



UNIVERSIDAD NACIONAL
DE EDUCACIÓN A DISTANCIA

Escuela Técnica Superior de Ingeniería Informática

SEGMENTACIÓN SEMÁNTICA EFICIENTE PARA DETECCIÓN DE
TUMORES EN ECOGRAFIAS MAMARIAS EN TIEMPO REAL

González-Novo Hueso, Jaime

Director: Cuadra Troncoso, José Manuel

Trabajo de Fin de Máster

Máster Universitario
en Investigación en Inteligencia Artificial

Septiembre 2024

Agradecimientos

Este trabajo presenta un hito en mi carrera académica que no hubiera sido posible sin una red de apoyo incondicional con la que he tenido la suerte de contar. Quisiera agradecer a mi familia por aportar los recursos, el tiempo y la motivación necesaria para obtener este logro, por celebrar los avances, entender las frustraciones y apoyar mis decisiones. En definitiva, por la inagotable confianza que demostráis tener en mí cada día. Agradezco enormemente a mi pareja por haberme acompañado durante mi paso por la Universidad dotándome de las herramientas y el tesón necesario para construir un perfil académico sólido. Al mismo tiempo has sido capaz de darme momentos irrepetibles y espectaculares que atesoro en la memoria. Gracias a ti, a tu indomable espíritu de auto-superación y a tu altruismo radical soy más y soy mejor en todos los ámbitos ya que siempre llevo parte de ti conmigo. Agradezco a todos aquellos amigos y personas cercanas que han sido capaces de desviar mi atención hacia otras áreas en las que he podido nutrirme de forma personal. Gracias a vosotros puedo preservar la creatividad y aumentar el pensamiento lateral por medio de interacciones entre distintos campos de conocimiento. También agradecer a mis compañeros en la técnica. Con vosotros he crecido y mejorado en cada proyecto compartido por medio de la valoración y admiración mutua en la multitud de problemas enfrentados y soluciones encontradas. Por último, pero no por ello menos relevante, agradecer al director de este trabajo, José Manuel, que me introdujo en el campo de la Ciencia de Datos, me ayudó de forma desinteresada a conseguir las herramientas necesarias para iniciar mi aprendizaje y no dudó de mis capacidades al proponerse como director de este trabajo.

A todos vosotros, mi más sincero agradecimiento.

Resumen

La segmentación semántica para detección de tumores mamarios en imágenes de ultrasonido utilizando aprendizaje profundo es un problema de alta complejidad técnica. La deriva actual de la investigación sobre este caso de uso consiste en aumentar la potencia predictiva de los modelos por medio del incremento del tamaño de los modelos en sí. En este trabajo, por el contrario, se apuesta por la eficiencia para disminuir la latencia en la respuesta, aumentar la accesibilidad y disminuir el coste computacional y el consecuente impacto medioambiental. Con este objetivo en mente, se aplica la arquitectura EFSNet a los datos abiertos BUSI. EFSNet es de la familia de las U-Net y cuenta con 179k parámetros (143 *Mb en inferencia*) posicionándose como una de las arquitecturas más livianas del estado del arte. En los experimentos realizados sobre el conjunto de datos BUSI se obtiene un 94,06 % de *accuracy*, un 70,44 % de *dsc* y un 58,14 % de *mIoU*, y se comprueba una efectividad similar a la obtenida con modelos de varios órdenes de magnitud más de parámetros. Además, EFSNet muestra un tiempo de inferencia de 82 imágenes por segundo (FPS) en una única tarjeta gráfica (GPU) *NVidia GeForce 4060 RTX* que permite la inferencia en tiempo real. Comparado con el estado del arte la solución presentada obtiene predicciones precisas con un consumo de recursos extremadamente bajo.

Palabras clave: EFSNet, Segmentación Semántica, Recursos Limitados, Tiempo Real, Tumores Mamarios, Imágenes de Ultrasonido.

Abstract

Semantic segmentation for breast tumor detection on ultrasound images using deep learning is a problem of high technical complexity. The current drift of research on this use case is to increase the predictive power of the models by increasing the size of the models themselves. In this work, on the contrary, we focus on efficiency to reduce response latency, increase accessibility and reduce computational cost and the consequent environmental impact. To achieve this goal the EFSNet architecture is applied to BUSI open dataset. EFSNet is from the U-Net family and has $179k$ parameters (143 Mb *in inference*), positioning itself as one of the lightest architectures in the state of the art. In the experiments carried out on the BUSI Dataset, 94,06 % accuracy, 70,44 % dsc and 58,14 % mIoU are obtained, and an effectiveness similar to that obtained with models with several orders of magnitude more parameters is verified. Additionally, EFSNet displays an inference time of 82 frames per second (FPS) on a single *NVidia GeForce 4060 RTX* graphics card (GPU) enabling real-time inference. Compared to the state of the art, the presented solution obtains accurate predictions with extremely low resource consumption.

Keywords: EFSNet, Semantic Segmentation, Resource Constrained, Real Time, Breast Tumor, Ultrasound Images.

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Propuesta y objetivos	3
1.3. Estructura del documento	3
2. Estado del Arte	5
2.1. Visión Artificial	5
2.1.1. Arquitecturas	6
2.1.1.1. Bloques de Atención	6
2.1.1.2. Modelos fundacionales y multimodalidad	7
2.1.1.3. Eficiencia computacional	9
2.1.1.4. Otros avances	12
2.1.2. Algoritmos de optimización	13
2.1.2.1. Difusión guiada	13
2.1.2.2. Redes Generativas Adversarias	14
2.1.2.3. Pasos de aprendizaje variables y decaimiento de pesos	15
2.1.2.4. Aprendizaje Federado	16
2.1.2.5. Afinamiento y Transferencia de Aprendizaje	18
2.2. Segmentación semántica en aplicaciones biomédicas	19
2.2.1. Principales fuentes de datos y tareas	19
2.2.2. Principales arquitecturas y soluciones	21
2.2.3. Detección de tumores en ecografías mamarias	24
3. Materiales y métodos	29
3.1. Materiales	29
3.1.1. Software	29
3.1.2. Hardware	30
3.1.3. Datos	30
3.1.3.1. Origen	30
3.1.3.2. Cualidades	32
3.1.3.3. Transformaciones iniciales	32

