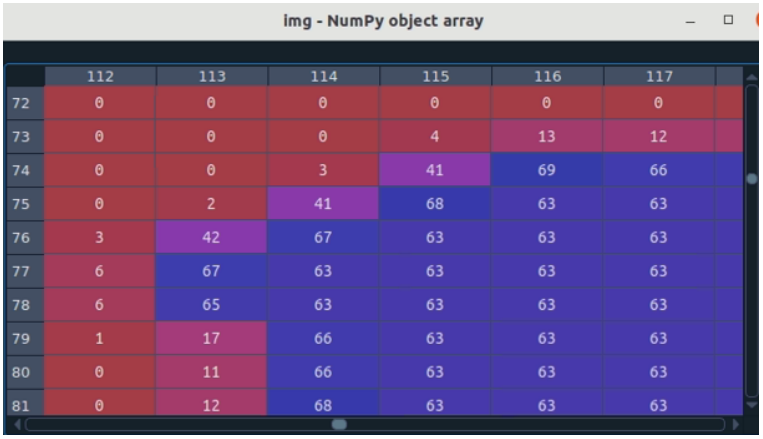
La desventaja que se tiene aplicando el escalamiento, es que la misma función hace un suavizado en la imagen. La imagen 3.3, muestra los píxeles de la zona de la imagen como números, donde cada número diferente representa un color diferente. Para esta imagen se tienen 2 colores, el 0 que representa el fondo o *background* de la imagen, y el 63 que representa un órgano.

img - NumPy object array

	126	127	128	129	130	131
81	0	0	0	0	0	0
82	0	0	0	0	0	0
83	0	0	0	0	63	63
84	0	0	0	63	63	63
85	0	0	63	63	63	63
86	0	63	63	63	63	63
87	0	63	63	63	63	63
88	0	63	63	63	63	63
89	0	0	63	63	63	63
90	0	0	63	63	63	63

Figura 3.3: Arreglo *numpy* del *ground truth* de una imagen de la base de datos CHAOS.

Una vez aplicado el escalamiento se obtiene una imagen como la siguiente 3.4.



	112	113	114	115	116	117
72	0	0	0	0	0	0
73	0	0	0	4	13	12
74	0	0	3	41	69	66
75	0	2	41	68	63	63
76	3	42	67	63	63	63
77	6	67	63	63	63	63
78	6	65	63	63	63	63
79	1	17	66	63	63	63
80	0	11	66	63	63	63
81	0	12	68	63	63	63

Figura 3.4: Arreglo *numpy* del *ground truth* de una imagen de la base de datos CHAOS después de aplicar la función *resize*.

En la imagen 3.4, se observa como aparecen los números, $1,2,3,4,6,11,12,13,17,41,42$, entre otros. Esto para realizar la segmentación representa un problema, porque el modelo tomará cada número como un órgano diferente, por lo que si se tienen 100 números diferentes, esperará encontrar 100 órganos diferentes. Para resolver este problema, se realizó un código en Python, con el cual se leen todas las imágenes de los pacientes que se encontraban en dimensión 288×288, se escala a 256×256, y se convierten los píxeles cercanos a 0, con un rango de ± 31, a la etiqueta de pixel 0. De la misma forma, los píxeles cercanos a 63, con el rango de ± 31, a la etiqueta de pixel 63. Esto para cada una de las 5 etiquetas posibles (1 para el fondo de la imagen y 4 para los órganos).

Teniendo todas las imágenes de la base de datos en la dimensión correcta, el siguiente paso es mapear los píxeles al rango de 0 al n, con n siendo el número de clases u órganos a segmentar.

$$M : \{0,63,126,189,252\} \rightarrow M_t\{0,1,2,3,4\} \tag{3.1}$$

Con esto, obtendremos una imagen de *ground truth* como la mostrada en la figura 3.5.

Por último, las imágenes, tanto del *ground truth* como las que se usan para la extracción de características, se convierten a formato *.npy*, teniendo disponible tanto imágenes en formato *.png* como *.npy* para realizar la segmentación en el modelo.

Figura 3.5: Arreglo *numpy* del *ground truth* de una imagen de la base de datos CHAOS después de aplicar la función *resize* y con los píxeles en el rango de 0 a 4.

3.2. Segmentación de imágenes médicas

Los modelos basados en arquitecturas *encoder-decoder* son ideales para realizar la segmentación de imágenes, los cuales pueden realizar la extracción de características y devolver una imagen con las mismas dimensiones a la que se utilice como entrada.

El conjunto de imágenes para realizar la segmentación, después de aplicar un *resizer* a dimensión de 256×256 píxeles, y aplicar una normalización a las imágenes que se encuentran en escala de grises. La normalización transforma la imagen en escala de grises con dimensión n:

$$I : \{\mathbb{X} \subseteq \mathbb{R}^n\} \rightarrow \{Min, ..., Max\} \tag{3.2}$$

Con valores de intensidad en el rango (Min, Max), en una nueva imagen:

$$I_N : \{\mathbb{X} \subseteq \mathbb{R}^n\} \rightarrow \{newMin, ..., newMax\} \tag{3.3}$$

Con nuevos valores de intensidad $(newMin, newMax)$. Esta normalización se realiza con la fórmula:

$$I_N = (I - Min)\frac{newMax - newMin}{Max - Min} + newMin \tag{3.4}$$

Al momento de tener las imágenes, se define la arquitectura con la que se trabajarán las mismas. Los modelos con arquitecturas *encoder-decoder* realizan el trabajo requerido, al entregar como producto final una imagen con las mismas dimensiones a las que se proporcionaron como entrada. Primero utilizando el *encoder* con una serie de capas convolucionales y de agrupación que reducen gradualmente la resolución espacial de la imagen de entrada, mientras que la sección de decodificación utiliza capas de deconvolución y de unión para aumentar la resolución espacial y generar una máscara de segmentación de salida.

Cuando se ha definido el modelo, se entrena utilizando los datos de entrenamiento preparados anteriormente. Durante el entrenamiento, la red aprende a generar máscaras de segmentación precisas a partir de las imágenes de entrada. Una vez que se ha entrenado y evaluado la red, se puede utilizar para realizar la segmentación de imágenes. Para segmentar una nueva imagen, se pasa la imagen a través de la red entrenada, que genera una máscara de segmentación que indica qué píxeles pertenecen a cada clase.

3.3. Modelos que utilizan la base de datos CHAOS

En el siguiente apartado, se presentan diferentes modelos que utilizan la base de datos CHAOS. En la sección 4 se proporciona una tabla comparativa con los trabajos mencionados.

3.3.1. *PRDNet*

PRDNet (*Progressive Residual Deconvolution Network*) es un modelo de red neuronal convolucional que se utiliza para la segmentación de imágenes médicas. Este modelo se basa en la arquitectura de red de convolución residual (ResNet), con dos ramas, una que continúa con convoluciones residuales y otra con convoluciones dilatadas, además utiliza una técnica de deconvolución progresiva para aumentar la resolución de las características aprendidas.

El modelo consta de 5 capas en el *encoder* que reducen la imagen de entrada, siendo la

capa 4 y 5, dividida en 2 ramas paralelas, una que sigue las convoluciones residuales y la otra con convoluciones dilatadas. En la decodificación, se fusionan todos los mapas de características de cada capa para generar la imagen de salida de la máscara de segmentación, con dimensiones iguales a las de la imagen de entrada.

La modalidad en que se evaluó el modelo PRDNet fue en imágenes de resonancias magnéticas (RM - T1 Dual), segmentando 4 órganos, de la base de datos CHAOS, reportando una partición de 65 % (13 pacientes), 10 % (2 pacientes) y 25 % (5 pacientes), para entrenamiento, validación y pruebas, respectivamente. La media del coeficiente DICE que se obtuvo al evaluar el modelo en la base de datos después de 3 repeticiones del experimento fue de 90,2 % [8].

3.3.2. *RMS-UNet*

RMS-UNet es un modelo de red neuronal convolucional que se basa en la arquitectura de UNet y utiliza una técnica de modulación de escala regional (RMS) para mejorar la precisión de la segmentación.

En lugar de utilizar una sola escala para realizar la segmentación, RMS-UNet utiliza múltiples escalas para adaptarse a las diferentes características y tamaños de las estructuras de interés en la imagen. La técnica de modulación de escala regional (RMS) se utiliza para ajustar el peso de las diferentes escalas en función de la información relevante en cada región de la imagen. La arquitectura del modelo se muestra en la figura 3.6.

El modelo utiliza tomografías computarizadas (TC) en la modalidad de segmentación del hígado. Para los experimentos que se reportan se usa una partición utilizando únicamente 10 pacientes es de 70 %, 15 % y 15 %, para entrenamiento, validación y pruebas, respectivamente. Al evaluar, el coeficiente DICE para la segmentación del hígado con el modelo RMS-UNet es de 95,49 % [9].

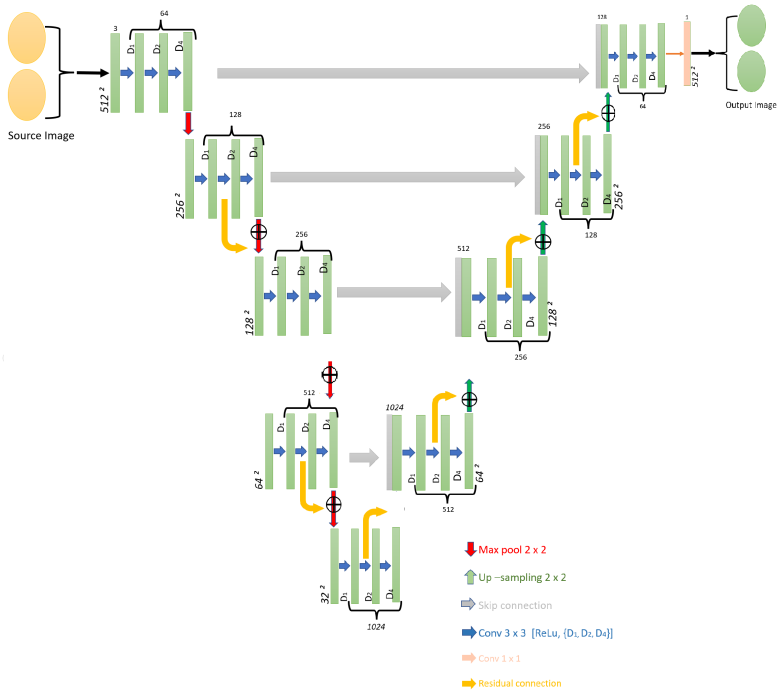


Figura 3.6: Arquitectura del modelo RMS-UNet.

3.3.3. *MiDeepSeg*

MiDeepSeg es un modelo de red neuronal convolucional diseñado específicamente para la segmentación de imágenes médicas, como imágenes de tomografía computarizada (TC) y resonancia magnética (MRI). Este modelo se basa en la arquitectura de red neuronal convolucional UNet y utiliza una técnica de aprendizaje semi-supervisado para mejorar la precisión de la segmentación. En lugar de depender exclusivamente de conjuntos de datos etiquetados, MiDeepSeg utiliza una combinación de datos etiquetados y no etiquetados para entrenar la red.

Esta técnica basada en la interacción con el usuario, únicamente requiere de pocos datos y a su vez, que el usuario proporcione una pequeña cantidad de clics en las áreas de interés, usando como referencia el margen del objeto u órgano a segmentar. Esto es útil en situaciones en las que es difícil obtener grandes conjuntos de datos etiquetados para el entrenamiento del modelo [10].

El modelo solo toma 2 pacientes tanto del conjunto de TC como de MR (T1 y T2), para la

validación y segmentar el hígado y el bazo, de la base de datos CHAOS. El coeficiente DICE que se obtuvo al evaluar el modelo en la validación, fue de 96,93 %.

3.3.4. *DCACNet*

DCACNet (*Dual-Channel Attention-Based Convolutional Neural Network*) es un modelo de red neuronal convolucional que se utiliza para la segmentación de imágenes médicas. Este modelo utiliza una arquitectura de red dual que se compone de dos ramas independientes, una para extraer características de la imagen y otra para extraer características de la etiqueta de segmentación. La información extraída de estas dos ramas se fusiona en una capa de atención para mejorar la precisión de la segmentación.

La capa de atención en DCACNet utiliza una técnica de atención espacial para resaltar las regiones de la imagen que son más importantes para la segmentación y una técnica de atención de canal para resaltar los canales de características más relevantes para la tarea de segmentación. Este modelo cuenta con 3 partes: el codificador, el módulo de atención DCAM y el decodificador.

El modelo utiliza CHAOS solo en la modalidad de RM-T1 para segmentar los 4 órganos abdominales, usando los 20 pacientes que cuentan con el *ground truth*, reportando una partición de 60 % (12 pacientes), 20 % (4 pacientes) y 20 % (4 pacientes), para entrenamiento, validación y pruebas, respectivamente. La media del coeficiente DICE que se obtuvo al evaluar el modelo en la base de datos fue de 91,03 % [11].

3.3.5. *FFANet*

FFANet (Feature Fusion Attention Network) es un modelo de red neuronal convolucional basada en la fusión de características, en la que las características extraídas de diferentes niveles de la red se fusionan para mejorar la precisión de la segmentación. Además, utiliza una capa de atención para resaltar las regiones más importantes de la imagen y mejorar la precisión de la segmentación. La arquitectura de la red se puede ver en la figura 3.7.

La capa de atención en FFANet utiliza una técnica de atención basada en canales y otra basada en características espaciales para resaltar las regiones más relevantes para la tarea de segmentación. Además, la arquitectura de red basada en la fusión de características utilizada por FFA-Net permite que el modelo capture información más rica y detallada de la imagen, lo que puede mejorar la precisión de la segmentación.

La modalidad en que se evaluó el modelo FFANet fue en imágenes de resonancias magnéticas (RM - T1 Dual), segmentando 4 órganos, de la base de datos CHAOS, reportando una partición de 65 % (13 pacientes), 10 % (2 pacientes) y 25 % (5 pacientes), para entrenamiento, validación y pruebas, respectivamente. La media del coeficiente DICE que se obtuvo al evaluar el modelo fue de 90,9 % [14].

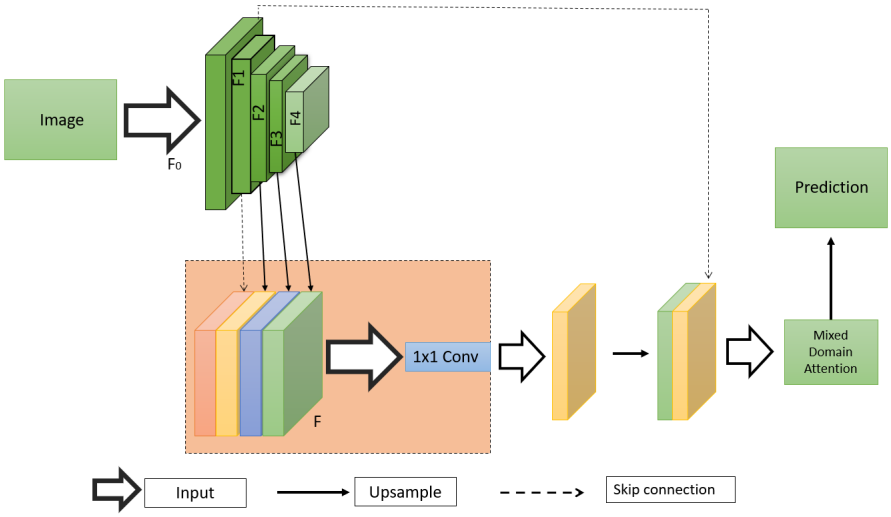


Figura 3.7: Arquitectura del modelo FFANet.

3.3.6. *MS-GAN*

MS-GAN (*Multi-Scale Guided Attention Network*) es un modelo de red neuronal convolucional que utiliza una arquitectura de red basada en atención y múltiples escalas para la segmentación de imágenes médicas. El modelo utiliza una técnica de atención guiada para enfocar la atención del modelo en las regiones relevantes de la imagen para la tarea de segmentación. La técnica de atención guiada en MS-GAN utiliza un mapa de atención previamente generado para

guiar la atención en las regiones importantes.

Utilizando múltiples escalas para la segmentación de imágenes permite que el modelo capture características a diferentes niveles de detalle. Esta técnica permite que el modelo se centre en las regiones importantes de la imagen, lo que puede mejorar la precisión de la segmentación. Esto ayuda a evitar que se utilicen varias veces la misma información extraída de múltiples escalas y también que no se modele de forma correcta las dependencias de largo alcance.

Usando mecanismos de autoatención guiada se pueden integrar características locales con sus dependencias globales, así como resaltar mapas de características interdependientes, adaptativamente. Además, los mecanismos de atención descuidan la información irrelevante y se enfocan en regiones de mayor importancia de la imagen, asociando características relevantes. La arquitectura de la red se puede ver en la figura 3.8.

La modalidad en que se evaluó el modelo MS-GAN fue en imágenes de resonancias magnéticas (RM - T1 Dual), segmentando 4 órganos, de la base de datos CHAOS, reportando una partición de 65 % (13 pacientes), 10 % (2 pacientes) y 25 % (5 pacientes), para entrenamiento, validación y pruebas, respectivamente. La media del coeficiente DICE que se obtuvo al evaluar el modelo fue de 86,75 % [88].

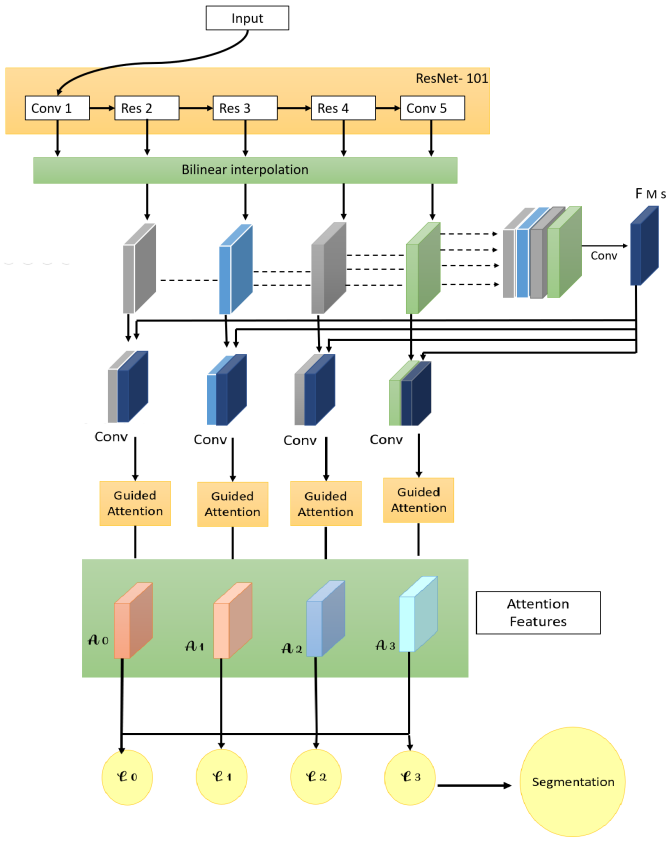


Figura 3.8: Arquitectura del modelo MS-GAN.

3.3.7. *CGAN*

El método propuesto se basa en dos redes neuronales convolucionales profundas en cascada y una red adversaria generativa (GAN) para mejorar la precisión de la segmentación y reducir el ruido de las imágenes de TC. La primera red convolucional en cascada se utiliza para la segmentación inicial de los órganos, mientras que la segunda red se utiliza para la refinación de la segmentación. La red GAN se utiliza para mejorar la calidad de las imágenes de entrada.

Esta técnica se basa en el uso de múltiples dominios para entrenar el modelo y mejorar su capacidad para segmentar órganos en diferentes situaciones y con diferentes tipos de ruido. La red realiza un entrenamiento *end-to-end* para beneficiarse de los refinamientos de segmentación multi-nivel simultáneos utilizando el contexto automático. La combinación de redes GAN y convolucionales en cascada fortalece la capacidad de las canalizaciones de aprendizaje profundo para

delinear automáticamente múltiples órganos abdominales, con una buena capacidad de generalización. En la figura 3.9 se muestra la arquitectura de este modelo.

La modalidad en que se evaluó el modelo CGAN fue en imágenes de resonancias magnéticas (RM - T1 Dual), segmentando 4 órganos, y con las imágenes de tomografías computarizadas (TC) para segmentar el hígado, reportando una partición de 50 % (10 pacientes), y 50 % (10 pacientes), para entrenamiento y pruebas, respectivamente. La media del coeficiente DICE que se obtuvo al evaluar el modelo fue de 89,853 % y 97,87 % para RM y TC, respectivamente [89].

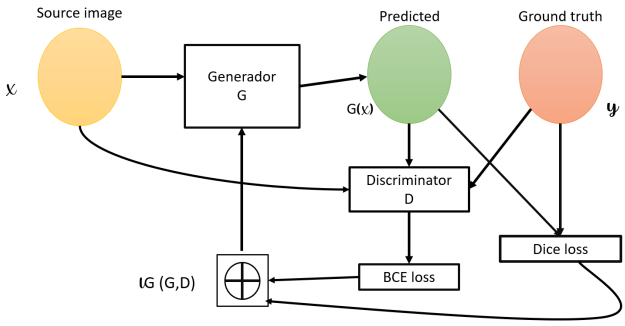


Figura 3.9: Arquitectura del modelo CGAN.

3.3.8. *UGAN*

El método propuesto utiliza una red generativa adversaria unificada (UGAN) que combina un modelo de segmentación y un modelo de generación de imágenes en una única arquitectura de red. El modelo de generación de imágenes se entrena para generar imágenes sintéticas a partir de diferentes modalidades, mientras que el modelo de segmentación se entrena para segmentar las imágenes sintéticas generadas.

El modelo se entrena utilizando un conjunto de datos de imágenes médicas en 3D no pareadas, lo que significa que las imágenes en diferentes modalidades no están directamente relacionadas una con la otra. Para abordar este problema, se utilizan dos técnicas de preprocesamiento: la normalización del espacio de características y el emparejamiento de intensidad. Estas técnicas permiten que las imágenes en diferentes modalidades se transformen en el mismo espacio de características y se ajusten a la misma escala de intensidad, lo que permite que el modelo de UGAN

funcione de manera efectiva.

La modalidad en que se evaluó el modelo UGAN fue en imágenes de resonancias magnéticas (RM - T1 Dual), segmentando 4 órganos, reportando una partición de 65 % (13 pacientes), y 35 % (7 pacientes), para entrenamiento y validación, respectivamente. La media del coeficiente DICE que se obtuvo al evaluar el modelo en la base de datos después de 5 repeticiones del experimento fue de 84,03 % [90].

3.3.9. *Cycle-Consistent GAN*

El modelo *Cycle-Consistent GAN* presenta un enfoque de aprendizaje profundo para la segmentación de imágenes médicas que utiliza etiquetas pseudo-ruidosas y aprendizaje adversario para mejorar la eficiencia del proceso de anotación.

En este enfoque, se utiliza una red neuronal convolucional (CNN) para la segmentación de imágenes médicas y se entrena con un conjunto inicial de datos etiquetados. Luego, se utiliza la CNN entrenada para generar etiquetas pseudo-ruidosas para un conjunto no etiquetado de imágenes médicas. Estas etiquetas pseudo-ruidosas se utilizan para entrenar nuevamente la CNN en un proceso de aprendizaje adversario, donde una red discriminadora se entrena para distinguir entre las etiquetas reales y las etiquetas pseudo-ruidosas.

Este proceso de aprendizaje adversario mejora la calidad de las etiquetas pseudo-ruidosas y, por lo tanto, mejora la precisión de la segmentación de imágenes médicas. Además, este enfoque también reduce la necesidad de anotaciones manuales costosas y mejora la eficiencia del proceso de anotación.

La modalidad en que se evaluó el modelo *Cycle-Consistent GAN* fue en imágenes de tomografías computarizadas, segmentando el hígado, reportando una partición de 70 % (14 pacientes), 10 % (2 pacientes) y 20 % (4 pacientes), para entrenamiento, validación y prueba, respectivamente. La media del coeficiente DICE que se obtuvo al evaluar el modelo en la base de datos fue de 88,9 % [91].

3.3.10. *CASA*

El CASA presenta un enfoque de aprendizaje profundo para la segmentación de imágenes médicas que utiliza la apariencia colaborativa y adaptación semántica. En este enfoque, se utilizan dos redes neuronales convolucionales (CNN) diferentes para realizar la segmentación de la imagen. La primera red, llamada Red de Adaptación de Apariencia (AAN), se entrena para aprender las características de apariencia de la imagen. La segunda red, llamada Red de Adaptación Semántica (SAN), se entrena para aprender las características semánticas de la imagen.

El proceso de segmentación se realiza mediante una combinación de las características de apariencia y semántica de la imagen. La red AAN se utiliza para adaptar las características de apariencia de la imagen a un espacio común, mientras que la red SAN se utiliza para adaptar las características semánticas de la imagen a un espacio común. Estos dos espacios se combinan para obtener características adaptadas que se utilizan para la segmentación de la imagen.

La modalidad en que se evaluó el modelo CASA fue en imágenes de resonancias magnéticas (RM - T1 Dual), segmentando 4 órganos, y en tomografías computarizadas, segmentando el hígado, usando como partición todo el conjunto de RM más el 80 % del conjunto de TC para entrenamiento, y el restante 20 % del conjunto de TC para pruebas. La media del coeficiente DICE que se obtuvo al evaluar el modelo con los dos conjuntos simultáneos es de 84,9 % [92].

3.3.11. *FCCRF*

El modelo describe un enfoque eficiente para la segmentación interactiva de imágenes médicas en 2D y 3D utilizando un campo aleatorio condicional completamente conectado (FCCRF, por sus siglas en inglés *Fully Connected Conditional Random Field*).

La herramienta de segmentación interactiva *Interactive Segmentation GUI* propuesta permite a los usuarios definir y modificar las regiones de interés (*ROI*, regions of interest) en las imágenes médicas mediante un proceso iterativo en el que el usuario interactúa con la herramienta y proporciona información sobre la segmentación deseada.

El modelo FCCRF utiliza información de características de intensidad de píxeles y características de textura para realizar la segmentación de la imagen. Además, el modelo utiliza información de contexto en la imagen para mejorar la precisión de la segmentación. Esto se logra mediante la creación de un grafo completamente conectado que conecta todos los píxeles en la imagen y utiliza la información de intensidad y textura de los píxeles vecinos para mejorar la segmentación de cada píxel en la imagen.

Los resultados experimentales muestran que la herramienta de segmentación propuesta es capaz de segmentar con precisión imágenes médicas. La modalidad en que se evaluó el modelo FCCRF fue en imágenes de resonancias magnéticas (RM - T1 Dual), segmentando 4 órganos, únicamente para pruebas. La media del coeficiente DICE que se obtuvo al evaluar el modelo en la base de datos fue de 89,95 % [93].

3.4. *Transformers*

Los mecanismos de atención son la base fundamental de los transformers. Los transformers son una arquitectura de red neuronal que utiliza múltiples capas de atención para procesar secuencias de entrada, como texto, señales de audio o imágenes.

En otras palabras, los transformers utilizan mecanismos de atención para enfocarse en partes relevantes de la entrada y para calcular las relaciones entre las diferentes partes de la misma. Estos mecanismos de atención permiten a los transformers modelar secuencias de entrada con mayor precisión que las arquitecturas de red neuronal anteriores.

Los transformers se han utilizado principalmente en el procesamiento de lenguaje natural, pero también se pueden adaptar para su uso en imágenes. El proceso de adaptación implica modificar la arquitectura del transformer para que pueda procesar datos de imagen en lugar de datos de texto.

- Cambiar la entrada: Los transformers para procesamiento de lenguaje natural utilizan vectores de palabras como entrada, mientras que para procesamiento de imágenes se utiliza una matriz de píxeles. Por lo tanto, se debe modificar la entrada de la red para que acepte matrices de píxeles como parches.

- Modificar la atención: En el procesamiento de lenguaje natural, la atención se enfoca en las relaciones entre las palabras en una secuencia. En el procesamiento de imágenes, la atención se enfoca en las relaciones espaciales entre los píxeles. Por lo tanto, es necesario modificar la atención en el transformer para que tenga en cuenta las relaciones espaciales entre los píxeles.

- Cambiar la salida: En el procesamiento de lenguaje natural, la salida es una secuencia de palabras, mientras que en el procesamiento de imágenes, la salida es una etiqueta de clase. Por lo tanto, es necesario modificar la salida del transformer para que genere una etiqueta de clase en lugar de una secuencia de palabras.

- Ajustar la arquitectura: También es posible que se deba ajustar la arquitectura del transformer para adaptarse a las imágenes. Por ejemplo, se puede agregar una capa de convolución antes de la entrada para extraer características de la imagen.

3.5. Modelos que utilizan *Transformers*

En el campo de la segmentación de imágenes médicas, el uso de modelos basados en Transformers ha demostrado ser una estrategia eficaz para lograr resultados sobresalientes en una variedad de tareas. Sin embargo, hasta ahora no se ha propuesto ningún modelo de segmentación basado en Transformers diseñado específicamente para manejar la base de datos CHAOS. A pesar de esta falta de literatura, es importante resaltar que dichos modelos se han aplicado con éxito a otras tareas de segmentación en el campo médico y tienen el potencial de abordar los desafíos únicos de la base de datos CHAOS. El objetivo de este estudio es llenar este vacío e investigar la aplicabilidad de modelos basados en transformadores en la segmentación de la base de datos CHAOS, con la esperanza de lograr resultados igualmente notables.

3.5.1. *Trans-UNet*

El modelo Trans-UNet es una arquitectura de red neuronal que combina dos técnicas de procesamiento de imágenes: los *transformers* y U-Net.

El funcionamiento del modelo Trans-UNet se puede resumir de la siguiente manera:

- Extracción de características y codificación: El modelo comienza extrayendo características de la imagen de entrada utilizando capas convolucionales. La imagen de entrada se pasa a través de una serie de capas de convolución y pooling para extraer características de la imagen y reducir su resolución.
- Transformación: Las características se pasan a través de una serie de capas de *transformers* para calcular las relaciones espaciales entre las características. Estas capas de *transformers* son similares a las que se utilizan en el procesamiento de lenguaje natural para calcular las relaciones entre las palabras. Estas capas transforman la información de la imagen en una representación más abstracta y jerárquica que contiene información contextual sobre la imagen.
- Decodificación: Las características transformadas se pasan a través de una serie de capas de convolución y *upsampling* para reconstruir la imagen original. Esta etapa utiliza una arquitectura U-Net, que es conocida por su capacidad para reconstruir imágenes de alta calidad a partir de características de baja resolución.
- Fusión de características: En este punto, el modelo utiliza capas de atención para fusionar la información de las capas de codificación y decodificación. Esta fusión permite que el modelo utilice información contextual de la imagen a diferentes escalas, lo que ayuda a mejorar la precisión de la segmentación de la imagen.
- Segmentación: Finalmente, el modelo utiliza una capa de salida para generar una máscara de segmentación que identifica las diferentes regiones de la imagen.

3.5.2. *MCTrans*

El modelo MCTrans es un modelo de segmentación de imágenes que utiliza una arquitectura de transformer para lograr una segmentación precisa y eficiente. La idea central detrás de MCTrans es utilizar múltiples canales de entrada, incluyendo la imagen original, características de baja resolución y características de alta resolución, para crear una representación completa de la imagen.

El funcionamiento del modelo MCTrans se puede resumir de la siguiente manera:

- Preprocesamiento: se realiza el procesamiento previo a la entrada de la imagen, como por ejemplo, la normalización y redimensionamiento de la imagen de acuerdo a lo que espere el modelo como entrada.
- Extracción de características y codificación: El modelo comienza extrayendo características de la imagen preprocesada de entrada utilizando capas convolucionales (CNN) para extraer características, mediante un *backbone* de una red VGG o *ResNet*.
- Transformación: Las características se pasan a través de una serie de capas de *transformers* para calcular las relaciones espaciales entre las características. Estas capas de *transformers* son capas de *transformer Self-Attention* (TSA) y *transformer Cross-Attention* (TCA), de los cuales el TSA se modifica para tener un *transformer Deformable Self-Attention* (DSA), para reducir la complejidad computacional dentro del cálculo del *transformer*, de complejidad cuadrática a complejidad $nlog(n)$.
- Decodificación: Las salidas de la última capa del TCA se pasan a través de una serie de capas de convolución y *upsampling* para reconstruir la imagen original.
- Fusión de características: En este punto, el modelo utiliza capas de atención para fusionar la información de las capas de codificación y decodificación. Esta fusión se realiza de acuerdo a los niveles en que se realizó la codificación, agrupando el nivel 1 de la codificación, con el último nivel de la decodificación, un proceso similar al que realiza el modelo UNet.

- Segmentación: Finalmente, el modelo utiliza una capa de salida para generar una máscara de segmentación que identifica las diferentes regiones de la imagen.

En el transcurso de esta investigación, se ha llevado a cabo una exhaustiva evaluación de diversos modelos de segmentación basados en Transformers. Esta elección se sustenta en los resultados sobresalientes previamente reportados en tareas de segmentación con otras bases de datos médicas. La variedad de modelos evaluados en este estudio se justifica por la necesidad de adaptar y optimizar las capacidades de estos modelos para abordar los desafíos específicos presentados por la base de datos CHAOS.

3.6. Modelo propuesto: FF-TransUnet (Fuzzy Focal TransUnet)

En esta sección se presenta un modelo de aprendizaje profundo que utiliza la base de datos CHAOS para abordar el problema de la segmentación de órganos abdominales. El modelo sugerido se basa en U-Net, una arquitectura de red neuronal que ha demostrado ser efectiva en tareas de procesamiento de imágenes y segmentación. Sin embargo, la poderosa técnica de atención basada en transformers se ha integrado para mejorar aún más el rendimiento del modelo y capturar relaciones a largo plazo en los datos.

Además, se propone una modificación de la función de pérdida utilizada en el entrenamiento del modelo. Se utiliza una función de pérdida focal en lugar de la función de pérdida tradicional. Esta función se caracteriza por asignar un mayor peso a los datos con pocas muestras, lo que ayuda al modelo a concentrarse en los casos más difíciles y mejora su capacidad de generalización.

Se ha implementado un sistema de control difuso de tipo Mamdani para optimizar los parámetros de la función de pérdida focal. Los parámetros de la función de pérdida focal de este sistema de control difuso se pueden ajustar dinámicamente según las características y complejidad de los datos de entrada. De esta manera, el objetivo es maximizar la capacidad de aprendizaje y el desempeño del modelo en una variedad de escenarios y condiciones. Las siguientes secciones detallarán cómo se implementó y entrenó este modelo. También se realizaron experimentos para

evaluar su eficacia y compararlo con otros enfoques del estado del arte.

3.6.1. *Arquitectura FF-TransUnet*

La segmentación de la base de datos CHAOS se realiza utilizando la arquitectura U-Net como base del modelo sugerido. La arquitectura U-Net es ampliamente utilizada en tareas de segmentación porque puede capturar tanto información de bajo nivel como contextual a través de conexiones skip. Las características de diferentes escalas se pueden fusionar gracias a estas conexiones, lo que mejora la precisión de la segmentación.

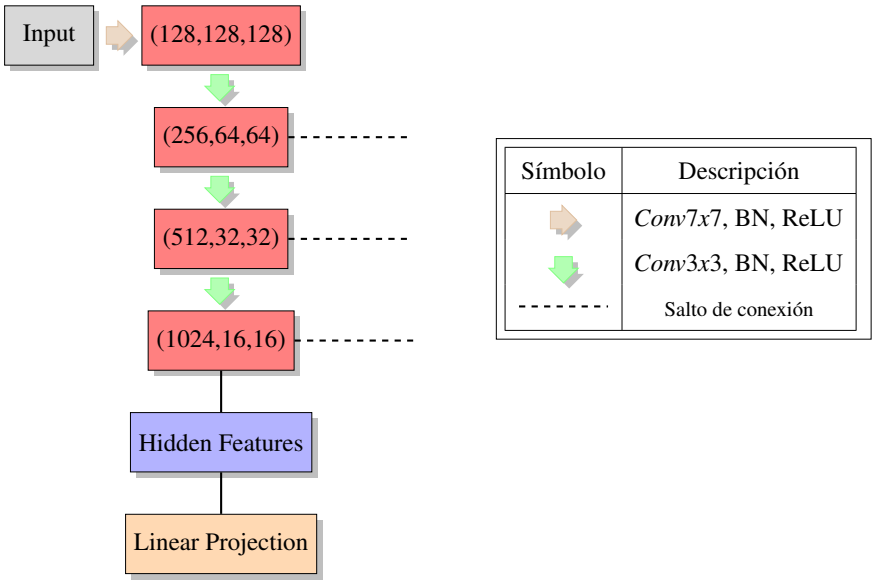


Figura 3.10: Codificador del modelo FF-TransUnet propuesto.