3.2. Métodos	34
3.2.1. EFSNet	34
3.2.1.1. Bloque Convolucional	37
3.2.1.2. Bloque Inicial	37
3.2.1.3. Bloque <i>Downsampling</i>	38
3.2.1.4. Bloque Factorizado	40
3.2.1.5. Bloque CSDC	41
3.2.1.6. Bloque SDC	42
3.2.1.7. Módulo de <i>Upsampling</i>	43
3.2.1.8. <i>ShuffleNet Unit</i>	44
3.2.1.9. Predicción	45
3.2.2. Entrenamiento y validación	45
3.2.2.1. Aumento de imágenes	46
3.2.2.2. Función de pérdida	47
3.2.2.3. Optimización	48
4. Resultados	49
4.1. Métricas	49
4.2. Aciertos	53
4.3. Errores	54
4.4. Tiempo Real	56
5. Conclusiones y trabajos futuros	59
5.1. Trabajo futuro	60
Bibliografía y referencias	62
A. Aprendizaje Adversario con Segmentación Semántica Semi-Supervisada	83
A.1. Algoritmo AL-S4	83
A.2. Optimización	84
A.3. Resultados	87
A.4. Conclusiones	89
B. Formulación matemática	91
B.1. BatchNorm y sesgo	91
B.2. Número de parámetros en convoluciones desagregadas	91
B.3. Entropía cruzada	92
B.4. Accuracy	93
B.5. Coeficiente de similaridad de Dice	93
B.6. Índice de Jaccard	93

C. Imágenes adicionales	95
C.1. Principales metodologías para convolución eficiente	96
C.2. Transformaciones de aumento de datos	98
C.3. Matrices de confusión	99

Índice de figuras

2.1.	Mecanismo de atención.	6
2.2.	Mapa de Auto-Atención.	7
2.3.	Esquema de un modelo fundacional.	8
2.4.	Estrategias para la multimodalidad.	9
2.5.	Difusión Guiada.	14
2.6.	Metodología <i>GAN</i>	15
2.7.	Aprendizaje Federado.	17
2.8.	Arquitectura U-Net.	22
3.1.	Imagen con aberraciones.	32
3.2.	Redimensionado de imagen.	33
3.3.	Procesamiento de máscaras.	33
3.4.	EFSNet con parámetros por capa en de fondo en gris.	35
3.5.	Bloque Convolucional.	37
3.6.	Bloque inicial.	38
3.7.	Bloque de reducción de dimensionalidad.	39
3.8.	Bloque Factorizado.	40
3.9.	Bloque CSDC.	41
3.10.	Bloque SDC.	42
3.11.	Módulo de <i>Upsampling</i>	43
3.12.	<i>ShuffleNet Unit</i>	44
3.13.	Entrenamiento de EFSNet.	48
4.1.	Aciertos en tumores benignos pequeños y/o compactos.	53
4.2.	Aciertos en imágenes sin presencia de tumores.	53
4.3.	Errores por agujeros en las predicciones.	55
4.4.	Errores por derrame en las predicciones.	55
4.5.	Errores por mezcla e intercambio de clases.	56
4.6.	Errores de dudosa veracidad.	56
A.1.	Esquema de implementación de AL-S4.	84
A.2.	Aciertos AL-S4.	88

A.3. Errores AL-S4.	88
C.1. Reducción de parámetros en operaciones de convolución.	96
C.2. Bloque residual y <i>skip-connections</i>	97
C.3. Transformaciones de aumento de datos.	98
C.4. Matrices de confusión.	99

Índice de tablas

3.1.	Número de imágenes por categoría.	30
3.2.	Dimensiones de las imágenes originales.	32
3.3.	Parámetros de EFSNet por bloque.	36
3.4.	Distribución de imágenes por categoría y conjunto de datos.	45
3.5.	Pesos CCE.	47
4.1.	Tablas de métricas.	50
4.2.	Comparación de modelos binaria.	52
4.3.	Tiempo y frecuencia de inferencia.	56
A.1.	Comparativa de métricas clásico y AL-S4.	88

Nomenclatura

AL-S4 Adversarial Learning with Semi-Supervised Semantic Segmentation,
página 81

BatchNorm Batch Normalization, página 10

BCE Binary Cross-Entropy, página 45

BUSI Breast UltraSound Images, página 28

CCE Categorical Cross-Entropy, página 45

CNN Convolutional Neural Network, página 21

CSDC Continuous Shuffled Dilated Convolutions, página 32

CT Computerized Tomography, página 19

DSC Dice Score Coefficient, página 47

EFSNet Efficient Fast Semantic Segmentation Network, página 2

FCN Fully Convolutional Network, página 21

FPS Frames Per Second, página 54

GAN Generative Adversarial Networks, página 6

GNN Graph Neural Network, página 12

MIOU Mean Intersection over Union, página 47

MoE Mixture of Experts, página 1

MRI Magnetic Resonance Imaging, página 19

OASBUD Open Access Series of Breast Ultrasonic Data, página 29

OCT Optical Coherence Tomography, página 19

PEFT Parameter Efficient Fine Tuning, página 1

PET Positron Emission Tomography, página 19

PFDD Progressive Feature Distribution Distillation, página 2

PReLU Parametrized Rectified Linear Unit, página 10

QLoRA Quantized Low Rank Adaptation, página 1

ReLU Rectified Linear Unit, página 42

RFLPA Robust Federated Learning Framework against Poisoning Attacks, página 17

SDC Shuffled Dilated Convolution, página 35

SGD Stochastic Gradient Descent, página 15

SGD Stochastic Gradient Descent, página 83

SSSM Selective State Space Model, página 12

USI UltraSound Imaging, página 19

WSI Whole Slide Imaging, página 19

Capítulo 1

Introducción

1.1. Motivación

El trabajo que a continuación se presenta es una clara apuesta por la eficiencia y optimización de las técnicas comúnmente utilizadas en Inteligencia Artificial en el campo de la biomedicina.

Esta apuesta se fundamenta en la accesibilidad por parte de entidades médicas y grupos sociales con bajo nivel de recursos a técnicas avanzadas de diagnóstico de enfermedades, particularmente, detección de tumores en ecografías mamarias.