Se aplican tres capas de convoluciones en el bloque de convoluciones hacia abajo del modelo para extraer características de la imagen de entrada. A medida que profundizamos en la red, estas capas de convoluciones capturan características más abstractas y se encargan de reducir gradualmente la resolución espacial de la imagen. Además, se agrega una capa de proyección lineal, que sirve como entrada para las capas de transformadores, como se muestra en la Figura 3.10.

El uso de capas de transformadores en el modelo es una incorporación novedosa y efectiva. Los transformadores son conocidos por su capacidad para capturar relaciones a largo plazo en los

datos y han demostrado ser muy buenos en tareas de procesamiento de lenguaje natural. Se busca aprovechar esta capacidad para capturar relaciones y contextos complejos en la imagen al utilizar las características proyectadas de la capa lineal como entrada para las capas de transformadores.

Las capas iniciales de transformadores del modelo se encargan de procesar las características proyectadas. Cada capa de transformador se compone de múltiples cabezales de atención y capas de alimentación hacia adelante, lo que permite al modelo capturar relaciones no lineales y modelar la dependencia a largo plazo entre los píxeles de la imagen.

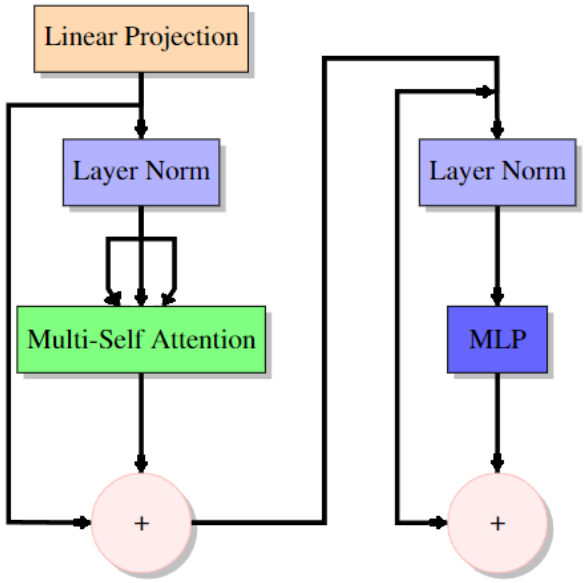


Figura 3.11: Esquema de un bloque de transformadores usados en el modelo FF-TransUnet. Dentro de donde puede ajustarse el número de capas deseadas.

Luego de que los datos son procesados en las capas de transformadores se realiza un reagrupamiento de datos en la salida del modelo para pasar al bloque de convoluciones hacia arriba. El objetivo de este bloque es crear una imagen con dimensiones iguales a las de entrada utilizando las características que han sido procesadas por las capas de transformadores.

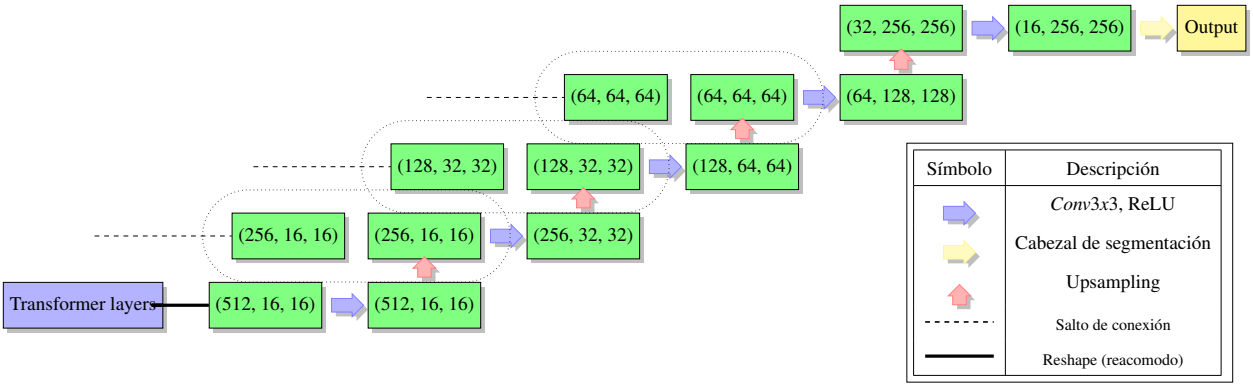


Figura 3.12: Decodificador del modelo FF-TransUnet propuesto.

En la etapa del decodificador se utilizan capas de convoluciones, al igual que en el bloque de convoluciones hacia abajo, para restaurar la resolución espacial y producir una imagen segmentada del mismo tamaño que la imagen original. Un diagrama simplificado se muestra en la figura 3.13.

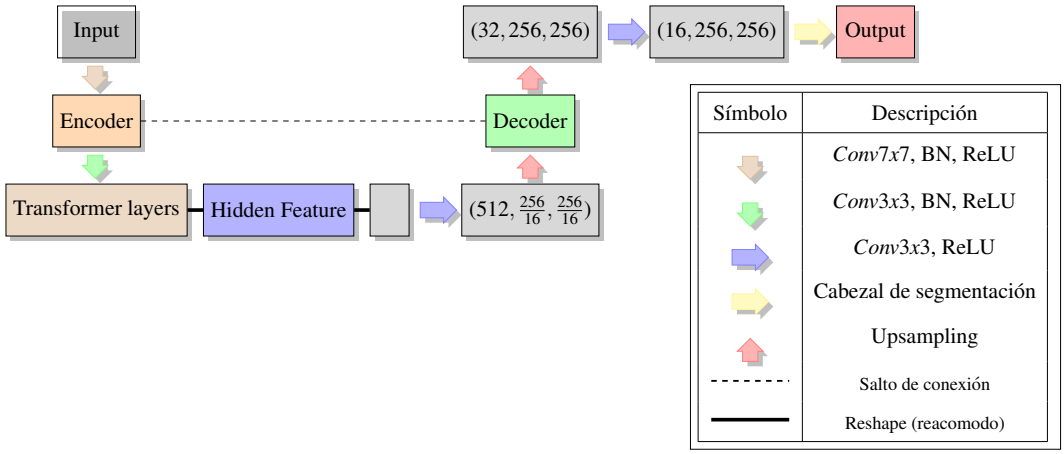


Figura 3.13: Diagrama simplificado del modelo TransUnet. La imagen de entrada pasa al *Down Block*, luego el mapa de características aplanado se pasa a las *Transformer layers*. Después de la capa *Hidden Feature* se realiza un *reshape*, después una convolución de 3*x*3 con una capa *ReLU* y pasa al *Up Block*. Por último, se realiza una convolución final y se pasa al cabezal de segmentación para obtener la imagen de salida.

El modelo sugerido, TransUnet, utiliza la función de pérdida de entropía cruzada, que es una función común para la segmentación de múltiples clases. Sin embargo, en este modelo se utilizó una función de pérdida focal en lugar de la función de pérdida de entropía cruzada.

Como ya se mencionó anteriormente, una alternativa a la función de entropía cruzada, la función de pérdida focal ha demostrado ser eficaz para resolver desequilibrios en conjuntos de datos y mejorar la segmentación de clases difíciles. Esta función da un mayor peso a los píxeles de las clases difíciles de segmentar, lo que permite al modelo concentrarse más en esas clases y mejorar su capacidad de segmentación.

Se ha observado que, en el caso de la base de datos utilizada en este estudio, existen clases que enfrentan problemas adicionales debido a la falta de datos de entrenamiento. El modelo propuesto ha incorporado la función de pérdida focal para abordar estos problemas. Se busca mejorar la capacidad del modelo para segmentar estas clases de manera más precisa y robusta al asignar un mayor peso a los píxeles de las clases difíciles de segmentar debido a la falta de datos.

La función de pérdida focal definida en la sección 2,7,3, utiliza 2 parámetros: α y γ. Estos parámetros tienen la siguiente utilidad:

- α es un parámetro que controla el balance entre las clases difíciles y fáciles de segmentar. Un valor alto de α da más peso a las clases difíciles. El valor que se le da comúnmente a este parámetro es de 0,25, sin embargo no hay un valor único y recurrente para el parámetro alfa en la función de pérdida focal que sea aplicable a todos los problemas.
- γ es un parámetro llamado el exponente focal, que controla la amplitud del peso asignado a los errores. Un valor alto de γ enfoca el modelo en corregir los errores cometidos en los píxeles más difíciles de clasificar.

En el presente estudio, se ha incorporado un sistema innovador que actualiza de forma dinámica el parámetro α para cada clase durante el entrenamiento. A diferencia de la práctica común, donde α se mantiene fijo a lo largo del entrenamiento, esta investigación propone ajustar periódicamente el parámetro α con el objetivo de mejorar la segmentación de imágenes médicas de resonancia magnética en la base de datos CHAOS.

En la Figura 3.14, se muestra un diagrama de flujo que ilustra el proceso de ajuste del parámetro α a partir de la época 3 del entrenamiento. Se emplea un sistema de control difuso de

tipo Mamdani con el método del centroide, el cual se basa en dos entradas fuzzyficadas.

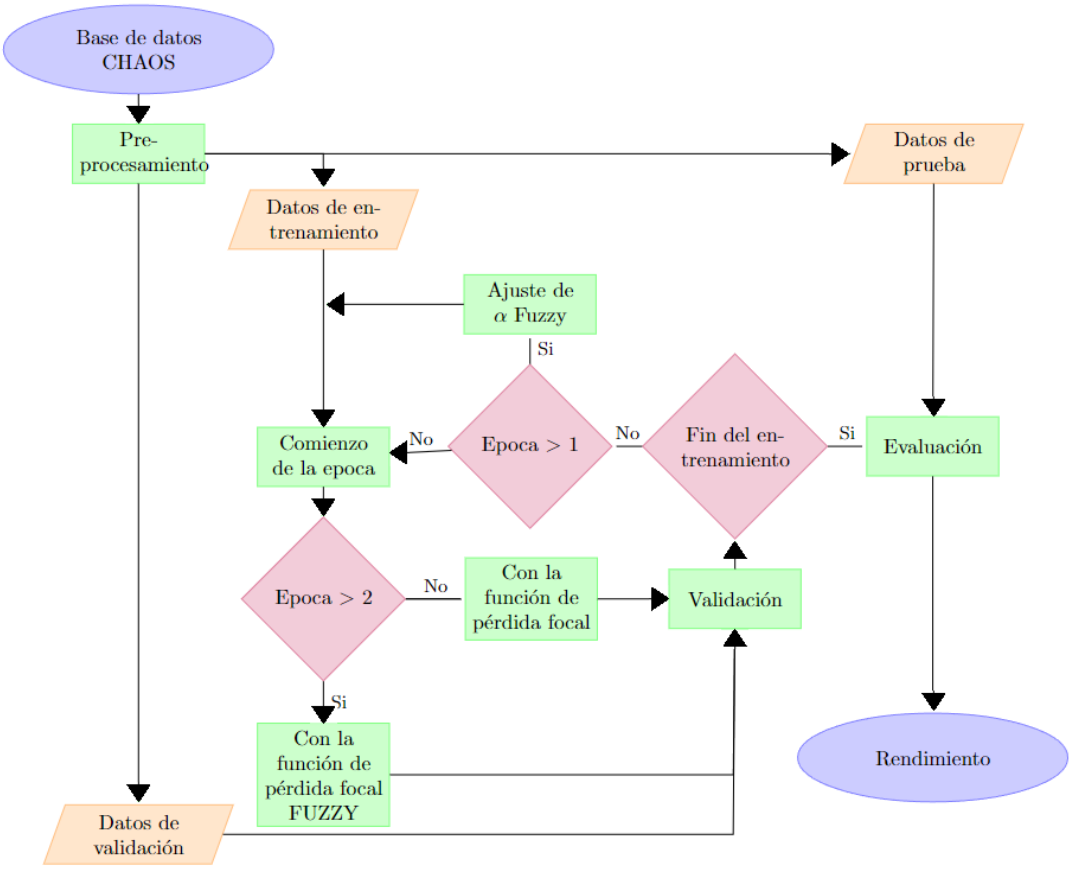
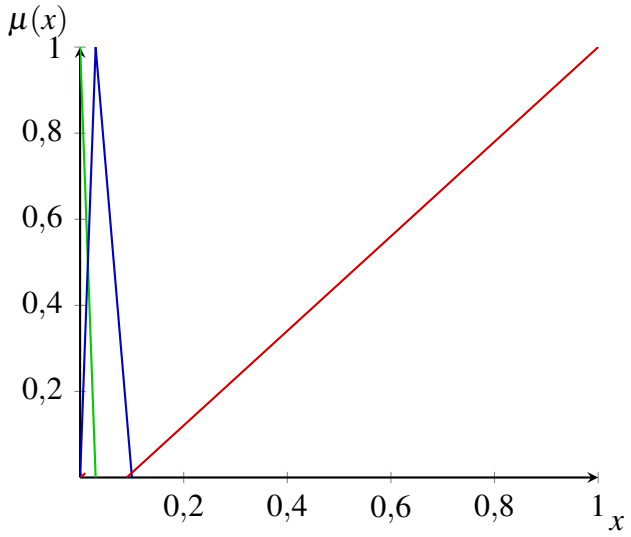


Figura 3.14: Diagrama de flujo del modelo FF-TransUnet, incorporando la actualización del parámetro α en la función de perdida focal.

El modelo puede adaptarse a las características cambiantes de los datos de entrenamiento y concentrarse en las clases más difíciles de segmentar en cada etapa de entrenamiento gracias a esta actualización periódica del parámetro α. El modelo puede priorizar las clases con problemas específicos y mejorar la precisión de la segmentación ajustando α de forma iterativa.

La primera entrada utilizada en el sistema difuso se refiere a la razón del tamaño de las clases, considerando las clases de fondo, hígado, riñón izquierdo, riñón derecho y bazo. Cada clase tiene asociado un valor que representa su tamaño relativo, y se ha realizado la fuzzyficación para categorizar los tamaños en tres etiquetas: ”muchos”, clases con muchos píxeles, ”masomenos”, para pocos datos pero no escasos, y ”pocos”, para datos con escasas muestras, como se muestra en

la figura 3.15.



Símbolo	Descripción
	Tamaño de clase ”pocos”
	Tamaño de clase ”masomenos”
	Tamaño de clase ”muchos”

Figura 3.15: Función de membresía para el tamaño de las clases.

Se define la Figura 3.15 de acuerdo a las siguientes funciones para cada clase:

$$\mu_A(x) = \begin{cases} \frac{100x-9}{91} & \text{si } 0{,}09 \leq x \leq 1 \\ 0 & \text{en otro caso} \end{cases} \tag{3.5}$$

$$\mu_B(x) = \begin{cases} 10x & \text{si } 0 \leq x < 0{,}05 \\ 2 - 10x & \text{si } 0{,}05 \leq x \leq 0{,}1 \\ 0 & \text{en otro caso} \end{cases} \tag{3.6}$$

$$\mu_C(x) = \begin{cases} 1 - \frac{100x}{3} & \text{si } 0 \leq x \leq 0{,}03 \\ 0 & \text{en otro caso} \end{cases} \tag{3.7}$$

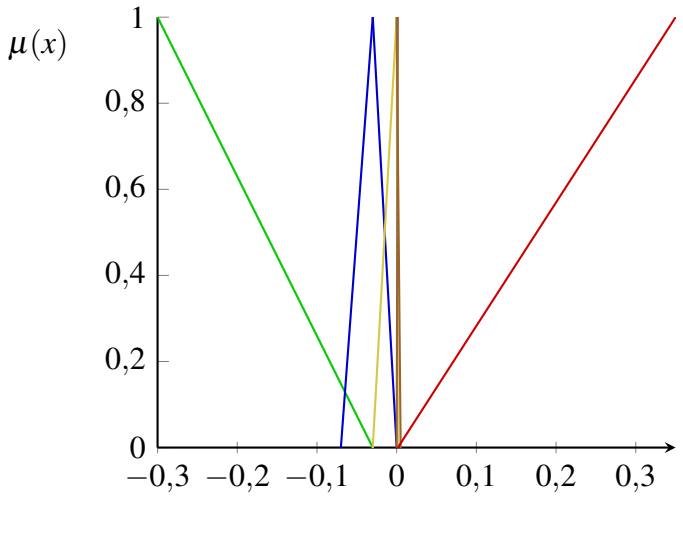
Donde la función 3.5 representa la clase ”muchos”, la función 3.6 representa la clase ”masomenos”, y la función 3.7 representa la clase ”pocos”.

Por ejemplo, en el total de imágenes que tiene la base de datos, se ha observado que los píxeles correspondientes al fondo representan un alto porcentaje, en este caso, el 95,34 % del total. Los píxeles asociados al hígado representan el 3,366 %, mientras que los píxeles que pertenecen

al riñón izquierdo, riñón derecho y bazo constituyen el 0,3782 %, 0,3745 % y 0,539 % respectivamente.

Para incorporar esta información al sistema difuso, se han definido tres variables difusas anteriormente mencionadas, para la primera entrada. La variable ”muchos”, se asigna a las clases cuya proporción de píxeles está en el rango de 1 a 0,09 (normalizado del 100 % al 9 %). En este caso, el fondo de la imagen se clasificaría como ”muchos”, debido a su alta presencia. La variable ”masomenos”, se utiliza para las clases cuyos valores están entre 0 y 0,1 (normalizado del 0 % al 10 %), lo que sería apropiado para el hígado, ya que su proporción está en ese rango. Por último, la variable ”pocos”, engloba a las clases con valores menores a 0,03 (normalizado al 3 %), lo que sería aplicable a las tres clases restantes: riñón izquierdo, riñón derecho y bazo, ya que tienen una presencia más reducida en la imagen.

La segunda entrada utilizada en el sistema difuso corresponde a la diferencia entre la pérdida obtenida en la época ”n”, y la pérdida obtenida en la época ”$n-1$”, como se muestra en la figura 3.17. Esta diferencia se ha categorizado en cinco etiquetas: ”muy negativa ”, para diferencias en el rango de $-0{,}3$ a $-0{,}03$ (aumento significativo en la pérdida), ”Negativa”, para diferencias en el rango de $-0{,}07$ a 0 (incremento en la pérdida, pero moderado), ”Neutra”, para diferencias en el rango de $-0{,}03$ a 0,0015 (diferencia insignificante), ”Positiva”, para diferencias en el rango de 0 a 0,005 (incremento leve en la pérdida) y ”muy positiva”, para diferencias en el rango de 0,0015 a 0,35 (disminución relevante en la pérdida).



Color	Descripción
	Diferencia ”muy negativa”
	Diferencia ”negativa”
	Diferencia ”neutro”
	Diferencia ”positiva”
	Diferencia ”muy positiva”

Figura 3.16: Función de membresía para la diferencia entre la perdida focal de la epoca n y la época $n-1$.

Se define la grafica 3.17 de acuerdo a las siguientes funciones para cada clase:

$$\mu_A(x) = \begin{cases} \frac{-100x-3}{27} & \text{si } -0{,}3 \leq x \leq -0{,}03 \\ 0 & \text{en otro caso} \end{cases} \tag{3.8}$$

$$\mu_B(x) = \begin{cases} \frac{100x+7}{4} & \text{si } -0{,}07 \leq x < -0{,}035 \\ \frac{-100x}{4} & \text{si } -0{,}035 \leq x \leq 0 \\ 0 & \text{en otro caso} \end{cases} \tag{3.9}$$

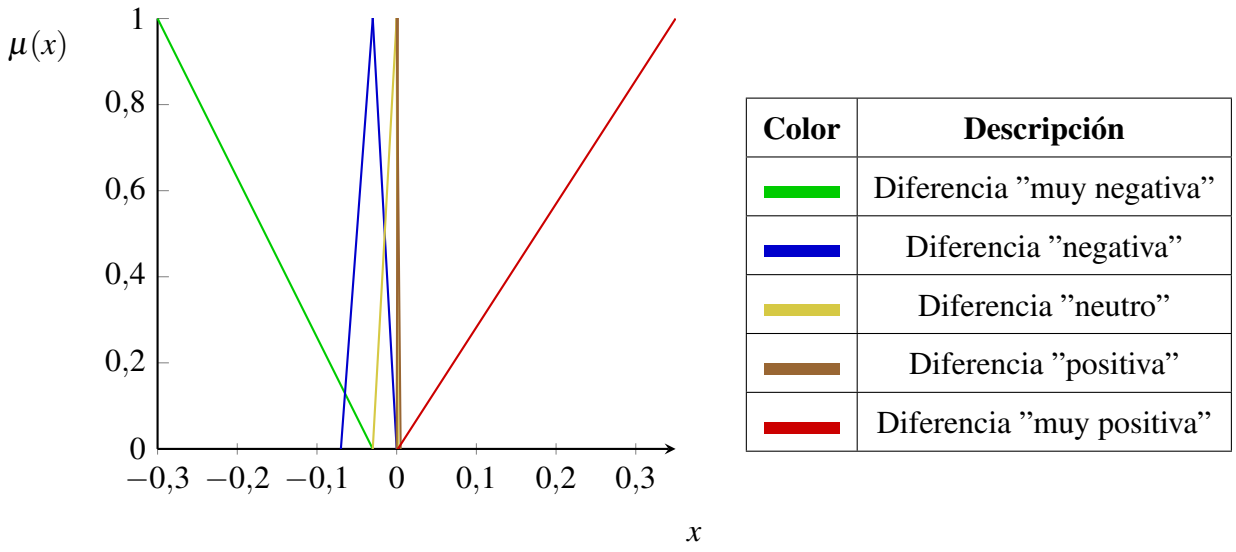
$$\mu_C(x) = \begin{cases} \frac{100x+3}{3} & \text{si } -0{,}03 \leq x < 0 \\ \frac{-2000x}{3} + 1 & \text{si } 0 \leq x \leq 0{,}0015 \\ 0 & \text{en otro caso} \end{cases} \tag{3.10}$$

$$\mu_D(x) = \begin{cases} 400x & \text{si } 0 \leq x < 0{,}0025 \\ -400x+2 & \text{si } 0{,}0025 \leq x \leq 0{,}005 \\ 0 & \text{en otro caso} \end{cases} \tag{3.11}$$

$$\mu_E(x) = \begin{cases} \frac{2000x-3}{697} & \text{si } 0{,}0015 \leq x \leq 0{,}35 \\ 0 & \text{en otro caso} \end{cases} \tag{3.12}$$

Donde la función 3.8 representa la diferencia ”muy negativa”, entre la función focal de la época n y la época $n-1$, la función 3.9 representa la diferencia ”Negativa”, la función 3.10 representa la diferencia ”Neutra”, la función 3.11 representa la diferencia ”Positiva”, y la función 3.12 representa la diferencia ”muy positiva”.

La Figura 3.18 presenta la gráfica con la salida de la fuzzyficación del sistema de control difuso, lo que permite visualizar la relación entre las etiquetas y los rangos de valores considerados. Esta representación gráfica contribuye a una comprensión más clara y concisa del proceso de ajuste del parámetro α y su influencia en la calidad de las segmentaciones obtenidas.



Color	Descripción
	Diferencia ”muy negativa”
	Diferencia ”negativa”
	Diferencia ”neutro”
	Diferencia ”positiva”
	Diferencia ”muy positiva”

Figura 3.17: Función de membresía para la diferencia entre la perdida focal de la epoca n y la época $n-1$.

Se define la grafica 3.18 de acuerdo a las siguientes funciones para cada clase:

$$\mu_A(x) = \begin{cases} \frac{-20x-1}{3} & \text{si } -0{,}2 \leq x \leq -0{,}05 \\ 0 & \text{en otro caso} \end{cases} \tag{3.13}$$

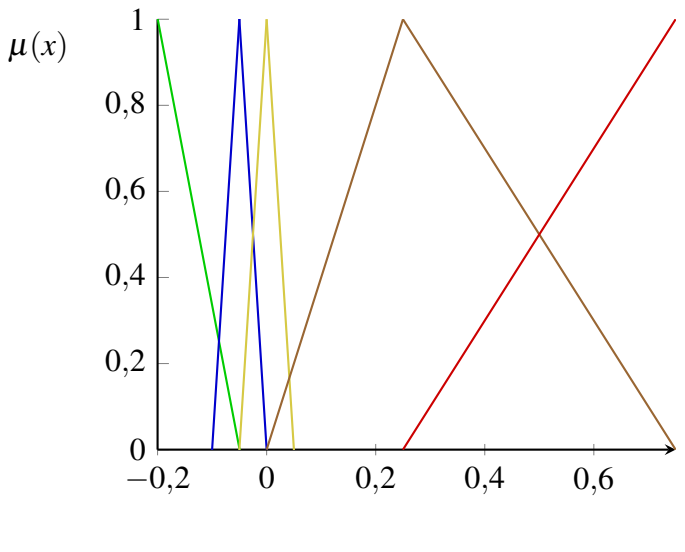
$$\mu_B(x) = \begin{cases} 20x+2 & \text{si } -0{,}1 \leq x < -0{,}05 \\ -20x & \text{si } -0{,}05 \leq x \leq 0 \\ 0 & \text{en otro caso} \end{cases} \tag{3.14}$$

$$\mu_C(x) = \begin{cases} 20x+1 & \text{si } -0{,}05 \leq x < 0 \\ -20x+1 & \text{si } 0 \leq x \leq 0{,}05 \\ 0 & \text{en otro caso} \end{cases} \tag{3.15}$$

$$\mu_D(x) = \begin{cases} 4x & \text{si } 0 \leq x < 0{,}25 \\ -x+1{,}5 & \text{si } 0{,}25 \leq x \leq 0{,}75 \\ 0 & \text{en otro caso} \end{cases} \tag{3.16}$$

$$\mu_E(x) = \begin{cases} 2x-0{,}5 & \text{si } 0{,}25 \leq x \leq 0{,}75 \\ 0 & \text{en otro caso} \end{cases} \tag{3.17}$$

Donde la función 3.14 representa el ajuste de α ”decrementar”, la función 3.14 representa el ajuste de α ”decrementar levemente”, la función 3.15 representa el ajuste de α ”nulo”, la función 3.16 representa el ajuste de α ïncrementar levemente”, y la función 3.17 representa el ajuste de α ïncrementar”.



Color	Descripción
	Ajuste de α “decrementar”
	Ajuste de α “decrementar levemente”
	Ajuste de α “nulo”
	Ajuste de α “incrementar levemente”
	Ajuste de α“incrementar”

Figura 3.18: Función de membresía de la salida para el ajuste del parámetro alfa en la función de perdida focal.

Las reglas utilizadas para el sistema de control difuso fueron las siguientes:

Tabla 3.1: Reglas difusas definidas para el sistema difuso del modelo ”FF-TransUnet”.

Reglas difusas
Si el tamaño de las clases es pocos **AND** *diferencia es* muy negativa **THEN** ajuste en α debe *ser nulo*.
Si el tamaño de las clases es pocos **AND** *diferencia es* negativa **THEN** ajuste en α debe *ser nulo*.
Si el tamaño de las clases es pocos **AND** *diferencia es* neutra **THEN** ajuste en α debe *incrementar levemente*.
Si el tamaño de las clases es pocos **AND** *diferencia es* positiva **THEN** ajuste en α debe *incrementar*.
Si el tamaño de las clases es pocos **AND** *diferencia es* muy positiva **THEN** ajuste en α debe *incrementar*.
Si el tamaño de las clases es masomenos **AND** *diferencia es* muy negativa **THEN** ajuste en α debe *ser nulo*.
Si el tamaño de las clases es masomenos **AND** *diferencia es* negativa **THEN** ajuste en α debe *ser nulo*.
Si el tamaño de las clases es masomenos **AND** *diferencia es* neutra **THEN** ajuste en α debe *ser nulo*.
Si el tamaño de las clases es masomenos **AND** *diferencia es* positiva **THEN** ajuste en α debe *ser nulo*.
Si el tamaño de las clases es masomenos **AND** *diferencia es* muy positiva **THEN** ajuste en α debe *incrementar levemente*.
Si el tamaño de las clases es muchos **AND** *diferencia es* muy negativa **THEN** ajuste en α debe *ser nulo*.
Si el tamaño de las clases es muchos **AND** *diferencia es* negativa **THEN** ajuste en α debe *ser nulo*.
Si el tamaño de las clases es muchos **AND** *diferencia es* neutra **THEN** ajuste en α debe *ser nulo*.
Si el tamaño de las clases es muchos **AND** *diferencia es* positiva **THEN** ajuste en α debe *decrementar levemente*.
Si el tamaño de las clases es muchos **AND** *diferencia es* muy positiva **THEN** ajuste en α debe *decrementar levemente*.

En donde un ajuste ”*nulo*”, significa que el parámetro alfa no se modificará, ya que la pérdida está bajando, que es el comportamiento que se espera. Mientras que un ajuste ”*incrementar levemente*”, y ”*incrementar*”, implica que el parámetro alfa se modificará para las clases que se encuentren en la variable ”pocos”, de manera que se aumente. Por otro lado, si el resultado es ”*decrementar levemente*”, es para ajustar el parámetro α en las clases que se encuentren en la

variable ”muchos”, de manera que se reduzca. Esto permite que alfa de las clases con pocos datos se aumente y, viceversa, en las clases con muchos datos, el parámetro alfa se reduzca.

Este sistema de control difuso permite adaptar de manera inteligente el parámetro α en cada época de entrenamiento, respondiendo a las variaciones en los tamaños de las clases y la evolución de la pérdida. La utilización de este enfoque novedoso tiene como objetivo optimizar el rendimiento del modelo ”FF-TransUnet”, y lograr segmentaciones más precisas y consistentes de los órganos abdominales en imágenes de resonancia magnética.

Este enfoque de actualización dinámica del parámetro α constituye una contribución significativa de esta investigación, ofreciendo una metodología avanzada y efectiva para mejorar los resultados de la segmentación en imágenes médicas y abriendo nuevas oportunidades para futuras investigaciones en el campo de la inteligencia artificial aplicada a la medicina.

CAPÍTULO 4. Resultados

En esta sección se presentan los resultados obtenidos en el estudio de investigación. Uno de los objetivos es comparar los modelos disponibles en el estado del arte que utilizan las base de datos CHAOS. Además de esto, se incluyen los resultados de arquitecturas que utilizan modelos de atención o *transformers*. Estos últimos se evaluaron en una base de datos diferente, que se reporta en la investigación correspondiente. Por último, se presentan los resultados de evaluar las base de datos CHAOS en los modelos implementados: *UNet*, *UNet++*, *Non-local UNet*, *Attention UNet*, *Ce-Net*, *TransUNet* y *MCTrans*.

4.1. Modelos que utilizan la base de datos CHAOS

En la Tabla 4.1 se presenta una variedad de modelos de aprendizaje profundo que se han desarrollado específicamente para la tarea de segmentación de imágenes médicas, y que han sido evaluados en la base de datos CHAOS. Esta base de datos ha sido utilizada en labores de investigación en este campo, ya que contiene variedad de imágenes médicas de alta calidad (MR y TC) que son difíciles de segmentar manualmente debido a su complejidad.

Los modelos presentados utilizan diferentes arquitecturas y técnicas de aprendizaje profundo para abordar el desafío de la segmentación de imágenes médicas en la base de datos. Algunos de ellos se basan en redes neuronales convolucionales profundas (CNN), mientras que otros utilizan técnicas de segmentación basadas en atención o redes adversarias generativas (GAN).

Además, algunos de los modelos presentados han logrado resultados sobresalientes en términos de precisión y eficiencia en la segmentación de imágenes médicas en la base de datos CHAOS. Estos modelos pueden ser útiles para mejorar la precisión y eficiencia en la práctica clínica, lo que puede tener un impacto significativo en el diagnóstico y tratamiento de diversas enfermedades y trastornos médicos.

Tabla 4.1: Comparación de los modelos que utilizan la base de datos CHAOS, evaluados con la métrica DICE.

Modelos	Año	DICE	Modalidad	Train-Val-Test	Param entrenables
DCACNet [11]	2021	91,03 %	RM T1 Dual - 4 órganos	60 % – 20 % – 20 %	+34M
FFANet [14]	2021	90,90 %	RM T1 Dual - 4 órganos	65 % – 10 % – 25 %	37,24 M
PRDNet [8]	2020	90,2 %	RM T1 Dual - 4 órganos	65 % – 10 % – 25 %	+34M
MS-GAN [88]	2020	86,75 %	RM T1 Dual - 4 órganos	65 % – 10 % – 25 %	43 – 82 M
UAGAN [90]	2020	84,03 %	RM T1 Dual - 4 órganos	65 % – 35 %	33,99M
CGAN [89]	2021	89,85 % – 97,87 %	RM T1 Dual - 4 órganos y CT - Hígado	50 % – 50 %	NA
Interactive GUI [93]	2022	89,95 %	RM T2 - 4 órganos	Test	NA
MIDeepSeg [10]	2021	96,93 %	RM (T1 y T2) y TC - Bazo y riñones	Validación	NA
CASA [92]	2021	84,9 %	MR y TC - Simultáneo	80 % – 20 %	NA
RMS-Unet [9]	2022	95,49 %	TC - Hígado	70 % – 15 % – 15 %	NA
Cycle-C GAN [91]	2020	88,9 %	TC - Hígado	70 % – 10 % – 20 %	NA

4.2. Resultados con modelos implementados

En el siguiente apartado se describen los resultados obtenidos al implementar siete modelos de aprendizaje profundo seleccionados de los últimos avances en el campo de la segmentación de imágenes médicas. Los modelos fueron evaluados utilizando la base de datos CHAOS, la cual contiene imágenes médicas de alta calidad. Cada experimento se entrenó durante 1000 épocas para las resonancias magnéticas y durante 700 épocas para las tomografías computarizadas, para asegurar la estabilidad y la precisión del modelo. Los datos se dividieron en 3 secciones: entrenamiento, validación y prueba, en los cuales se usaron las particiones de 70 %, 10 % y 20 %, respectivamente.

Para llevar a cabo la implementación de los modelos de aprendizaje profundo destinados a la segmentación de imágenes de resonancia magnética en esta tesis, se empleó un equipo de computación robusto y altamente especializado. El equipo utilizado cuenta con una memoria RAM de 32*GB*, la presencia de una tarjeta gráfica Nvidia TITAN RTX garantiza un rendimiento excepcional en tareas de entrenamiento de redes neuronales, aprovechando su capacidad de procesamiento y aceleración GPU.

El sistema operativo utilizado fue Ubuntu 21,10, conocido por su estabilidad y compatibilidad con herramientas de desarrollo de código abierto. Para implementar y ejecutar los modelos de

aprendizaje profundo, se utilizó Python 3,6+ como lenguaje de programación, aprovechando su amplia gama de bibliotecas y marcos de trabajo para la inteligencia artificial, en particular PyTorch 1,7,0.

La tarjeta madre $ROGSTRIXZ390-E$ proporciona una base sólida para el rendimiento del sistema. Además, el procesador Intel (R) Core $i9-9900K$, con sus múltiples núcleos y alto rendimiento, aseguró que los cálculos intensivos se completaran de manera eficiente y oportuna.

Además, se deben tener en cuenta los sufijos utilizados en los modelos denotados con $_{Res}$ y $_{VGG}$, indican los diferentes *backbones* empleados en cada uno de ellos. Específicamente, se utilizó la popular *ResNet34* y *VGG19* para evaluar el desempeño de los modelos. Estos sufijos permiten diferenciar claramente entre los modelos y proporcionan información adicional sobre el enfoque utilizado en cada uno de ellos para lograr una segmentación precisa de las imágenes médicas en la base de datos CHAOS.

4.2.1. *Resultados con la modalidad RM - T1 para* 4 *órganos*

En este apartado se muestran los resultados de la segmentación de la base de datos CHAOS, se muestran 3 tablas en las que se tienen los resultados de evaluar la base de datos en cada uno de los 14 modelos. En cada tabla se muestra los resultados de la métrica DICE, y el IoU para la segmentación de cada órgano: hígado, riñón izquierdo, riñón derecho y el bazo. Y en la tercera tabla muestra los resultados de la media de cada una de las dos métricas, junto con la época donde se obtuvo el mejor resultado.

Experimento en la modalidad RM - T1

Tabla 4.2: Modelos implementados, evaluados en la modalidad de RM - T1, para segmentación de 4 órganos con la métrica DICE. Cada órgano se remarca un color para él primer lugar, segundo lugar y tercer lugar.