En la actualidad, el cáncer es una de las principales causas de muerte a nivel global (Siegel et al. (2019)) y, mientras que la causa no depende exclusivamente de la situación socio-económica del individuo, el tratamiento sí. Entre las mujeres, el cáncer de mama es el tipo de cáncer más común (de Registros del Cáncer (2019)) por lo que la detección temprana es determinante para la reducción de casos fatales.

Este trabajo escoge una metodología basada en la reducción de parámetros de las redes neuronales, ya que fomenta mejores prácticas en términos éticos y morales que las que devienen del grueso de modelos actuales, fundamentadas en la idea de “a mayor número de parámetros mejores predicciones”.

Comenzando por la eficiencia energética, la reducción del coste computacional del algoritmo requiere menos energía para ser ejecutada reduciendo: las emisiones derivadas de la obtención de dicha energía, el coste monetario y los requisitos de hardware.

Esta vía de desarrollo ha sido tomada en los últimos años por las grandes empresas del sector como *OpenAI*, *Meta* o *Google* que han logrado una Inteligencia Artificial Generativa con mayor potencia predictiva y menor número de parámetros, haciendo uso de técnicas como el Modelos de Mezcla de Expertos (Eigen et al. (2013)), *MoE*, estrategias de ajuste fino con eficiencia de parámetros (Liao et al. (2023); Hu et al. (2021)), *PEFT*, cuantización de

parámetros (BitNet (Wang et al. (2023)) y QLoRA (Dettmers et al. (2024))) y destilación de conocimiento y podado de modelo (EPSD (Chen et al. (2024)), PFDD (Zhou et al. (2022a)), Prune Before Distill (Park and No (2022))).

Estas empresas han optado por la optimización únicamente con la finalidad de reducir costes y aumentar beneficios dado que sus avances, generalmente, no son compartidos con la comunidad. Además, la optimización se utiliza como propaganda pro-ecologismo para dar una imagen de concienciación alejada de la realidad (*green-washing*) (Seele and Schultz (2022)). En consecuencia, la eficiencia energética es necesaria pero no suficiente y, por lo tanto, exige un requisito adicional: ser fiel a la filosofía *open-source* del lenguaje de programación sobre el que se sustentan estas tecnologías (*Python*).

Las entidades mencionadas evitan esta filosofía por temor a la competencia, ora ofuscando sus avances, ora evitando la replicación y utilización de sus técnicas por terceras partes y, por tanto, lastrando la ciencia que les ha permitido implementar y desplegar dichas tecnologías.

Este trabajo servirá de guía para la replicación de la red neuronal *EFSNet* (Hu and Wang (2020)) de forma que cualquier entidad o individuo sea capaz de implementarla y utilizarla.

Finalmente, la mejora de eficiencia en términos de número de parámetros permite una disminución en la latencia de la respuesta, tanto es así que es posible realizar predicciones en tiempo real. La inferencia en tiempo real facilita el uso, aliviando el trabajo de obtener capturas de los momentos que el personal médico experto considere más relevantes.

Adicionalmente, la posibilidad de realizar predicciones consecutivas suaviza errores debido a que las posibles aberraciones de cada resultado variarán más que las regiones de interés (*tumor*). Además, dado sus bajos requerimientos de memoria y computación, permite una integración sencilla en el *streaming* del ecógrafo.

La posesión de esta tecnología por un centro médico facilita también el uso de datos de carácter privado como son los datos médicos. Debido a los requisitos de computación, cualquier centro médico puede entrenar esta red *on-premise* para diversos casos de uso sin preocuparse de filtraciones o robo de información por terceras partes. La posibilidad de entrenar y realizar inferencia en privado, sin siquiera necesidad de acceso a internet, añade una capa de protección suficiente que evita procesos burocráticos y reduce el riesgo de incursión en delitos por mala praxis de uno o de terceros.

1.2. Propuesta y objetivos

Se propone una red neuronal tipo EFSNet (Hu and Wang (2020)) con $\simeq 179k$ parámetros (suma de pesos y sesgo), entrenada con los datos abiertos de BUSI (Al-Dhabayni et al. (2020)) para obtener un modelo eficiente y en tiempo real ajustado a la tarea de segmentación semántica de tumores en ecografías mamarias. El objetivo es que el modelo presentado pueda ser implementado, ejecutado y entrenado de forma sencilla (previo conocimiento técnico) en cualquier centro médico que disponga de un aparato de ultrasonidos.

La arquitectura EFSNet se fundamenta en la eficiencia vía optimización de recursos sacrificando el mínimo desempeño posible en tareas de Visión Artificial. Con este fin, explota las virtudes de técnicas conocidas como los bloques residuales (Szegedy et al. (2015, 2017)), la regularización *dropout* (Srivastava et al. (2014)), la dilatación en convoluciones (Yu and Koltun (2015)) o la normalización por lotes, *BatchNorm*, (Ioffe and Szegedy (2015)) de una forma inteligente y efectiva. La reciente presentación de esta arquitectura (2020) junto con sus resultados la sitúa por encima de otras arquitecturas que han surgido en esta línea como ESPNet (Mehta et al. (2018)), FSSNet (Zhang et al. (2019)), ENet (Paszke et al. (2016a)) o VGGNet (Simonyan and Zisserman (2014)).

La mejora en la precisión respecto de otros modelos no ha sido el punto principal de la investigación. Es evidente que otras redes con mayor capacidad y mayor cantidad de datos pueden obtener mejores métricas de evaluación. En contraparte, la inferencia en tiempo real permite al experto al cargo discriminar con mayor certeza los posibles errores del modelo, mientras que si se hace uso de un modelo *pesado* habrá que elegir previamente qué imagen o conjunto de imágenes son de interés. Por lo tanto, la posible comparación de métricas no es honesta. Conociendo este hecho, se ha apostado por obtener la mejor precisión (*accuracy*) posible para la segmentación semántica de las imágenes en las categorías: *normal*, *benigno*, *maligno*, así como en las categorías: *normal*, *tumor*; para lo que se han tomado en consideración varias vías de procesamiento y aumento de la cantidad de datos.

Los principales retos que se han debido afrontar han sido: la baja cantidad de muestras disponibles, el desbalance de clases, la calidad del dato y la implementación de la red neuronal desde el esquema teórico propuesto por los autores. A lo largo del trabajo se presentarán soluciones y decisiones tomadas al respecto de los retos que han permitido la obtención de una solución aceptable.