Modelos	Batch Size	Tiempo (hrs)	$DICE_{higado}$	$DICE_{RIzq}$	$DICE_{RDer}$	$DICE_{Bazo}$
$MCTrans_{Res}$	16	5,4	94,4 %	88,65 %	90,62 %	90,97 %
$MCTrans_{VGG}$	32	4,6	94,05 %	88,45 %	90,41 %	90,59 %
$TransUNet_{Res}$	16	5,6	94,42 %	88,56 %	91,13 %	91,62 %
$TransUNet_{VGG}$	32	5	94,06 %	88,56 %	90,21 %	90,88 %
$UNet_{Res}$	32	4	93,37 %	88,82 %	90,86 %	88,69 %
$UNet_{VGG}$	64	3,5	94,01 %	88,85 %	89,47 %	89,34 %
$UNet{+}{+}_{Res}$	16	7	94,07 %	88,69 %	90,06 %	89,43 %
$UNet{+}{+}_{VGG}$	32	6,5	93,96 %	85,25 %	87,25 %	89,85 %
$AttUNet_{Res}$	32	3	92,67 %	87,25 %	87,42 %	85,99 %
$AttUNet_{VGG}$	32	3,25	92,23 %	79,85 %	83,63 %	86,35 %
$CeNet_{Res}$	32	2,75	93,91 %	86,48 %	89,54 %	86,09 %
$CeNet_{VGG}$	32	2,55	94,1 %	89,02 %	89,9 %	87,33 %
$NonLocal_{Res}$	32	3,4	93,76 %	88,45 %	88,43 %	89,25 %
$NonLocal_{VGG}$	32	3,2	93,31 %	87,21 %	89,13 %	90,23 %

Tabla 4.3: Modelos implementados, evaluados en la modalidad de RM - T1, para segmentación de 4 órganos con la métrica IoU. Cada órgano se remarca un color para él primer lugar, segundo lugar y tercer lugar.

Modelos	Batch Size	Tiempo (hrs)	IoU_{higado}	IoU_{RIzq}	IoU_{RDer}	IoU_{Bazo}
$MCTrans_{Res}$	16	5,4	89,39 %	79,61 %	82,84 %	83,44 %
$MCTrans_{VGG}$	32	4,6	88,77 %	79,29 %	82,49 %	82,8 %
$TransUNet_{Res}$	16	5,6	89,43 %	79,47 %	83,7 %	84,53 %
$TransUNet_{VGG}$	32	5	88,79 %	79,48 %	82,17 %	83,28 %
$UNet_{Res}$	32	4	87,56 %	79,73 %	83,25 %	79,68 %
$UNet_{VGG}$	64	3,5	88,7 %	79,94 %	80,95 %	80,73 %
$UNet++_{Res}$	16	7	88,8 %	79,67 %	81,91 %	80,87 %
$UNet++_{VGG}$	32	6,5	88,61 %	74,29 %	77,39 %	81,57 %
$AttUNet_{Res}$	32	3	86,35 %	77,39 %	77,65 %	75,43 %
$AttUNet_{VGG}$	32	3,25	85,57 %	66,45 %	71,87 %	75,98 %
$CeNet_{Res}$	32	2,75	88,53 %	76,18 %	81,06 %	75,57 %
$CeNet_{VGG}$	32	2,55	88,85 %	80,21 %	81,66 %	77,51 %
$NonLocal_{Res}$	32	3,4	88,26 %	79,29 %	79,26 %	80,59 %
$NonLocal_{VGG}$	32	3,2	87,46 %	77,31 %	80,4 %	82,2 %

Tabla 4.4: Modelos implementados, evaluados en la modalidad de RM - T1, para segmentación de 4 órganos con la métrica DICE e IoU. La media de cada métrica se colorea para él primer lugar, segundo lugar y tercer lugar.

Modelos	Mejor época	$DICE_{mean}$	IoU_{mean}	# parámetros
$MCTrans_{Res}$	602	91,16 %	83,82 %	25,865,484
$MCTrans_{VGG}$	471	90,88 %	83,34 %	9,193,580
$TransUNet_{Res}$	280	91,43 %	84,28 %	34,245,958
$TransUNet_{VGG}$	458	90,93 %	83,43 %	17,574,054
$UNet_{Res}$	270	90,41 %	82,55 %	24,524,614
$UNet_{VGG}$	302	90,42 %	82,58 %	7,852,710
$UNet++_{Res}$	514	90,56 %	82,81 %	26,223,046
$UNet++_{VGG}$	479	89,08 %	80,46 %	9,163,494
$AttUNet_{Res}$	296	88,33 %	79,2 %	25,055,634
$AttUNet_{VGG}$	112	85,51 %	74,97 %	8,381,682
$CeNet_{Res}$	112	89 %	80,33 %	31,876,423
$CeNet_{VGG}$	296	90,09 %	82,06 %	15,738,655
$NonLocal_{Res}$	438	89,93 %	81,85 %	25,051,206
$NonLocal_{VGG}$	536	89,97 %	81,84 %	8,379,302

Con estos resultados se puede observar que los primeros 7 modelos tienen resultados superiores a 90,4 % en la media del coeficiente DICE, por lo que para siguientes experimentos únicamente se utilizarán estos 7 modelos de segmentación.

Experimentos

En esta subsección se muestran los resultados de realizar 3 corridas de experimentos con cada uno de los 7 mejores modelos de la subsección anterior.

Tabla 4.5: Los 7 mejores modelos implementados, evaluados en la modalidad de RM - T1, para segmentación de 4 órganos con la métrica DICE. Cada órgano se remarca un color para él primer lugar, segundo lugar y tercer lugar.

No.	Modelos	Batch Size	Tiempo (hrs)	$DICE_{higado}$	$DICE_{RIzq}$	$DICE_{RDer}$	$DICE_{Bazo}$
1	$MCTrans_{Res}$	16	4,8	94,3 %	90,7 %	90,01 %	89,31 %
2	$MCTrans_{Res}$	16	4,8	94,26 %	90,02 %	90,34 %	89,88 %
3	$MCTrans_{Res}$	16	4,8	94,46 %	90,13 %	91,05 %	90,24 %
1	$MCTrans_{VGG}$	32	4	94 %	88,72 %	90,38 %	90,04 %
2	$MCTrans_{VGG}$	32	4	94,1 %	88,52 %	90,26 %	90,42 %
3	$MCTrans_{VGG}$	32	4	94,22 %	88,65 %	90,59 %	90,18 %
1	$TransUNet_{Res}$	16	5	94,81 %	88,9 %	91,05 %	91,08 %
2	$TransUNet_{Res}$	16	5	95,06 %	88,96 %	90,95 %	91,56 %
3	$TransUNet_{Res}$	16	5	94,76 %	89,2 %	91,11 %	91,59 %
1	$TransUNet_{VGG}$	32	4,5	94,1 %	89,02 %	90,21 %	90,34 %
2	$TransUNet_{VGG}$	32	4,5	94,32 %	88,98 %	90,46 %	90,76 %
3	$TransUNet_{VGG}$	32	4,5	94,25 %	89,12 %	90,53 %	90,15 %
1	$UNet_{Res}$	32	3,5	93,42 %	88,85 %	90,5 %	88,9 %
2	$UNet_{Res}$	32	3,5	92,92 %	88,35 %	90,05 %	89,64 %
3	$UNet_{Res}$	32	3,5	94,05 %	88,85 %	90,16 %	89,02 %
1	$UNet_{VGG}$	64	3	94,1 %	88,9 %	89,4 %	89,34 %
2	$UNet_{VGG}$	64	3	92,98 %	88,5 %	89,95 %	89,41 %
3	$UNet_{VGG}$	64	3	94,13 %	88,62 %	90,01 %	89,16 %
1	$UNet{+}{+}_{Res}$	16	7	94,26 %	88,79 %	89,21 %	89,51 %
2	$UNet{+}{+}_{Res}$	16	7	94,21 %	88,72 %	89,88 %	89,24 %
3	$UNet{+}{+}_{Res}$	16	7	94,36 %	88,82 %	90,24 %	89,89 %

Tabla 4.6: Los 7 mejores modelos implementados, evaluados en la modalidad de RM - T1, para segmentación de 4 órganos con la métrica IoU. Cada órgano se remarca un color para él primer lugar, segundo lugar y tercer lugar.

No.	Modelos	Batch Size	Tiempo (hrs)	IoU_{higado}	IoU_{RIzq}	IoU_{RDer}	IoU_{Bazo}
1	$MCTrans_{Res}$	16	4,8	90,03 %	83,7 %	81,44 %	81,08 %
2	$MCTrans_{Res}$	16	4,8	90,89 %	86,1 %	80,55 %	80,59 %
3	$MCTrans_{Res}$	16	4,8	91,09 %	86,21 %	81,26 %	80,95 %
1	$MCTrans_{VGG}$	32	4	89,73 %	81,72 %	81,81 %	81,81 %
2	$MCTrans_{VGG}$	32	4	91,08 %	84,8 %	80,68 %	81,53 %
3	$MCTrans_{VGG}$	32	4	91,2 %	84,93 %	81,01 %	81,29 %
1	$TransUNet_{Res}$	16	5	90,54 %	81,9 %	82,48 %	82,85 %
2	$TransUNet_{Res}$	16	5	91,67 %	85,13 %	80,65 %	81,67 %
3	$TransUNet_{Res}$	16	5	91,37 %	85,37 %	80,81 %	81,7 %
1	$TransUNet_{VGG}$	32	4,5	89,83 %	82,02 %	81,64 %	82,11 %
2	$TransUNet_{VGG}$	32	4,5	91,29 %	85,15 %	81,08 %	81,57 %
3	$TransUNet_{VGG}$	32	4,5	91,22 %	85,29 %	81,15 %	80,96 %
1	$UNet_{Res}$	32	3,5	89,15 %	81,85 %	81,93 %	80,67 %
2	$UNet_{Res}$	32	3,5	90,58 %	84,36 %	80,02 %	82,65 %
3	$UNet_{Res}$	32	3,5	91,71 %	84,89 %	80,13 %	82,03 %
1	$UNet_{VGG}$	64	3	89,83 %	81,9 %	80,83 %	81,11 %
2	$UNet_{VGG}$	64	3	90 %	84,38 %	81,31 %	81,82 %
3	$UNet_{VGG}$	64	3	91,15 %	84,5 %	81,37 %	81,57 %
1	$UNet{+}{+}_{Res}$	16	7	89,99 %	81,79 %	80,64 %	81,28 %
2	$UNet{+}{+}_{Res}$	16	7	91,17 %	84,76 %	80,65 %	81,55 %
3	$UNet{+}{+}_{Res}$	16	7	91,32 %	84,86 %	81,01 %	82,2 %

Tabla 4.7: Los 7 mejores modelos implementados, evaluados en la modalidad de RM - T1, para segmentación de 4 órganos con la media de la métrica DICE e IoU. Cada órgano se remarca un color para él primer lugar, segundo lugar y tercer lugar.

No.	Modelos	Mejor época	Tiempo (min)	$DICE_{mean}$	IoU_{mean}	# parámetros
1	$MCTrans_{Res}$	308	62,37	91,08 %	84,06 %	25865484
2	$MCTrans_{Res}$	483	97,80	91,12 %	84,53 %	25865484
3	$MCTrans_{Res}$	494	100,03	91,47 %	84,87 %	25865484
1	$MCTrans_{VGG}$	467	80,55	90,78 %	83,76 %	9193580
2	$MCTrans_{VGG}$	257	44,33	90,82 %	84,52 %	9193580
3	$MCTrans_{VGG}$	458	79	90,91 %	84,60 %	9193580
1	$TransUNet_{Res}$	360	75,6	91,46 %	84,44 %	34245958
2	$TransUNet_{Res}$	460	96,6	91,63 %	84,78 %	34245958
3	$TransUNet_{Res}$	364	76,44	91,66 %	84,81 %	34245958
1	$TransUNet_{VGG}$	405	75,93	90,91 %	83,9 %	17574054
2	$TransUNet_{VGG}$	416	78	91,13 %	84,77 %	17574054
3	$TransUNet_{VGG}$	316	59,25	91,01 %	84,65 %	17574054
1	$UNet_{Res}$	268	40,2	90,41 %	83,4 %	24524614
2	$UNet_{Res}$	388	58,2	90,24 %	84,40 %	24524614
3	$UNet_{Res}$	418	62,7	90,52 %	84,68 %	24524614
1	$UNet_{VGG}$	390	51,18	90,43 %	83,41 %	7852710
2	$UNet_{VGG}$	447	58,66	90,21 %	84,37 %	7852710
3	$UNet_{VGG}$	295	38,71	90,48 %	84,64 %	7852710
1	$UNet{+}{+}_{Res}$	367	96,33	90,44 %	83,42 %	26223046
2	$UNet{+}{+}_{Res}$	263	69,03	90,51 %	84,53 %	26223046
3	$UNet{+}{+}_{Res}$	428	112,35	90,82 %	84,84 %	26223046

4.2.2. *Resultados de las pruebas estadísticas*

Prueba de *Mann-Whitney U*

De entre los 14 modelos presentados en los resultados, se seleccionaron los mejores 7 del primer experimento, con los cuales se realizaron 3 experimentos en total para la modalidad de

RM-T1, segmentando 4 órganos.

Los 7 modelos seleccionados son:

- *MCTrans* - *ResNet* (Modelo 1)
- *MCTrans* - *VGG* (Modelo 2)
- *TransUNet* - *ResNet* (Modelo 3)
- *TransUNet* - *VGG* (Modelo 4)
- *UNet* - *ResNet* (Modelo 5)
- *UNet* - *VGG* (Modelo 6)
- *UNet++* - *ResNet* (Modelo 7)

En este estudio, se compararon los siete modelos de segmentación de imágenes utilizando el rendimiento del coeficiente DICE en tres experimentos diferentes. Con el objetivo de evaluar si existen diferencias significativas en el rendimiento entre los modelos, se aplicó la prueba no paramétrica de *Mann-Whitney U*, que es una versión del test de *Wilcoxon* para muestras independientes. La prueba de *Mann-Whitney U* es apropiada en este caso porque no se asume una distribución normal de los datos y se comparan dos grupos de muestras independientes (la precisión de dos modelos diferentes en este caso).

Los resultados de la prueba de *Mann-Whitney U* para cada par de modelos se presentan en la Tabla 4.8.

Tabla 4.8: Resultados de la prueba de *Mann-Whitney U* para cada par de modelos. En rojo se resaltan escenarios en donde ocurren diferencias significativas.

	MCT RN	MCT vgg	TUnet RN	TUnet vgg	UNet RN	UNet vgg	UNet++
MCT RN		0,1	0,2	0,4	0,1	0,1	0,1
MCT vgg	0,1		0,1	0,121	0,1	0,1	0,268
TUnet RN	0,2	0,1		0,1	0,1	0,1	0,1
TUnet vgg	0,4	0,121	0,1		0,1	0,1	0,1
UNet RN	0,1	0,1	0,1	0,1		1	0,4
UNet vgg	0,1	0,1	0,1	0,1	1		0,2
UNet++	0,1	0,268	0,1	0,1	0,4	0,2	

De acuerdo con los resultados de la prueba de *Mann-Whitney U*, se encontraron diferencias significativas en el rendimiento del coeficiente DICE entre los siguientes pares de modelos:

- El modelo *MCTrans - ResNet* con los modelos: *MCTrans - VGG, UNet -ResNet, UNet -VGG* y *UNet ++*.
- El modelo *MCTrans - VGG* con los modelos: *MCTrans - ResNet, TransUNet - ResNet, UNet -ResNet* y *UNet - VGG*.
- El modelo *TransUNet - ResNet* con los modelos: *MCTrans - VGG, TransUNet - VGG, UNet -ResNet, UNet - VGG* y *UNet ++*.
- El modelo *TransUNet - VGG* con los modelos: *TransUNet - ResNet, UNet -ResNet, UNet - VGG* y *UNet ++*.

Los resultados indican que los modelos: *UNet - ResNet, UNet - VGG* y *UNet++* no muestran diferencias significativas en cuanto al coeficiente DICE en comparación con los demás modelos. Mientras que el modelo *TransUNet - ResNet* muestra diferencias significativas con todos los modelos, excepto con el modelo *MCTrans - ResNet*. Sin embargo, es importante señalar que la prueba *Mann-Whitney U* solo identifica si hay una diferencia significativa entre los modelos, pero no proporciona información sobre qué modelo es el mejor. Además, el coeficiente DICE es solo

una medida de rendimiento, y también deben tenerse en cuenta otras métricas de evaluación de la segmentación semántica a la hora de seleccionar el modelo más adecuado para una determinada tarea de segmentación de imágenes.

Prueba de *Friedman* y comparaciones *post hoc*

En este estudio, se compararon siete modelos de segmentación de imágenes utilizando el rendimiento del coeficiente DICE en tres experimentos diferentes. Con el objetivo de evaluar si existen diferencias significativas en el rendimiento del coeficiente DICE entre los modelos, se aplicó la prueba no paramétrica de *Friedman*, que es apropiada para comparar múltiples modelos en datos de medidas repetidas. La prueba de *Friedman* no asume una distribución normal de los datos y es adecuada para comparar múltiples tratamientos o modelos en el mismo conjunto de sujetos o experimentos.

Los resultados de la prueba de *Friedman* para la comparación de la precisión de los modelos en los tres experimentos son los siguientes:

Estadístico de *Friedman*: 17,42857142, p-valor: 0,0078307

Dado que el p-valor es menor que el nivel de significancia (usualmente 0.05), se puede concluir que hay diferencias significativas en el coeficiente DICE entre los modelos.

Para identificar qué pares de modelos tienen diferencias significativas en su rendimiento, se realizaron comparaciones *post hoc*. Los resultados de las comparaciones *post hoc* se presentan a continuación en la tabla 4.9:

Tabla 4.9: Comparaciones *post hoc*.

	MCT RN	MCT vgg	TUnet RN	TUnet vgg	UNet RN	UNet vgg	UNet++
MCT RN	1.000000	0.900000	0.900000	0.900000	0.259680	0.175084	0.900000
MCT vgg	0.900000	1.000000	0.599380	0.900000	0.821633	0.710506	0.900000
TUnet RN	0.900000	0.599380	1.000000	0.900000	0.040135	0.022344	0.259680
TUnet vgg	0.900000	0.900000	0.900000	1.000000	0.366292	0.259680	0.821633
UNet RN	0.259680	0.821633	0.040135	0.366292	1.000000	0.900000	0.900000
UNet vgg	0.175084	0.710506	0.022344	0.259680	0.900000	1.000000	0.900000
UNet++	0.710506	0.900000	0.259680	0.821633	0.900000	0.900000	1.000000

De acuerdo con los resultados de la prueba de *Friedman* y las comparaciones *post hoc*, se encontraron diferencias significativas en el rendimiento del coeficiente DICE entre los siguientes pares de modelos:

- El modelo *TransUNet - ResNet* con los modelos: *UNet -ResNet* y *UNet - VGG*.
- El modelo *UNet - ResNet* con el modelo: *TransUNet - ResNet*.
- El modelo *UNet - VGG* con el modelo: *TransUNet - ResNet*.

Estos resultados sugieren que el modelo: *TransUNet - ResNet* tiene un rendimiento significativamente diferente en términos del coeficiente DICE en comparación con otros modelos. Dentro de los modelos, podemos incluir a *MCTrans - VGG, TransUNet - VGG* y *MCTrans - ResNet*. Estos no muestran diferencias significativas, respecto a los otros modelos, pero analizando los resultados en la métrica DICE, podemos observar que *UNet - VGG, UNet - ResNet* tienen resultados inferiores a estos modelos, por lo que, podemos excluirlos de los mejores modelos.

Resultados con el Criterio de Información Akaike (AIC) y Criterio de Información Bayesiano (BIC)

En este estudio, se evaluaron varios modelos de segmentación con el objetivo de identificar el mejor en términos de rendimiento y complejidad. Para seleccionar el modelo óptimo, se utilizaron dos criterios de información: el Criterio de Información de Akaike (AIC) y el Criterio

de Información Bayesiano (BIC) [94] [95]. Ambos criterios tienen en cuenta tanto la bondad de ajuste del modelo como la complejidad del mismo, penalizando modelos con un mayor número de parámetros. Aunque ambos criterios tienen propósitos similares, el BIC tiende a ser más riguroso en la penalización de modelos más complejos que el AIC.

En la Tabla 4.10, se presentan los valores de AIC y BIC para cada modelo evaluado. Basándonos en estos criterios, se seleccionó el modelo MCTrans - VGG como el mejor modelo para la segmentación, ya que obtuvo los valores más bajos tanto en AIC como en BIC.

Tabla 4.10: Valores de AIC y BIC para cada modelo

Modelo	AIC	BIC	Mejor según
MCTrans - ResNet	51730968.1837	161558574.7885	
MCTrans - VGG	18387160.1922	57424082.4292	AIC, BIC
TransUNet - ResNet	68491916.1758	213903910.1300	
TransUNet - VGG	35148108.1882	109769417.7745	
UNet++ - ResNet	52446092.1976	163791945.2297	

4.2.3. *Análisis de resultados*

En función de los datos presentados en la Tabla de resultados 4.4, se observa que el modelo *TransUNet - ResNet* exhibe un rendimiento superior en términos de la métrica DICE. Sin embargo, las pruebas estadísticas indican que las diferencias significativas son más pronunciadas entre los modelos *MCTrans - ResNet* y *TransUNet - ResNet*.

En relación con los valores de los criterios de información AIC y BIC, se destaca el modelo *MCTrans - VGG* como la elección óptima. Esta elección se basa en la consideración del número de parámetros del modelo, siendo estos penalizados más fuertemente a medida que aumentan, y la métrica DICE como variables. Si nos enfocamos únicamente en estos criterios, el modelo *MCTrans - VGG* se establece como la mejor opción en términos de los valores AIC y BIC.

No obstante, si ampliamos nuestra perspectiva para incluir el rendimiento en la métrica

DICE y los resultados del análisis estadístico, emerge un panorama diferente. En este contexto, el modelo *TransUNet - ResNet* emergería como la opción preferida.

4.3. Resultados con el modelo propuesto: FF-TransUnet

En esta sección de resultados finales, se presentan las comparaciones entre los cuatro mejores modelos seleccionados de las secciones anteriores: "*TransUnet - ResNet*", "*TransUnet - VGG*", "*MCTrans - ResNet*", y "*MCTrans - VGG*", junto con los resultados obtenidos por el modelo propuesto "*FF-TransUnet*". Todos los modelos fueron evaluados en la modalidad de segmentación de los cuatro órganos abdominales (hígado, bazo, riñón izquierdo y derecho) a lo largo de 1200 épocas, en condiciones iguales de entrenamiento.

En particular, los cuatro modelos seleccionados fueron previamente identificados como los más prometedores en términos de desempeño. La comparación con el modelo propuesto "*FF-TransUnet*", permitirá evaluar la efectividad y robustez del enfoque propuesto en relación con las técnicas de vanguardia.

Además de los resultados cuantitativos, se realizará un análisis estadístico exhaustivo, similar al aplicado en las subsecciones anteriores, para garantizar la validez y la confiabilidad de las conclusiones extraídas. Esta evaluación estadística permitirá identificar patrones, tendencias y diferencias significativas entre los modelos, respaldando aún más la elección del modelo más adecuado para la segmentación de órganos abdominales en imágenes de resonancia magnética.

En conjunto, esta sección brindará una visión completa y rigurosa de la comparativa entre los modelos, proporcionando información valiosa para la toma de decisiones sobre la elección del enfoque más eficaz y prometedor para la tarea de segmentación de órganos abdominales en imágenes médicas.

En las siguientes tablas se presentan los resultados obtenidos del modelo propuesto "FF-TransUnet", en diez corridas de experimentos, en los cuales se realizaron modificaciones en los parámetros clave, como el tamaño del lote (*batch size* o BS), el número de bloques de transforma-

dores (TB), el número de cabezas (HN) y el número de neuronas en la capa *Multi-Layer Perceptron* (MLP).

La Tabla 4.11 muestra los resultados obtenidos por órgano utilizando la métrica de evaluación DICE. Cada fila corresponde a un órgano específico (hígado, bazo, riñón izquierdo y derecho) y cada columna representa un experimento con los parámetros mencionados anteriormente. Los valores presentados en esta tabla son la medida del coeficiente DICE para cada órgano y experimento, lo que permitirá una evaluación detallada del desempeño del modelo en la segmentación de cada estructura.

Tabla 4.11: Resultados por órgano del modelo propuesto: FF-TransUnet, evaluados en la modalidad de RM - T1, para segmentación de 4 órganos con la métrica DICE. Se muestran los parámetros que influyen en el número de parámetros entrenables como lo son: bloques de transformadores (TB), capa *multi-layer perceptron* (MLP) y número de cabezas en el transformador (HN). Cada órgano se remarca un color para él primer lugar, segundo lugar y tercer lugar.

Modelo	BS	TB	MLP	HN	P. Entrenables	$DICE_{higado}$	$DICE_{RIzq}$	$DICE_{RDer}$	$DICE_{Bazo}$
FF-TransUnet	8	2	256	16	34539189	96,06 %	86,94 %	89,05 %	92,91 %
FF-TransUnet	8	2	256	16	34539189	96,15 %	89,18 %	88,9 %	92,88 %
FF-TransUnet	8	2	256	16	34539189	95,96 %	87,77 %	88,21 %	92,72 %
FF-TransUnet	8	4	256	32	43987125	95,55 %	88,79 %	87 %	90,39 %
FF-TransUnet	8	1	64	2	29421813	95,94 %	88,41 %	88,74 %	90,83 %
FF-TransUnet	8	1	64	64	29421813	95,78 %	87,08 %	88,77 %	92,86 %
FF-TransUnet	8	1	128	256	29552949	95,86 %	87,8 %	89,26 %	92,06 %
FF-TransUnet	8	2	64	16	33752373	95,26 %	87,37 %	86,55 %	90 %
FF-TransUnet	8	2	64	64	33752373	95,61 %	89,03 %	88,65 %	93,27 %
FF-TransUnet	8	4	64	128	42413493	96,09 %	88,03 %	88,87 %	93,53 %

Asimismo, la Tabla 4.12 proporciona información similar a la Tabla 4.11, pero utilizando la métrica de evaluación IoU. De manera similar, los valores presentados en esta tabla representan los resultados obtenidos en cada experimento para cada órgano específico, permitiendo un análisis más completo del rendimiento del modelo en términos de superposición y concordancia entre las segmentaciones predichas y las segmentaciones de referencia.

Tabla 4.12: Resultados por órgano del modelo propuesto: FF-TransUnet, evaluados en la modalidad de RM - T1, para segmentación de 4 órganos con la métrica IoU. Se muestran los parámetros que influyen en el número de parámetros entrenables como lo son: bloques de transformadores (TB), capa *multi-layer perceptron* (MLP) y número de cabezas en el transformador (HN). Cada órgano se remarca un color para él primer lugar, segundo lugar y tercer lugar.

Modelo	BS	TB	MLP	HN	P. entrenables	IoU_{higado}	IoU_{RIzq}	IoU_{RDer}	IoU_{Bazo}
FF-TransUnet	8	2	256	16	34539189	92,42 %	76,9 %	80,27 %	86,76 %
FF-TransUnet	8	2	256	16	34539189	92,6 %	80,48 %	80,02 %	86,72 %
FF-TransUnet	8	2	256	16	34539189	92,34 %	77,88 %	80,1 %	85,64 %
FF-TransUnet	8	4	256	32	43987125	91,48 %	79,85 %	77 %	82,48 %
FF-TransUnet	8	1	64	2	29421813	92,21 %	79,23 %	79,76 %	83,2 %
FF-TransUnet	8	1	64	64	29421813	91,91 %	77,12 %	79,8 %	86,67 %
FF-TransUnet	8	1	128	256	29552949	92,05 %	78,41 %	80,61 %	85,3 %
FF-TransUnet	8	2	64	16	33752373	90,95 %	77,58 %	76,29 %	81,82 %
FF-TransUnet	8	2	64	64	33752373	91,59 %	80,23 %	79,61 %	87,4 %
FF-TransUnet	8	4	64	128	42413493	92,48 %	78,63 %	79,97 %	87,85 %

Finalmente, en la Tabla 4.13 se presentan los resultados promedio de ambas métricas (DICE e IoU) para los diez experimentos realizados. Esta tabla ofrece una visión general del rendimiento global del modelo propuesto en las diferentes configuraciones de parámetros. Los promedios permiten una evaluación más generalizada de la capacidad del modelo para la segmentación precisa de los órganos abdominales.

Tabla 4.13: Resultados del modelo propuesto: FF-TransUnet, evaluados en la modalidad de RM - T1, con la media de la métrica DICE e IoU. Se muestran los parámetros que influyen en el número de parámetros entrenables como lo son: bloques de transformadores (TB), capa *multi-layer perceptron* (MLP) y número de cabezas en el transformador (HN). La media de cada métrica se colorea para él primer lugar, segundo lugar y tercer lugar.

Modelo	BS	TB	MLP	HN	Tiempo (min)	Mejor época	$DICE_{mean}$	IoU_{mean}
FF-TransUnet	8	2	256	16	28,83	124	91,24 %	84,08 %
FF-TransUnet	8	2	256	16	20,69	89	91,77 %	84,95 %
FF-TransUnet	8	2	256	16	22,32	96	91,16 %	83,99 %
FF-TransUnet	8	4	256	32	22,27	90	90,43 %	82,7 %
FF-TransUnet	8	1	64	2	29,28	137	90,98 %	83,6 %
FF-TransUnet	8	1	64	64	24,79	116	91,12 %	83,87 %
FF-TransUnet	8	1	128	256	30,82	137	91,24 %	84,09 %
FF-TransUnet	8	2	64	16	14,94	65	89,79 %	81,66 %
FF-TransUnet	8	2	64	64	33,1	144	91,64 %	84,7 %
FF-TransUnet	8	4	64	128	43,53	172	91,63 %	84,73 %

Estas tablas son esenciales para la comprensión y evaluación de los resultados obtenidos, proporcionando evidencia sólida y cuantificable del desempeño del modelo propuesto “FF-TransUnet”, en diversas condiciones de experimentación. Los análisis detallados derivados de estas tablas contribuirán significativamente a las conclusiones y recomendaciones finales de la investigación.

4.3.1. *Comparación de los resultados con el modelo propuesto: FF-TransUnet*

En la presente sección, se presenta una tabla comparativa con los resultados promedio de las métricas IoU y DICE, para los modelos implementados y para el modelo propuesto “FF-TransUnet”, las cuales fueron evaluadas a lo largo de las 1200 épocas de entrenamiento.

Tabla 4.14: Los 7 mejores modelos implementados, evaluados en la modalidad de RM - T1, para segmentación de 4 órganos con la media de la métrica DICE e IoU. Cada órgano se remarca un color para él primer lugar, segundo lugar y tercer lugar.

No.	Modelos	Mejor época	Tiempo (min)	$DICE_{mean}$	IoU_{mean}	# parámetros
1	$MCTrans_{Res}$	308	62,37	91,08 %	84,06 %	25865484
2	$MCTrans_{Res}$	483	97,80	91,12 %	84,53 %	25865484
3	$MCTrans_{Res}$	494	100,03	91,47 %	84,87 %	25865484
1	$MCTrans_{VGG}$	467	80,55	90,78 %	83,76 %	9193580
2	$MCTrans_{VGG}$	257	44,33	90,82 %	84,52 %	9193580
3	$MCTrans_{VGG}$	458	79	90,91 %	84,60 %	9193580
1	$TransUNet_{Res}$	360	75,6	91,46 %	84,44 %	34245958
2	$TransUNet_{Res}$	460	96,6	91,63 %	84,78 %	34245958
3	$TransUNet_{Res}$	364	76,44	91,66 %	84,81 %	34245958
1	$TransUNet_{VGG}$	405	75,93	90,91 %	83,9 %	17574054
2	$TransUNet_{VGG}$	416	78	91,13 %	84,77 %	17574054
3	$TransUNet_{VGG}$	316	59,25	91,01 %	84,65 %	17574054
1	$UNet_{Res}$	268	40,2	90,41 %	83,4 %	24524614
2	$UNet_{Res}$	388	58,2	90,24 %	84,40 %	24524614
3	$UNet_{Res}$	418	62,7	90,52 %	84,68 %	24524614
1	$UNet_{VGG}$	390	51,18	90,43 %	83,41 %	7852710
2	$UNet_{VGG}$	447	58,66	90,21 %	84,37 %	7852710
3	$UNet_{VGG}$	295	38,71	90,48 %	84,64 %	7852710
1	$UNet{+}{+}_{Res}$	367	96,33	90,44 %	83,42 %	26223046
2	$UNet{+}{+}_{Res}$	263	69,03	90,51 %	84,53 %	26223046
3	$UNet{+}{+}_{Res}$	428	112,35	90,82 %	84,84 %	26223046
1	FF-TransUnet	89	20,69	91,77 %	84,95 %	34539189
2	FF-TransUnet	124	28,83	91,24 %	84,08 %	34539189
3	FF-TransUnet	96	22,32	91,16 %	83,99 %	34539189

Con respecto a la Tabla 4.14, se reflejan mejores resultados en la métrica DICE, con un aumento del 0,11 %, entre el mejor modelo en el estado del arte siendo *TrasUNet - ResNet* y el mejor

resultado del modelo propuesto *FF-TransUnet*. De manera similar, en la métrica IoU, se tiene un incremento del 0,08 % entre el mejor modelo en el estado del arte siendo *MCTrans - ResNet* y el mejor resultado del modelo propuesto *FF-TransUnet*.

Adicionalmente, podemos agregar al análisis los tiempos en los que cada modelo (cuarta columna de la tabla 4.14) obtuvo el mejor resultado en la métrica DICE. Podemos ver que el mejor tiempo en el modelo propuesto fue de $20,69$ minutos, mientras que el mejor tiempo para el mejor modelo propuesto (el modelo *TrasUNet - ResNet*), fue de $76,44$ minutos, dando una diferencia de $55,75$ minutos.

Los hallazgos presentados en esta tabla ponen de manifiesto el impacto positivo y significativo del enfoque propuesto. Estos incrementos en las métricas de evaluación y la disminución en los tiempos para obtener los mejores resultados, señalan la capacidad del modelo ”FF-TransUnet”, para lograr una segmentación más precisa y coherente de los órganos abdominales en imágenes de resonancia magnética. Así mismo, resaltan la relevancia del modelo en el ámbito de aplicaciones médicas y su potencial para mejorar los diagnósticos basados en imágenes.

4.3.2. *Resultados de las pruebas estadísticas con el modelo propuesto: FF-TransUnet*

Para evaluar las posibles diferencias significativas en el rendimiento entre los ocho modelos de segmentación, se analizaron los resultados utilizando la prueba estadística de Mann-Whitney U. Estos modelos incluyen tanto el modelo propuesto “FF-TransUnet”, como los mejores siete modelos previamente implementados utilizando imágenes de resonancia magnética con modalidad T1. Para garantizar una comparación justa, todos los modelos fueron sometidos a los mismos tres experimentos.

Debido a que no se asumió una distribución normal en los datos evaluados, la prueba de Mann-Whitney U fue la mejor opción para este análisis. La prueba en cuestión es adecuada para casos donde no se cumplen los supuestos de normalidad porque no depende de la distribución de los datos.

Al realizar el análisis, se obtuvieron los resultados de las comparaciones entre cada par de modelos. Estos resultados fueron sometidos a un análisis cuidadoso para determinar si existían diferencias estadísticamente significativas en el rendimiento entre los modelos evaluados.

Los 8 modelos seleccionados son:

- *MCTrans - ResNet* (Modelo 1)
- *MCTrans - VGG* (Modelo 2)
- *TransUNet - ResNet* (Modelo 3)
- *TransUNet - VGG* (Modelo 4)
- *UNet - ResNet* (Modelo 5)
- *UNet - VGG* (Modelo 6)
- *UNet++ - ResNet* (Modelo 7)
- *FF-TransUnet* (Modelo 8)

Los resultados de la prueba de *Mann-Whitney U* para cada par de modelos se presentan en la Tabla 4.15.

Tabla 4.15: Resultados de la prueba de *Mann-Whitney U* para cada par de modelos. En rojo se resaltan escenarios en donde ocurren diferencias significativas.

	MCT RN	MCT vgg	TUnet RN	TUnet vgg	UNet RN	UNet vgg	UNet++	FF-Trans
MCT RN		0,1	0,2	0,4	0,1	0,1	0,1	0,1
MCT vgg	0,1		0,1	0,121	0,1	0,1	0,268	0,1
TUnet RN	0,2	0,1		0,1	0,1	0,1	0,1	0.506
TUnet vgg	0,4	0,121	0,1		0,1	0,1	0,1	0,1
UNet RN	0,1	0,1	0,1	0,1		1	0,4	0,1
UNet vgg	0,1	0,1	0,1	0,1	1		0,2	0,1
UNet++	0,1	0,268	0,1	0,1	0,4	0,2		0,1
FF-Trans	0,1	0,1	0,506	0,1	0,1	0,1	0,1	

De acuerdo con los resultados de la prueba de *Mann-Whitney U*, se encontraron diferencias significativas en el rendimiento del coeficiente DICE entre los siguientes pares de modelos:

- El modelo *MCTrans - ResNet* con los modelos: *MCTrans - VGG, UNet -ResNet, UNet -VGG, UNet ++* y *FF-TransUnet*.
- El modelo *MCTrans - VGG* con los modelos: *MCTrans - ResNet, TransUNet - ResNet, UNet - ResNet, UNet - VGG* y *FF-TransUnet*.
- El modelo *TransUNet - ResNet* con los modelos: *MCTrans - VGG, TransUNet - VGG, UNet -ResNet, UNet - VGG* y *UNet ++*.
- El modelo *TransUNet - VGG* con los modelos: *TransUNet - ResNet, UNet -ResNet, UNet - VGG, UNet ++* y *FF-TransUnet*.
- El modelo *FF-TransUnet* con los modelos: *MCTrans - ResNet, MCTrans - VGG, TransUNet - VGG, UNet -ResNet, UNet - VGG* y *UNet ++*.