1.3. Estructura del documento

La memoria presentada se divide en cinco capítulos.

Una vez vista la “*Introducción*” (capítulo 1), el capítulo actual, se continúa con el “*Estado del Arte*” (capítulo 2) donde se exponen los avances que han dado lugar a la obtención del modelo de Inteligencia Artificial presentado. Se detallarán diferentes líneas de investigación con sus ventajas y desventajas y se justificará cuáles son las razones que han llevado a implementar la solución presentada. Se describen las posibles opciones a tener en cuenta especificando cuáles se alinean más con el objetivo, siguiendo un enfoque desde una perspectiva general a una específica.

En tercer lugar, se realiza una descripción detallada de la solución implementada en el capítulo de “*Materiales y Métodos*” (capítulo 3) en la que se detallan los datos utilizados para el ajuste del modelo, sus características y preprocesamiento, y la arquitectura de la red neuronal EFSNet junto con metodología de implementación y uso en términos algorítmicos, de hardware y de software. En este capítulo se especifican los parámetros e hiperparámetros utilizados en cada punto del estudio de manera que se pueda replicar con exactitud el experimento realizado.

En el cuarto capítulo, “*Resultados*” (capítulo 4), se expondrán los resultados numéricos obtenidos, comentando fortalezas y debilidades de la aplicación de la EFSNet al conjunto de datos mencionado. Por otro lado, en esta misma sección, se realiza un análisis visual de los resultados a través de los cuales se formulan hipótesis que pueden ser de ayuda en la interpretación de la red neuronal y su convergencia.

Finalmente, en la sección de “*Conclusiones*” (capítulo 5), se evalúa el cumplimiento de las propuestas y objetivos, y se establecen vías de desarrollo para futuros trabajos.

Las secciones principales mencionadas están acompañadas de las secciones de: “*Referencias*”, “*Apéndice*” y “*Agradecimientos*”; en la que se puede encontrar información adicional sobre otros trabajos, procedimientos específicos o personas relevantes para la consecución de esta memoria.

Capítulo 2

Estado del Arte

En este capítulo se expondrá el estado actual de la detección de tumores en ecografías mamarias desde el prisma de las arquitecturas de las redes neuronales artificiales y su optimización. Para obtener una visión de conjunto holística que permita el entendimiento de lo general y lo particular se realiza un estudio sobre los avances en el campo más global dentro del campo de la Inteligencia Artificial, el denominado como Visión Artificial, hasta el correspondiente al trabajo, detección de tumores en ecografías mamarias por segmentación semántica.

2.1. Visión Artificial

En el campo de la Visión Artificial se pueden observar varias vías de desarrollo en cuestión de arquitecturas y algoritmos de optimización aplicables a multitud de casos de uso.

Desde los primeros hitos de las arquitecturas de redes neuronales como fueron las redes convolucionales (LeCun et al. (1989, 1995); Krizhevsky et al. (2012)) y la arquitectura tipo U-Net (Ronneberger et al. (2015)) la taxonomía de redes neuronales ha ido actualizándose de varias formas.

A la par de estas innovaciones, se han realizado multitud de avances en los algoritmos de optimización e inferencia de las redes neuronales que explotan diferentes ideas para lidiar con problemas comunes encontrados en la metodología clásica como el desvanecimiento del gradiente (Hochreiter (1998)), la falta de datos o la falta de recursos.

A continuación, se presentarán los principales campos de desarrollo del estado del arte que no deben tomarse como entes independientes sino como un conjunto de técnicas que pueden ser utilizadas simultáneamente si fuera necesario. Por ejemplo, varias arquitecturas del estado del arte hacen uso de atención en la arquitectura U-Net (Oktay et al. (2018)), o utilizan la difusión para optimizar los resultados de

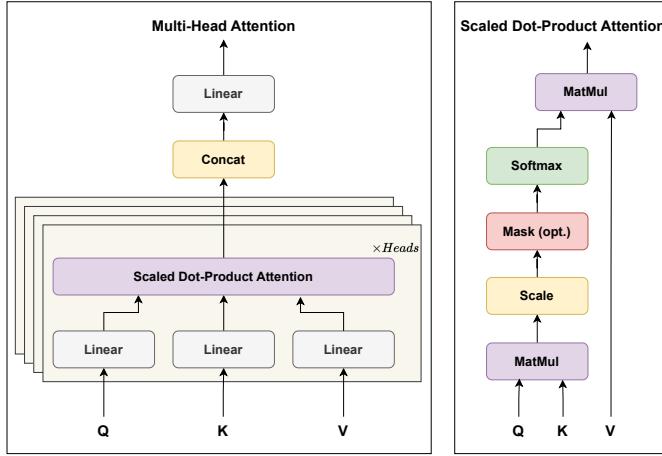


Figura 2.1: Mecanismo de atención. A la izquierda se muestra el bloque clásico de atención que toma tres entradas (Q, K, V) y devuelve una salida pasando por \times cabezas ($Heads$). A la derecha se muestra la capa de atención por producto escalar donde se realiza la atención vía *MatMul* y *Softmax*.

arquitecturas ya conocidas (Tan et al. (2023)), o reducen los parámetros de bloques convolucionales en el “generador” de un esquema de redes generativas adversarias, *GAN* (Sarker et al. (2021)).

Es necesario obtener una visión general del estado del arte para escoger continuo la metodología para un proyecto concreto. Entendiéndose el proyecto como el conjunto de datos, modelo y despliegue, se deberán ajustar todas las metodologías basándose en los avances actuales.

2.1.1. Arquitecturas

En esta subsección se exponen las vías de desarrollo más utilizadas en el estado del arte del campo de la Visión Artificial en relación con las arquitecturas de red neuronal.

2.1.1.1. Bloques de Atención

Desde la invención de los bloques de atención en 2017 (Vaswani et al. (2017)) en el contexto de predicción de datos secuenciales el mecanismo de atención, figura 2.1, se ha propagado a la mayoría de los campos de la Inteligencia Artificial.

En Visión Artificial se ha convertido en una de las vías de desarrollo con mayor impacto en los últimos años (Li et al. (2018); Dosovitskiy et al. (2020); Xie et al. (2021); Hatamizadeh et al. (2022, 2023); Cao et al. (2023); Zhang et al. (2023); Wu et al. (2024), etc.).

El mecanismo de atención, como su nombre indica, se utiliza para obtener un mapa de atención a diferentes partes de los datos de entrada, figura 2.2, imágenes

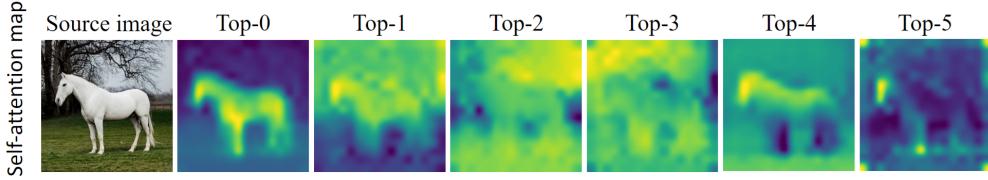


Figura 2.2: Mapa de Auto-Atención (Liu et al. (2024a)).

en este caso.

Este procedimiento se realiza en el *Scaled Dot-Product Attention*, figura 2.1, donde (Q, K, V) son tensores procedentes de capas anteriores. Q representa la “*solicitud*” de información, K representa cómo de relevante es cada punto para dicha solicitud y V contiene la información real que va a ser agregada. Si (Q, K, V) son el mismo tensor, es decir, proceden de la misma transformación previa, el proceso es denominado *Auto-Atención*. El mapa de atención se obtiene al aplicar el *Softmax* al resultado de la multiplicación matricial, *MatMul*, entre Q y K .

Esta operación usualmente se realiza N veces de forma paralelizada en las denominadas *cabezas (heads)* de la capa *Multi-Head Attention*, figura 2.1, por lo que se obtienen N mapas de atención independientes, figura 2.2. Si bien, alguna de las cabezas puede tener una alta correlación con la tarea dada puede haber otras que no la tengan (véanse las imágenes *Top-0* y *Top-3*, respectivamente, de la figura 2.2 para la tarea de segmentación de un caballo). En consecuencia, la interpretabilidad de los mapas de atención no es directa ni trivial sino que conlleva un trabajo de visualización exhaustivo y un riesgo de asunción de correlaciones espurias.

Este mecanismo tiene dos ventajas y una gran desventaja. La primera ventaja es la obtención de un mapa de regiones de interés que dota a la arquitectura de interpretabilidad dado que al visualizarlo se sabe en qué zonas se *fija* la red para tomar las decisiones. La siguiente ventaja es que el campo receptivo de la red abarca la imagen al completo ya que las operaciones *MatMul* (Multiplicación de matrices) tienen como resultado la multiplicación de cada elemento del tensor con todos los demás. Estas dos ventajas combinadas devienen en arquitecturas de altísima precisión para multitud de tareas. La gran desventaja reside en la mencionada operación *MatMul* que tiene complejidad $\mathcal{O}(n^2)$ y se realiza dos veces por capa y cabeza aumentando el coste computacional enormemente.

2.1.1.2. Modelos fundacionales y multimodalidad

Se denomina modelo fundacional a aquel modelo de Inteligencia Artificial capaz de ajustarse a una tarea global, i.e. segmentar un vasto número de categorías en una imagen.

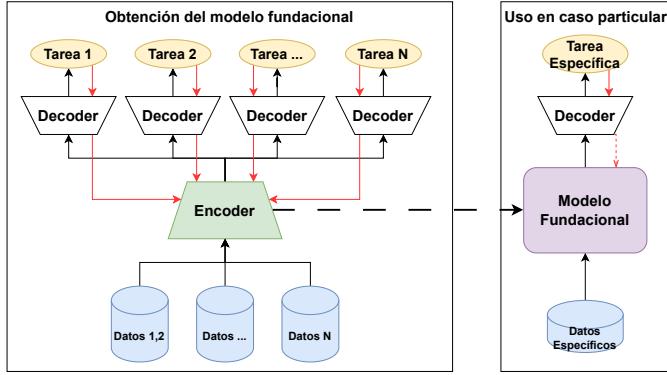


Figura 2.3: Esquema de obtención y utilización de un modelo fundacional. Las flechas sólidas negras representan un paso de mensaje, las flechas rojas representan una actualización de los pesos (discontinua si es opcional, *Fine Tuning vs. Transfer Learning, subsección 2.1.2.5*).

Estos modelos, generalmente, son desarrollados por grandes empresas o institutos de investigación ya que requieren de una enorme cantidad de datos para ser entrenados (*YOLOv8 (Reis et al. (2023))*, *GPT-4 (Achiam et al. (2023))*, *BLIP-2 (Li et al. (2023a))*, *Segment Anything (Kirillov et al. (2023))*, etc.). En esencia, cualquier modelo puede ser un modelo fundacional si se tiene acceso a los recursos necesarios.

El objetivo de los modelos fundacionales es servir como un extractor de características óptimas para una tarea general que se pueda utilizar como una capa en un modelo futuro designado para una tarea específica.

La obtención más habitual de un modelo fundacional consta de la definición de un *Encoder*, el extractor de características, y tantos *Decoders*, o predictores, como tareas se quieran definir. Por medio de técnicas como el aprendizaje por refuerzo (Li et al. (2023b)) o la transferencia de aprendizaje (Shin et al. (2016); Zhuang et al. (2020)), se ajustará el *Encoder* iterativamente a las tareas y/o conjuntos de datos definidos para obtener un modelo capaz de extraer la información más útil, figura 2.3.

La utilización de este tipo de modelos es similar a la utilización de una operación de transformación o a la utilización de una capa prediseñada en una red neuronal dependiendo de si se utiliza una optimización vía *Fine Tuning* o vía *Transfer Learning*, respectivamente, figura 2.3.

La utilización de estos modelos fundacionales asegura una extracción de características de calidad y, por tanto, una buena aproximación al problema si el modelo elegido es acorde. En contraposición, exige contar con los recursos de computación necesarios para realizar la inferencia de este modelo y las capas que formen el *Decoder* específico.