Estos resultados sugieren que los modelos: *TransUNet - ResNet* y *FF-TransUnet* no son significativamente diferentes en términos del coeficiente DICE entre sí. Sin embargo, es importante tener en cuenta que la prueba de *Mann-Whitney U* solo indica si existe o no una diferencia significativa entre los modelos, pero no proporciona información sobre qué modelo es mejor. Además, el coeficiente DICE es solo una medida de rendimiento, y otras métricas de evaluación para la segmentación semántica también deben considerarse al seleccionar el modelo más apropiado para una tarea de segmentación de imágenes en particular.

En particular, el modelo *FF-TransUNet* tiene una diferencia significativa evidente con el resto de los modelos a excepción del modelo *TransUNet - ResNet*, al tener un coeficiente DICE ligeramente más alto que los otros modelos y ser significativamente diferente a todos los demás modelos evaluados en la prueba.

4.3.3. *Prueba de Friedman y comparaciones post hoc con el modelo propuesto: FF-TransUnet*

En este estudio, se compararon ocho modelos de segmentación de imágenes utilizando el rendimiento del coeficiente DICE en tres experimentos diferentes. Con el objetivo de evaluar si existen diferencias significativas en el rendimiento del coeficiente DICE entre los modelos, se aplicó la prueba no paramétrica de *Friedman*, que es apropiada para comparar múltiples modelos en datos de medidas repetidas. La prueba de *Friedman* no asume una distribución normal de los datos y es adecuada para comparar múltiples tratamientos o modelos en el mismo conjunto de sujetos o experimentos.

Los resultados de la prueba de *Friedman* para la comparación de la precisión de los modelos en los tres experimentos son los siguientes:

Estadístico de *Friedman*: 20,333333333, p-valor: 0,004892947

Dado que el p-valor es menor que el nivel de significancia (usualmente 0.05), se puede concluir que hay diferencias significativas en el coeficiente DICE entre los modelos.

Para identificar qué pares de modelos tienen diferencias significativas en su rendimiento, se realizaron comparaciones *post hoc*. Los resultados de las comparaciones *post hoc* se presentan a continuación en la tabla 4.16:

Tabla 4.16: Comparaciones *post hoc* con el modelo propuesto.

	MCT RN	MCT vgg	TUnet RN	TUnet vgg	UNet RN	UNet vgg	UNet++	FF-Trans
MCT RN	1.000000	0.900000	0.900000	0.900000	0.482830	0.373563	0.882297	0.900000
MCT vgg	0.900000	1.000000	0.683012	0.900000	0.900000	0.882297	0.900000	0.583369
TUnet RN	0.900000	0.683012	1.000000	0.900000	0.087199	0.054757	0.373563	0.900000
TUnet vgg	0.900000	0.900000	0.900000	1.000000	0.583369	0.482830	0.900000	0.900000
UNet RN	0.482830	0.900000	0.087199	0.583369	1.000000	0.900000	0.900000	0.054757
UNet vgg	0.373563	0.882297	0.054757	0.482830	0.900000	1.000000	0.900000	0.033212
UNet++	0.882297	0.900000	0.373563	0.900000	0.900000	0.900000	1.000000	0.275374
FF-Trans	0.900000	0.583369	0.900000	0.900000	0.054757	0.033212	0.275374	1.000000

De acuerdo con los resultados de la prueba de *Friedman* y las comparaciones *post hoc*, se encontraron diferencias significativas en el rendimiento del coeficiente DICE entre los siguientes pares de modelos:

- El modelo *TransUNet - ResNet* con los modelos: *UNet -ResNet* y *UNet - VGG*.
- El modelo *FF-TransUNet* con los modelos: *UNet - ResNet* y *UNet - VGG*.

Estos resultados sugieren que los modelos: *TransUNet - ResNet* y *FF-TransUNet* tienen un rendimiento significativamente diferente en términos del coeficiente DICE en comparación con los modelos *UNet - ResNet* y *UNet - VGG*.

4.3.4. *Resultados con el Criterio de Información Akaike (AIC) y Criterio de Información Bayesiano (BIC) con el modelo propuesto: FF-TransUnet*

En la Tabla 4.17, se exhiben los valores de los criterios de información de Akaike (AIC) y Bayesiano (BIC) para cada modelo sometido a evaluación. Mediante un análisis riguroso basado en estos criterios, se procedió a seleccionar el modelo ”MCTrans - VGG”, como el mejor modelo para la tarea de segmentación. Esta elección se fundamenta en que dicho modelo logró alcanzar los valores más reducidos tanto en AIC como en BIC, lo que indica una mayor eficacia y consistencia en la representación de los datos.

Tabla 4.17: Valores de AIC y BIC para cada modelo, incluyendo la propuesta FF-TransUNet

Modelo	AIC	BIC	Mejor según
MCTrans - ResNet	51730968.18197	161558574.78682	
MCTrans - VGG	18387160.1922	57424082.4292	AIC, BIC
TransUNet - ResNet	68491916.17883	213903910.13303	
TransUNet - VGG	35148108.18965	109769417.77599	AIC, BIC
UNet++ - ResNet	52446092.19111	163791945.22325	
FF-TransUNet	67504746.17373	210820925.52008	

Es importante destacar que la selección de modelos se limitó a aquellos que obtuvieron los mejores rendimientos en términos de la métrica de evaluación DICE. Esta medida garantiza

que solo se consideren los modelos más prometedores y con mayor capacidad de adaptación y generalización en la segmentación de los órganos abdominales en imágenes médicas de resonancia magnética. Es relevante enfatizar que en esta evaluación se otorga una significativa importancia a los parámetros del modelo.

La combinación de los resultados obtenidos en las métricas DICE e IoU, el tiempo que tarda en dar el mejor resultado, junto con el análisis estadístico realizado, ratifican la posición destacada del modelo *FF-TransUNet* dentro de la selección de modelos examinados. Este hallazgo respalda la eficiencia y precisión del modelo propuesto en aplicaciones médicas, y proporciona una sólida base para una toma de decisiones fundamentada y confiable por parte de profesionales y especialistas en el área.

Además de los análisis cuantitativos previamente presentados, es esencial comprender cómo se traducen los números en resultados visuales. En este sentido, realizamos un análisis cualitativo de los resultados para proporcionar una comprensión más completa del rendimiento de los modelos. Para ello, presentaremos una selección de imágenes comparativas que muestran tanto las imágenes de entrada como las correspondientes imágenes de ground truth. Además, incluiremos las segmentaciones generadas por los diferentes modelos evaluados. Este enfoque permitirá una evaluación visual rápida y una apreciación de la calidad de las segmentaciones producidas por el modelo propuesto y otros modelos de referencia.

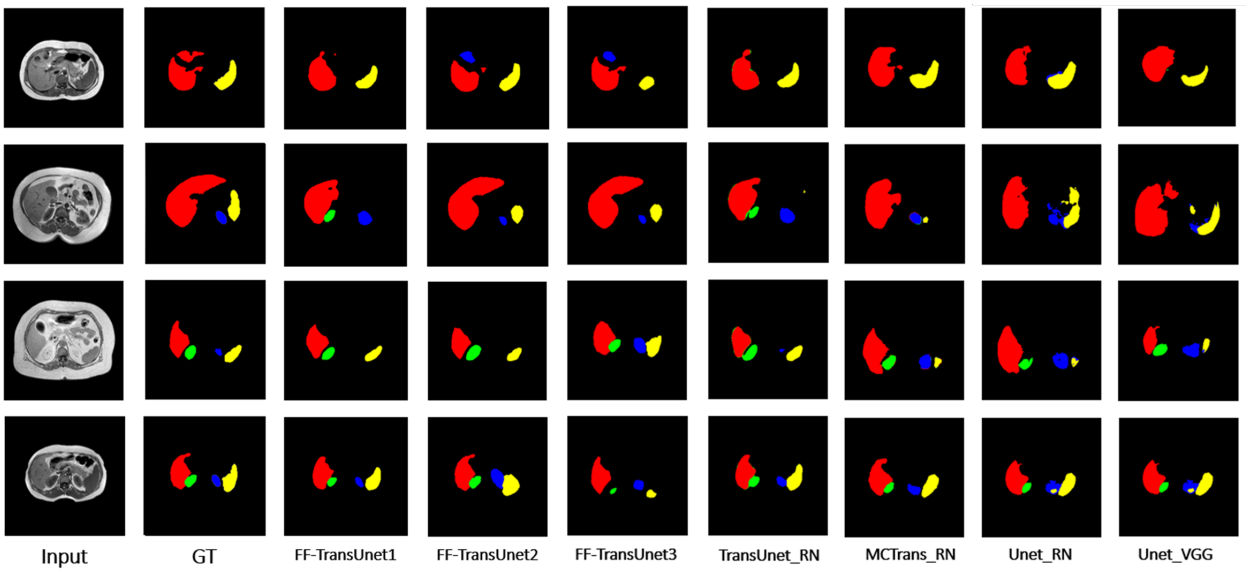


Figura 4.1: Imágenes comparativas de la base de datos CHAOS. Se muestran las imágenes usadas como entrada, su groundtruth y las imágenes generadas por los diferentes modelos de segmentación, incluyendo el modelo propuesto ”FF-TransUnet”.

Con el fin de ofrecer una representación visual exhaustiva del rendimiento de los diferentes modelos de segmentación en nuestro estudio, presentamos en la Figura 4.1 la evaluación de los resultados en imágenes obtenidas de la base de datos CHAOS. Cada fila de la figura corresponde a una imagen de resonancia magnética de un paciente diferente, con identificadores Paciente 31, Paciente 32, Paciente 33 y Paciente 34, respectivamente, y una indicación del corte utilizado (slice 18, slice 12, slice 15 y slice 17, en ese orden).

La primera columna de la figura muestra la imagen de entrada de resonancia magnética proporcionada al modelo de segmentación, mientras que la segunda columna presenta la imagen objetivo (groundtruth) correspondiente, que sirve como referencia. Las columnas de la tercera a la quinta muestran las segmentaciones generadas por nuestro modelo propuesto, ”FF-TransUNet”, organizadas de acuerdo con los tres mejores puntajes obtenidos según la evaluación detallada en la Tabla 4.14.

Las columnas de la sexta a la novena muestran las segmentaciones producidas por los modelos de segmentación MCTrans, TransUNet y Unet. Esta presentación visual permitirá una

evaluación rápida y efectiva de la calidad de las segmentaciones logradas por los modelos analizados.

CAPÍTULO 5. Conclusiones y recomendaciones

Las conclusiones del estudio se basaron en un método que utilizó la base de datos CHAOS, que contiene imágenes de resonancia magnética de pacientes saludables. Durante la investigación se entrenó a la red para identificar características pertinentes de los órganos y realizar segmentaciones precisas.

La validación de los resultados se llevó a cabo comparando las segmentaciones generadas por la red neuronal con las segmentaciones manuales realizadas por expertos médicos. El desempeño de la red fue evaluado a través de métricas como el coeficiente de DICE y la IoU, las cuales demostraron que los modelos basados en *UNet* y *transformers* fueron capaces de realizar segmentaciones precisas y consistentes en múltiples casos clínicos.

Se propuso un modelo que incluye el cambio de la función de pérdida de entropía cruzada por la función de pérdida focal, además el modelo sé basa en la *UNet* y *transformers*. Adicionalmente, durante el entrenamiento se implementó una actualización del parámetro α de la función de pérdida focal, basado en un enfoque difuso.

A partir del análisis de los resultados de este estudio, se pueden extraer las siguientes conclusiones:

1. Como resultado, el modelo propuesto *FF-TransUNet* demostró un rendimiento significativamente mejor en las métricas de evaluación IoU y DICE en comparación con los otros modelos evaluados. Al realizar las pruebas estadísticas, se obtuvieron resultados que respaldan la significativa diferencia de este modelo con respecto a los demás.

2. La prueba de *Mann-Whitney U*, la prueba de *Friedman* y las comparaciones *post hoc* proporcionaron resultados consistentes en términos de diferencias significativas en el rendimiento de los modelos. Estos resultados respaldan la validez de las conclusiones obtenidas a partir de estos análisis.

3. A pesar de que el modelo *TransUnet-VGG* fue seleccionado como el mejor modelo según

los valores de AIC y BIC, en términos de cantidad de parámetros y rendimiento, el modelo propuesto *FF-TransUNet* es competitivo en cuanto al tiempo para obtener el mejor resultado.

A partir de las conclusiones del estudio, se pueden formular las siguientes recomendaciones:

1. Explorar más enfoques de pérdida: Aunque el enfoque de pérdida focal utilizado en este estudio demostró ser efectivo, se sugiere explorar otras funciones de pérdida y evaluar cómo afectan el rendimiento del modelo propuesto. Diferentes funciones de pérdida pueden ofrecer ventajas en términos de convergencia y generalización del modelo.

2. Para tareas de segmentación de imágenes médicas en la modalidad de RM-T1, se recomienda utilizar modelos basados en *MCTrans - ResNet, MCTrans - VGG, TransUNet - ResNet, UNet++ - ResNet* y *FF-TransUNet*, ya que estos modelos mostraron un rendimiento significativamente mejor en términos del coeficiente DICE.

3. Incorporar más datos y modalidades: En este estudio, se utilizó la base de datos CHAOS con imágenes de resonancia magnética en la modalidad T1. Se sugiere incorporar más datos y modalidades, como imágenes en T2, para evaluar cómo afecta la variabilidad de datos en el rendimiento del modelo o incluso, la modalidad de TC, para la segmentación de un órgano.

4. Investigar métodos de aumento de datos: El aumento de datos es una técnica que puede mejorar el rendimiento y la generalización del modelo. Se recomienda investigar y aplicar diferentes técnicas de aumento de datos específicas para imágenes médicas para enriquecer el conjunto de entrenamiento.

5. Para futuras investigaciones, se aconseja explorar la posibilidad de combinar características de teoría difusa, aplicando diferentes funciones de membresía como entradas, para realizar el ajuste del parámetro α o del parámetro γ.

Por lo tanto, la implementación de la arquitectura *UNet* y *transformers* con un enfoque difuso en la práctica clínica puede ser de gran utilidad para los profesionales médicos, ya que facilita el análisis de imágenes de resonancia magnética y mejora la calidad del diagnóstico y la atención al paciente. Además, el uso de este tipo de técnicas de inteligencia artificial puede contribuir a

agilizar y optimizar el proceso de toma de decisiones en el ámbito médico, lo que se traduce en una mejor calidad de vida para los pacientes.

```python
import numpy as np
from scipy.stats import wilcoxon
from scipy.stats import mannwhitneyu
# Datos de DICE de los modelos en cada experimento
exp1 = np.array([91.08, 90.78, 91.46, 90.91, 90.41, 90.43, 90.44])
exp2 = np.array([91.12, 90.82, 91.63, 91.13, 90.24, 90.21, 90.51])
exp3 = np.array([91.47, 90.91, 91.66, 91.01, 90.52, 90.48, 90.82])

#exp1 = np.array([91.16, 90.88, 91.43, 90.93, 90.41, 90.42, 90.56])
#exp2 = np.array([91.16, 90.46, 91.09, 90.78, 89.25, 89.09, 90.1])
#exp3 = np.array([91.59, 91.65, 91.82, 91.15, 89.27, 89.19, 92])

# Funcion para realizar el test de Mann-Whitney U y mostrar los resultados
def compare_models(model1, model2, alpha=0.1):
    model1_accuracies = np.array([exp1[model1 - 1], exp2[model1 - 1], exp3[model1 - 1]])
    model2_accuracies = np.array([exp1[model2 - 1], exp2[model2 - 1], exp3[model2 - 1]])
    u, p = mannwhitneyu(model1_accuracies, model2_accuracies, alternative='two-sided')
    if p <= alpha:
        print(f"El modelo {model1} es significativamente diferente al modelo {model2} (p = {p}).")
    else:
        print(f"El modelo {model1} NO es significativamente diferente al modelo {model2} (p = {p}).")

# Comparar cada par de modelos
for i in range(1, 8):
    for j in range(i + 1, 8):
        compare_models(i, j)
```

Listing 6.1: Código para realizar la prueba de Mann-Whitney U

```python
import numpy as np
```

```python
import pandas as pd
from scipy.stats import friedmanchisquare
import scikit_posthocs as sp

# Datos de DICE de los modelos en cada experimento
accuracies = np.array([
    [91.16, 90.88, 91.43, 90.93, 90.41, 90.42, 90.56],
    [91.16, 90.46, 91.09, 90.78, 89.25, 89.09, 90.1],
    [91.59, 91.65, 91.82, 91.15, 89.27, 89.19, 92]
])

# Realizar el test de Friedman
stat, p = friedmanchisquare(*accuracies.T)
print(f"Estadistico de Friedman: {stat}, p-valor: {p}")

# Crear un DataFrame con los datos de DICE
df = pd.DataFrame(accuracies, columns=[f'Model {i}' for i in range(1, 8)])

# Realizar comparaciones post hoc utilizando la prueba de Nemenyi
post_hoc = sp.posthoc_nemenyi_friedman(df)
print("\nComparaciones post hoc (prueba de Nemenyi):\n")
print(post_hoc)
```

Listing 6.2: Código para realizar la prueba de Friedman

```python
# Datos de accuracy
accuracy_experiments = [
    [91.16, 90.88, 91.43, 90.93, 90.56],
    [91.16, 90.46, 91.09, 90.78, 90.1],
    [91.59, 91.65, 91.82, 91.15, 92]
]

# Numero de parametros para cada modelo
num_parameters = [25865484, 9193580, 34245958, 17574054, 26223046]

# Numero de observaciones
n = 516
```

```
13
14 # Calcula la media de DICE de cada modelo
15 mean_accuracies = np.mean(accuracy_experiments, axis=0)
16
17 # Calcula el AIC y BIC para cada modelo
18 aic_values = []
19 bic_values = []
20
21 for i in range(len(mean_accuracies)):
22     k = num_parameters[i]
23     L = mean_accuracies[i] / 100  # Convertir la precision a una
    probabilidad (0-1)
24     ln_L = np.log(L)
25
26     aic = 2 * k - 2 * ln_L
27     bic = k * np.log(n) - 2 * ln_L
28
29     aic_values.append(aic)
30     bic_values.append(bic)
31
32 # Mostrar los valores de AIC y BIC para cada modelo
33 print("AIC values:", aic_values)
34 print("BIC values:", bic_values)
35
36 # Encuentra el modelo con el AIC y BIC mas bajo
37 best_model_aic = np.argmin(aic_values) + 1
38 best_model_bic = np.argmin(bic_values) + 1
39
40 print("Best model according to AIC:", best_model_aic)
41 print("Best model according to BIC:", best_model_bic)
```

Listing 6.3: Código para realizar los criterios AIC y BIC

Bibliografía

[1] A. Rosebrock, *Deep Learning for Computer Vision with Python: Starte Bundle*. Pyimagesearch, 2017.