Una ventaja adicional de estos modelos es que no requieren de un gran conjunto

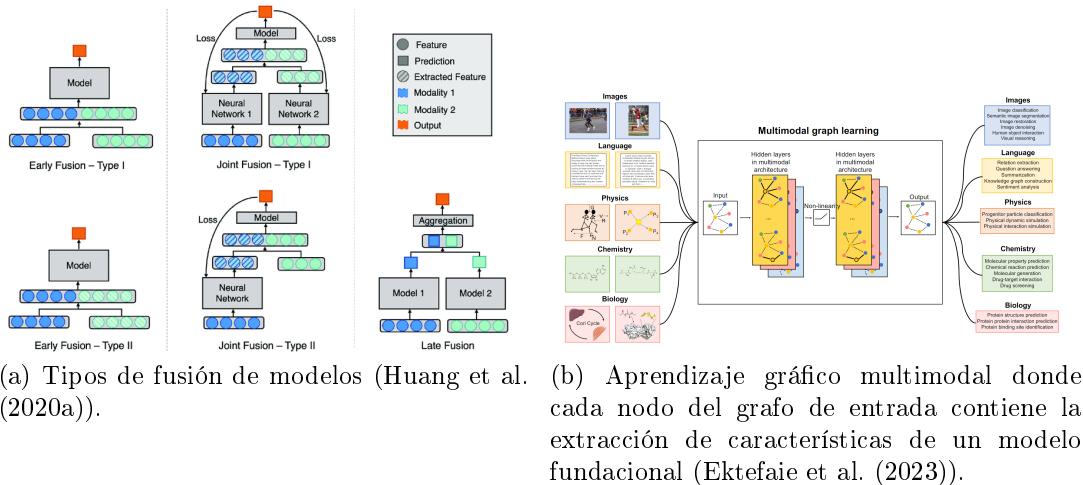


Figura 2.4: Estrategias para la multimodalidad.

de datos para ser ajustados a una nueva tarea ya que la extracción de características está aprendida.

Finalmente, estos modelos permiten construir por bloques modelos más grandes. Aprovechando esta característica y teniendo a mano todos los recursos de computación necesarios se puede llegar a obtener la *multimodalidad*.

La multimodalidad consiste en obtener modelos capaces de consumir o predecir diferentes tipos de datos (*texto, audio, video, ...*). Para ello, se combinan varios modelos *pre-entrenados* (fundacionales) utilizando técnicas como la fusión de modelos (Huang et al. (2020a)), figura 2.4a, o el aprendizaje gráfico (Ektfaie et al. (2023)), figura 2.4b.

Pese a que la multimodalidad no sea objeto de este trabajo es una vía de desarrollo prometedora y en auge, aunque también costosa y muy compleja.

2.1.1.3. Eficiencia computacional

En todos los campos de la Inteligencia Artificial se está investigando en optimizar las arquitecturas y sus componentes sin perder efectividad para así reducir el coste computacional y monetario así como el impacto medioambiental.

Esta es una vía en constante desarrollo que ha estado presente desde los inicios. A cada avance con alto coste computacional asociado le ha seguido un desarrollo que mantenga la efectividad con una drástica reducción de parámetros y requisitos.

En el contexto del mecanismo de atención, mencionado previamente, se ha evolucionado hacia:

- Prescindir de la multiplicación de matrices (Zhu et al. (2024)).
- Realizar atención flexible y jerárquica en pos de la reducción de las

dimensiones y el número de matrices multiplicadas (Li et al. (2024a); Hatamizadeh et al. (2023)).

- Linearizar el producto escalar del mecanismo de atención para la reducción de la complejidad computacional (Shen et al. (2021)).

Todas estas líneas de investigación se consideran la punta de lanza en el desarrollo de modelos de gran precisión. Estas técnicas reducen el coste del mecanismo de atención manteniendo la ventaja de la intepretabilidad. Sin embargo, la implementación es compleja y requiere de una parametrización muy específica para obtener resultados de la atención clásica con producto escalar de matrices, figura 2.1.

Por otra parte, y en relación con la vía de desarrollo del trabajo, las redes convolucionales y sus bloques constituyentes se han optimizado en diferentes aspectos. Las principales aportaciones históricas y actuales que siguen siendo utilizadas en el estado del arte se exponen a continuación y se verá su aplicación en más detalle en la sección de Métodos (3.2).

Normalización por lote, *BatchNorm*, con activación PReLU, figura C.1a

La normalización por lote, como su nombre indica, normaliza los valores resultantes de una capa por conjunto de datos de inferencia.

Este tipo de normalización permite eliminar el *bias*, o sesgo, de la capa previa ya que el resultado tras esta normalización es el mismo con un ligero desplazamiento que corrige la activación PReLU utilizando un parámetro entrenable adicional (sección B.1).

Esta estrategia es utilizada para reducir el número de parámetros en las operaciones de convolución pasando de un número de parámetros $p_0 = \omega + bias$ a $p_1 = \omega + 1$ donde ω son los pesos entrenables y el *bias* es igual al número de neuronas de la capa convolucional. Si se aplica esta metodología a todas las capas convolucionales la reducción acumulada es significativa.

Capas de convolución desagregadas, figura C.1b

Se intercambia una capa de convolución usual de tamaño de kernel k_s y stride s por tres capas de convolución consecutivas.

La primera aplica el stride, s , con $k_0 = (s \times s)$ para variar la dimensión, la segunda realiza la extracción de características utilizando $k_1 = k_s$ y, finalmente, la tercera aumenta la dimensión de los canales utilizando $k_2 = (1 \times 1)$.

Como se verá más adelante, existe una variación denominada bloque factorizado que intercambia la segunda convolución de kernel $k_s = (n \times n)$ por dos convoluciones consecutivas de kernels $k_0 = (1 \times n)$ y $k_1 = (n \times 1)$ reduciendo aún más el número de parámetros.

La justificación matemática de la reducción de parámetros obtenida se puede comprobar en la sección B.2.

Convolución por grupo con barajado de canales, figura C.1c

En ShuffleNet (Ma et al. (2018)) se adopta la idea de realizar convoluciones sobre grupos de canales que posteriormente son barajados y convolucionados en profundidad para relacionar los grupos.

Esta configuración se utiliza para aumentar la riqueza del mapa de características reduciendo el número de parámetros en un factor $f \approx C/g$ siendo C los canales de entrada y g los grupos establecidos.

Se considera este avance como el sucesor y la mejora de los bloques de exprimido y excitación (Hu et al. (2018)) en términos de efectividad y reducción de parámetros.

Regularización dropout en los canales, figura C.1d

Habitualmente se utiliza la regularización dropout sobre las neuronas que conforman una capa. En las arquitecturas eficientes se promueve la utilización de dropout sobre los canales, *Spatial Dropout*, (Lee and Lee (2020)) del mapa de características. De esta forma se impide la especialización de los canales en características concretas forzando a extraer características generales de la imagen.

Bloques residuales y *skip-connections*, figura C.2

Los bloques residuales, propuestos inicialmente en las arquitecturas ResNet (He et al. (2016)) e Inception (Szegedy et al. (2017)), utilizan una rama lineal que propaga la información a lo largo de un mismo bloque de transformación para dotar a este de contexto global.

A su vez, las *skip-connections*, originales de las arquitecturas tipo U-Net (Ronneberger et al. (2015)), conectan bloques del *Encoder* con bloques del *Decoder* estableciendo atajos de información para retener información valiosa entre secciones distantes de la red neuronal.

La combinación de ambas técnicas reduce la pérdida de información y dota de contexto a las diferentes partes de la arquitectura. Utilizando estas técnicas se reduce la carga de computación en los diferentes bloques de la red permitiendo reducir los parámetros sin impactar en la precisión.

Adicionalmente, estas conexiones sirven de atajo en la propagación de la pérdida durante el entrenamiento previniendo problemas relacionados con el desvanecimiento del gradiente (Hochreiter (1998)).

El trabajo actual hace uso de una de las últimas actualizaciones de esta vía, EFSNet (Hu and Wang (2020)), que sigue la línea de ENet (Paszke et al. (2016b)) y ShuffleNet (Zhang et al. (2018); Ma et al. (2018)) para obtener una arquitectura de

red neuronal definida por $\simeq 0,18$ millones de parámetros (suma de pesos y sesgos de las capas constituyentes) que requiere $\simeq 143$ Mb de memoria RAM dedicada para un paso de inferencia.

2.1.1.4. Otros avances

En los últimos años están ganando peso otras arquitecturas innovadoras que merecen ser mencionadas.

Modelos de mezcla de expertos, *MoE*

La arquitectura *MoE* contiene un módulo “*Enrutador*” y N bloques gemelos, *Expertos* (Liu et al. (2024b)).

La tarea del *Enrutador* es decidir qué *Experto* debe ejecutarse según los datos de entrada, mientras que la tarea del *Experto* es especializarse en la predicción de casos particulares de una tarea más global (Pavlitskaya et al. (2020)).

Estos modelos requieren de gran cantidad de datos o no habrá especialización posible pero permiten escalar el tamaño de las arquitecturas de forma drástica debido a que la inferencia únicamente computará uno de los expertos y no todos (Riquelme et al. (2021)).

Uso de grafos

Se consideran vías de desarrollo en el ámbito del aprendizaje profundo en grafos que se fundamentan en la transformación de una imagen a grafo, aplicación de una red neuronal gráfica, *GNN*, y transformación inversa (Camilus and Govindan (2012); Krzywda et al. (2022)). Se mencionan a continuación las principales formas de conversión de imagen a grafo.

En primer lugar, conversión de grupos de píxeles, *super-pixel*, a nodos enlazados con los vecinos más cercanos (Han et al. (2022)). Los grafos resultantes son homogéneos y no dirigidos con lo que se puede aplicar una GNN convolucional (Kipf and Welling (2016)).

En segundo lugar, conversiones de imágenes a hiper-grafos (Bretto and Gillibert (2005)) cuyos enlaces pueden unir una cantidad arbitraria de nodos. Estos hiper-grafos pueden ser modelizados vía Hyper-GNNs (Bai et al. (2021)).

En tercer lugar, existe una vía de utilización de *CNNs* para la conversión de imágenes en grafos de diversa tipología (Turaga et al. (2010)).

En cuarto lugar, conversiones a otros tipos como grafos dirigidos, heterogéneos, dinámicos o con signo (Zhou et al. (2020)). Estas conversiones se encuentran menos representadas en el estado del arte y presentan mayor dificultad de implementación.

Este campo se investigó en las etapas iniciales del trabajo, pero se descartó por el alto coste computacional y la falta de madurez que presenta. Aún no existe una

metodología específica acerca de la transformación de imagen a grafo o arquitectura más efectiva, aunque sí de las capas, para abordar el problema (Zhou et al. (2020)).

Modelos de Espacio de Estados Selectivo, SSSM

Los SSSM se propusieron inicialmente para la predicción de secuencias (como los bloques de atención, subsección 2.1.1.1), con la arquitectura Mamba (Gu and Dao (2023)) con la idea de eliminar la atención manteniendo el contexto global.

Los SSSM, parten de la combinación de los Modelos de Espacio de Estados, SSM, que consisten en transformar las entradas con dependencia temporal en un conjunto de salidas por medio de un estado oculto (Liu et al. (2024c)), y el mecanismo de selección que es una proyección lineal de la secuencia de entrada. En consecuencia, los SSSM aplican un SSM independientemente a cada canal emulando la atención y generando correlaciones aprendidas.

Dado que es un mecanismo que actúa sobre la última dimensión de los tensores de entrada (dependencia temporal), pronto fueron aplicados a los canales de una imagen (Liao et al. (2024); Liu et al. (2024d); Ye and Chen (2024)). De igual forma que en el caso de los bloques de atención, se presenta como un bloque fácilmente integrable en arquitecturas conocidas (paso de Swin-Unet (Cao et al. (2023)) a Swin-UMamba (Liu et al. (2024d))).

Este es un avance muy prometedor y reciente que dota a las redes neuronales de contexto total reduciendo el coste computacional.

2.1.2. Algoritmos de optimización

Los avances en algoritmos de optimización e inferencia de las arquitecturas de red neuronal cuentan con menor grado de eco mediático y, en multitud de ocasiones, son confundidos con la arquitectura. Lo cierto es que estos algoritmos tienen la capacidad de variar la convergencia de las arquitecturas de forma drástica con lo que elegir uno u otro puede ser determinante en el resultado obtenido.