[2] S. Ibrahim, R. Djemal, and A. Alsuwailem, "Electroencephalography (eeg) signal processing for epilepsy and autism spectrum disorder diagnosis." *Biocybernetics and Biomedical Engineering*, vol. 1, no. 38, pp. 16–26, 2018.

[3] M. Bowling, J. Fürnkranz, T. Graepel, and R. Musick, "Machine learning and games." *Machine learning*, vol. 3, no. 63, pp. 211–215, 2006.

[4] X. Wu, D. Sahoo, and S. C. Hoi, "Recent advances in deep learning for object detection." *Neurocomputing*, vol. 396, pp. 39–64, 2020.

[5] R. Rummeny, E.and Weissleder, D. D. Stark, S. Saini, C. C. Compton, W. Bennett, and J. T. Ferrucci, "Primary liver tumors: diagnosis by mr imaging." *American Journal of Roentgenology*, vol. 1, no. 152, pp. 63–72, 1989.

[6] M. Gupta, P. S. Choudhury, S. Singh, and D. Hazarika, "Liver functional volumetry by tc-99m mebrofenin hepatobiliary scintigraphy before major liver resection: a game changer." *Indian journal of nuclear medicine: IJNM: the official journal of the Society of Nuclear Medicine, India*, vol. 4, no. 33, p. 277, 2018.

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation." *In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015*, pp. 234–241, 2015.

[8] H. Guo and D. Yang, "Prdnet: medical image segmentation based on parallel residual and dilated network." *Measurement*, p. 173, 2021.

[9] R. A. Khan, Y. Luo, and F. X. Wu, "Rms-unet: Residual multi-scale unet for liver and lesion segmentation." *Artificial Intelligence in Medicine*, vol. 124, p. 50, 2022.

[10] X. Luo, G. Wang, T. Song, J. Zhang, M. Aertsen, J. Deprest, and S. Zhang, "Mideepseg: Minimally interactive segmentation of unseen objects from medical images using deep learning." *Medical Image Analysis*, vol. 72, 2021.

[11] H. Lu, S. Tian, L. Yu, L. Liu, J. Cheng, W. Wu, and D. Zhang, "Dcacnet: Dual context aggregation and attention-guided cross deconvolution network for medical image segmentation." *Computer Methods and Programs in Biomedicine*, vol. 214, 2022.

[12] Y. Lee, J. W. Hwang, S. Lee, Y. Bae, and J. Park, "An energy and gpu-computation efficient backbone network for real-time object detection." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 0–0, 2019.

[13] Y. Guo, Y. Li, Y. Zhang, Y. Li, and Y. He, "Handwritten digit recognition based on spiking-vovnet and extended application." *Journal of Physics: Conference Series*, vol. 1995, no. 1, 2021.

[14] J. Yu, D. Yang, and H. Zhao, "Ffanet: Feature fusion attention network to medical image segmentation." *Biomedical signal processing and control*, vol. 69, 2021.

[15] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning." *In Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018*, pp. 270–279, 2018.

[16] D. Zang, Z. Chai, J. Zhang, D. Zhang, and J. Cheng, "Vehicle license plate recognition using visual attention model and deep learning." *Journal of Electronic Imaging*, vol. 24, no. 3, pp. 033 001–033 001, 2015.

[17] M. Dhyani and R. Kumar, "An intelligent chatbot using deep learning with bidirectional rnn and attention model." *Materials today: proceedings*, vol. 34, pp. 817–824, 2021.

[18] A. de Santana Correia and E. L. Colombini, "Attention, please! a survey of neural attention models in deep learning." *Artificial Intelligence Review*, vol. 55, no. 8, pp. 6037–6124, 2022.

[19] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, and Y. Zhou, "Transunet: Transformers make strong encoders for medical image segmentation." 2021.

[20] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection." *In Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[21] X. Li, H. Zhao, and L. Zhang, "Recurrent retinanet: A video object detection model based on focal loss." *In Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part IV 25*, pp. 499–508, 2018.

[22] A. E. Kavur, *CHAOS - Combined (CT-MR) Healthy Abdominal Organ Segmentation*. Elsevier BV, Apr. 2021, vol. 69. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361841520303145

[23] B. Mahesh, "Machine learning algorithms-a review." *International Journal of Science and Research (IJSR).*, vol. 9, no. 1, pp. 381–386, 2020.

[24] J. C. Ang, A. Mirzal, H. Haron, and H. N. A. Hamed, "Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection." *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 13, no. 5, pp. 971–989, 2015.

[25] C. Donalek, "Supervised and unsupervised learning." *In Astronomy Colloquia*, vol. 27, 2011.

[26] R. Vargas, A. Mosavi, and R. Ruiz, "Deep learning: a review." 2017.

[27] Y. LeCun, "Deep learning." *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[28] R. Szeliski, *Computer vision: algorithms and applications.* Springer Nature, 2022.

[29] A. Peña-Peñate, L. G. Silva Rojas, and R. Alcolea Núñez, "Módulo de filtrado y segmentación de imágenes médicas digitales para el proyecto vismedic." *Revista Cubana de Ciencias Informáticas*, vol. 10, no. 1, pp. 13–27, 2016.

[30] Z. Huang, Z. Pan, and B. Lei, "Transfer learning with deep convolutional neural network for sar target classification with limited labeled data." *Remote Sensing*, vol. 9, no. 9, p. 907, 2017.

[31] S. Albawi, O. Bayat, S. Al-Azawi, and O. N. Ucan, "Social touch gesture recognition using convolutional neural network." *Computational Intelligence and Neuroscience*, vol. 2018, 2017.

[32] J. Ludwig, "Image convolution." *Portland State University.*, 2013.

[33] J. Nagi, F. Ducatelle, G. A. Di Caro, D. Cireşan, U. Meier, A. Giusti, and L. M. Gambardella, "Max-pooling convolutional neural networks for vision-based hand gesture recognition." *2011 IEEE international conference on signal and image processing applications (ICSIPA)*, pp. 342–347, 2011.

[34] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, and G. Fricout, "Steel defect classification with max-pooling convolutional neural networks." *The 2012 international joint conference on neural networks (IJCNN)*, pp. 1–6, 2012.

[35] B. Pourbabaee, M. J. Roshtkhari, and K. Khorasani, "Deep convolutional neural networks and learning ecg features for screening paroxysmal atrial fibrillation patients." *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 12, pp. 2095–2104, 2018.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556.*, 2014.

[37] J. Dai, K. He, Y. Li, S. Ren, and J. Sun, "Instance-sensitive fully convolutional networks." *In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016*, vol. 24, pp. 27–33, 2016.

[38] S. Tammina, "Transfer learning using vgg-16 with deep convolutional neural network for classifying images." *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 10, pp. 143–150, 2019.

[39] C. Sitaula and M. B. Hossain, "Attention-based vgg-16 model for covid-19 chest x-ray image classification." *Applied Intelligence*, vol. 51, pp. 2850–2863, 2021.

[40] M. Mateen, J. Wen, S. Nasrullah, Song, and Z. Huang, "Fundus image classification using vgg-19 architecture with pca and svd." *Symmetry*, vol. 11, no. 1, p. 1, 2018.

[41] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: Vgg and residual architectures." *Frontiers in neuroscience*, vol. 13, p. 95, 2019.

[42] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks." *Advances in neural information processing systems*, vol. 29, pp. 379–387, 2016.

[43] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks." *En Proceedings of the IEEE international conference on computer vision.*, pp. 3119–3127, 2015.

[44] Q. Chen, J. Xu, and V. Koltun, "Fast image processing with fully-convolutional networks." *En Proceedings of the IEEE International Conference on Computer Vision.*, pp. 2497–2506, 2017.

[45] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation." *In Proceedings of the IEEE conference on computer vision and pattern recognition.*, pp. 3431–3440, 2015.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[47] Z. Wu, C. Shen, and A. Van Den Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition." *Pattern Recognition*, vol. 90, pp. 119–133, 2019.

[48] S. Targ, D. Almeida, and K. Lyman, "Resnet in resnet: Generalizing residual architectures." *arXiv preprint arXiv:1603.08029*, 2016.

[49] X. Xiao, S. Lian, Z. Luo, and S. Li, "Weighted res-unet for high-quality retina vessel segmentation." *2018 9th international conference on information technology in medicine and education (ITME)*, pp. 327–331, 2018.

[50] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation." *In International Workshop on Deep Learning in Medical Image Analysis, International Workshop on Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*, pp. 179–187, 2016.

[51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, and I. Polosukhin, "Attention is all you need." *Advances in neural information processing systems*, vol. 30, 2017.

[52] Z. Wang, Y. Ma, Z. Liu, and J. Tang, "R-transformer: Recurrent neural network enhanced transformer." *arXiv preprint arXiv:1907.05572.*, 2019.

[53] Y. Liu, G. Sun, Y. Qiu, L. Zhang, A. Chhatkuli, and L. Van Gool, "Transformer in convolutional neural networks." *arXiv preprint arXiv:2106.03180.*, vol. 3, 2021.

[54] C. F. R. Chen, Q. Fan, and R. Panda, "Crossvit: Cross-attention multi-scale vision transformer for image classification." *In Proceedings of the IEEE/CVF international conference on computer vision.*, pp. 357–366, 2021.

[55] O. Petit, N. Thome, C. Rambour, L. Themyr, T. Collins, and L. Soler, "U-net transformer: Self and cross attention for medical image segmentation." *In Machine Learning in Medical Imaging: 12th International Workshop, MLMI 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, September 27, 2021, Proceedings 12.*, pp. 267–276, 2021.

[56] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection." *arXiv preprint arXiv:2010.04159.*, 2020.

[57] S. Pan, X. Liu, N. Xie, and Y. Chong, "Eg-transunet: Enhanced and guided u-net with transformer for biomedical image segmentation." *Research Square*, 2022.

[58] Y. Ji, R. Zhang, H. Wang, Z. Li, L. Wu, S. Zhang, and P. Luo, "Multi-compound transformer for accurate biomedical image segmentation." *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021*, pp. 326–336, 2021.

[59] K. H. Lee, *First course on fuzzy theory and applications.*, 2004, vol. 27.

[60] J. S. R. Jang, C. T. Sun, and E. Mizutani, *Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence.* IEEE Transactions on automatic control, 1997.

[61] M. F. Hamza, H. J. Yap, and I. A. Choudhury, "Genetic algorithm and particle swarm optimization based cascade interval type 2 fuzzy pd controller for rotary inverted pendulum system." *Mathematical Problems in Engineering*, 2015.

[62] A. V. Duka and S. E. Oltean, "Fuzzy control of a heat exchanger." *In Proceedings of 2012 IEEE International Conference on Automation, Quality and Testing, Robotics*, pp. 135–139, 2012.

[63] J. E. M. Reyes, J. J. C. Pérez, and A. I. C. Llanos, "Red neuronal autorregresiva difusa tipo sugeno con funciones de membresía triangular y trapezoidal: una aplicación al pronóstico de índices del mercado bursátil." *Estocástica: finanzas y riesgo.*, vol. 10, no. 1, pp. 77–101, 2020.

[64] T. Tapia Molina and J. L. Merma Aroni, "Interpolación chebyshev como función membresía en lógica difusa abancay." 2018.

[65] P. C. Shill, M. A. H. Akhand, M. D. Asaduzzaman, and K. Murase, "Optimization of fuzzy logic controllers with rule base size reduction using genetic algorithms." *International Journal of Information Technology Decision Making*, vol. 14, no. 5, pp. 1063–1092, 2015.

[66] I. Soesanti and R. Syahputra, "A fuzzy logic controller approach for controlling heat exchanger temperature." *Journal of Electrical Technology UMY*, vol. 3, no. 4, pp. 117–124, 2019.

[67] M. Prasad, Y. T. Liu, D. L. Li, C. T. Lin, R. R. Shah, and O. P. Kaiwartya, "A new mechanism for data visualization with tsk-type preprocessed collaborative fuzzy rule based system." *Journal of Artificial Intelligence and Soft Computing Research*, vol. 7, no. 1, pp. 33–46, 2017.

[68] L. Suganthi, S. Iniyan, and A. A. Samuel, "Applications of fuzzy logic in renewable energy systems–a review." *Renewable and sustainable energy reviews*, vol. 48, pp. 585–607, 2015.

[69] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling." *IEEE Transactions on fuzzy systems*, vol. 1, no. 1, p. 7, 1993.

[70] S. Dettori, V. Iannino, V. Colla, and A. Signorini, "An adaptive fuzzy logic-based approach to pid control of steam turbines in solar applications." *Applied Energy*, no. 227, pp. 655–664, 2018.

[71] B. Chen, X. P. Liu, S. S. Ge, and C. Lin, "Adaptive fuzzy control of a class of nonlinear systems by fuzzy approximation approach." *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 6, pp. 1012–1021, 2012.

[72] S. Jha, R. Kumar, I. Priyadarshini, F. Smarandache, and H. V. Long, "Neutrosophic image segmentation with dice coefficients." *Measurement*, vol. 134, pp. 762–772, 2019.

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

[74] A. Rusiecki, "Trimmed categorical cross-entropy for deep learning with label noise." *Electronics Letters.*, vol. 55, no. 6, pp. 319–320, 2019.

[75] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels." *Advances in neural information processing systems.*, vol. 31, 2018.

[76] G. Divine, H. J. Norton, R. Hunt, and J. Dienemann, "A review of analysis and sample size calculation considerations for wilcoxon tests." *Anesthesia Analgesia*, vol. 117, no. 3, pp. 699–710, 2013.

[77] R. S. Turcios, "Prueba de wilcoxon-mann-whitney: mitos y realidades." *Rev Mex Endocrinol Metab Nutr*, vol. 2, pp. 18–21, 2015.

[78] F. McElduff, M. Cortina-Borja, S. K. Chan, and A. Wade, "When t-tests or wilcoxon-mann-whitney tests won't do." *Advances in physiology education*, vol. 34, no. 3, pp. 128–133, 2010.

[79] D. W. Zimmerman and B. D. Zumbo, "Relative power of the wilcoxon test, the friedman test, and repeated-measures anova on ranks." *The Journal of Experimental Education*, vol. 62, no. 1, pp. 75–86, 1993.

[80] M. R. Sheldon, M. J. Fillyaw, and W. D. Thompson, "The use and interpretation of the friedman test in the analysis of ordinal-scale data in repeated measures designs." *The Journal of Experimental Education*, vol. 1, no. 4, pp. 221–228, 1996.

[81] M. Chatfield and A. Mander, “The skillings–mack test (friedman test when there are missing data).” *The Stata Journal*, vol. 9, no. 2, pp. 299–305, 2009.

[82] L. J. Williams and H. Abdi, “Post-hoc comparisons.” *Encyclopedia of research design*, pp. 1060–1067, 2010.

[83] S. I. Vrieze, “Model selection and psychological theory: a discussion of the differences between the akaike information criterion (aic) and the bayesian information criterion (bic).” *Psychological methods*, vol. 17, no. 2, p. 228, 2012.

[84] H. Bozdogan, “Model selection and akaike’s information criterion (aic): The general theory and its analytical extensions.” *Psychometrika*, vol. 52, no. 3, pp. 345–370, 1987.

[85] A. A. Neath and J. E. Cavanaugh, “The bayesian information criterion: background, derivation, and applications.” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 4, no. 2, pp. 199–203, 2012.

[86] MITA, *Digital Imaging and Communications in Medicine*. https://www.dicomstandard.org/.

[87] S. A. Stoykov., *Micro Dicom*. https://www.microdicom.com/.

[88] A. Sinha and J. Dolz, “Multi-scale self-guided attention for medical image segmentation.” *IEEE journal of biomedical and health informatics.*, vol. 25, no. 1, pp. 121–130, 2020.

[89] P. H. Conze, A. E. Kavur, E. Cornec-Le Gall, N. S. Gezer, Y. Le Meur, M. A. Selver, and F. Rousseau, “Abdominal multi-organ segmentation with cascaded convolutional and adversarial deep networks.” *Artificial Intelligence in Medicine.*, vol. 117, pp. 102–109, 2021.

[90] W. Yuan, J. Wei, J. Wang, Q. Ma, and T. Tasdizen, “Unified generative adversarial networks for multimodal segmentation from unpaired 3d medical images.” *Medical Image Analysis.*, vol. 67, p. 101731, 2020.

[91] L. Wang, D. Guo, G. Wang, and S. Zhang, “Annotation-efficient learning for medical image segmentation based on noisy pseudo labels and adversarial learning.” *IEEE Transactions on Medical Imaging.*, vol. 40, no. 10, pp. 2795–2807, 2020.

[92] Q. Wang, Y. Du, H. Fan, and C. Ma, “Towards collaborative appearance and semantic adaptation for medical image segmentation.” *Neurocomputing.*, vol. 491, pp. 633–643, 2022.

[93] R. Li and X. Chen, “An efficient interactive multi-label segmentation tool for 2d and 3d medical images using fully connected conditional random field.” *Computer Methods and Programs in Biomedicine.*, vol. 213, p. 106534, 2022.

[94] D. R. Martınez, J. Albın, J. Cabaleiro, T. Pena, F. Rivera, and V. Blanco, “El criterio de información de akaike en la obtención de modelos estadísticos de rendimiento.” *In Conference: XX Jornadas de Paralelismo.*, 2009.

[95] A. Montesinos-López, “Estudio del aic y bic en la selección de modelos de vida con datos censurados.” *Centro de investigación en matemáticas.*, 2011.