2.1.2.1. Difusión guiada

Esta metodología constituye un algoritmo iterativo de eliminación de ruido (Ho et al. (2020)).

Partiendo de una imagen semilla compuesta total o parcialmente por ruido blanco elimina paso a paso, inferencia a inferencia, el ruido para obtener la imagen final, figura 2.5.

La red aprende a eliminar una cantidad de ruido determinada en cada paso deviniendo en una red que debe ejecutarse tantos pasos como ruido total se quiera eliminar.

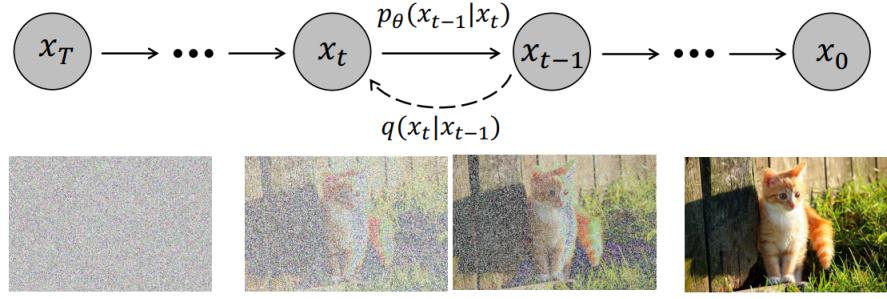


Figura 2.5: Difusión Guiada. (Chen et al. (2023a))

La eliminación de ruido está guiada por un conjunto de predictores que evaluarán si el resultado de cada inferencia se acerca o no a un objetivo dado. Por ejemplo, si se trata de un modelo de texto a imagen, el conjunto de predictores serán textos (“*a cat in the grass*” figura 2.5), si por el contrario se trata de segmentación biomédica (modelo de imagen a imagen), la eliminación de ruido deberá obtener una imagen que se corresponda con la imagen original (Tan et al. (2023)).

Con este algoritmo se obtiene una imagen nítida y acorde con el objetivo pero se incrementa el coste computacional tantas veces como pasos de eliminación de ruido se utilicen. Generalmente, esta vía se restringe a la Inteligencia Artificial Generativa (Betker et al. (2023); Esser et al. (2024)).

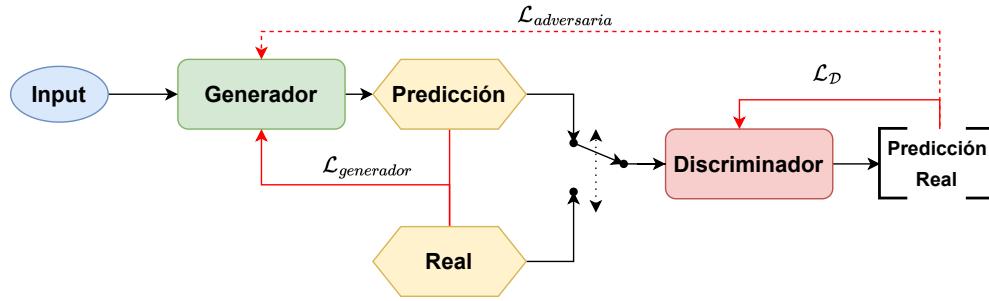
2.1.2.2. Redes Generativas Adversarias

Esta metodología se compone de una red “generadora”, $G(\cdot)$, y una “discriminadora”, $D(\cdot)$.

La red generadora debe generar el resultado final, mientras que la red discriminadora debe determinar si una imagen dada es del conjunto de los datos reales, y , o ha sido generada artificialmente (por la red generadora), $G(x)$.

La forma de utilizar esta metodología una vez definidas ambas redes es la siguiente, figura 2.6:

1. Se suministra un valor de entrada, x , a la red generadora. Esta red obtiene una predicción, $P = G(x)$, que se compara con el valor real, y , para ajustarse de forma clásica, $\mathcal{L}_{generador}(P, y)$.
2. Se suministra la predicción ó el valor real a la red discriminadora, $D(P|y)$, que debe determinar a qué conjunto pertenece. Se optimiza el discriminador con base en la respuesta dada, $\mathcal{L}_D(D(P|y), P|y)$.
3. Si se ha suministrado la predicción, $D(P)$, se optimiza la red generadora en función de cuánto peso le ha dado la red discriminadora a que dicha

Figura 2.6: Metodología *GAN*.

predicción perteneciese al conjunto de los valores reales, $\mathcal{L}_{adversaria}(D(P))$. Este paso de optimización que utiliza la pérdida adversaria da nombre a la metodología.

4. Se repite el proceso 1-3 hasta obtener una convergencia estable.

De esta forma, la red generadora aprende a optimizar dos frentes simultáneamente pero altamente correlacionados: ajustarse al problema original ($\mathcal{L}_{generador}$), y ajustarse a la distribución (a la forma) de los datos reales ($\mathcal{L}_{adversaria}$).

Este método es muy utilizado en problemas con baja cantidad de datos debido a que la pérdida adversaria asegura una convergencia con predicciones que pudieran pertenecer al conjunto de los datos originales.

En contraposición, complejiza la implementación de soluciones al tener que optimizar simultáneamente dos modelos con tres pérdidas que deben ser parametrizadas y estudiadas de forma univariante, en el mejor de los casos.

Las *GANs* (Goodfellow et al. (2014)) son una de las metodologías más comunes en problemas de Visión Artificial. El esquema generalista que propone esta metodología, figura 2.6, permite variar cada punto acorde a las necesidades del proyecto.

Se pueden encontrar *GANs* en casi todos los campos, véase redes eficientes (Sarker et al. (2021)), aprendizaje semi-supervisado (Hung et al. (2018)), *in-painting* (Dolhansky and Ferrer (2018)), super-resolución (Wang et al. (2018)) y otras muchas (Pan et al. (2019); Jabbar et al. (2021); Gui et al. (2021)).

2.1.2.3. Pasos de aprendizaje variables y decaimiento de pesos

La utilización de pasos de aprendizaje variables (Defazio et al. (2023)) y/o decaimiento en los pesos (Krogh and Hertz (1991)) añade una regularización en el entrenamiento que evita que la optimización converja a un mínimo local.

El paso de aprendizaje define cuál es la magnitud de optimización de los pesos de una red neuronal en función a una observación dada. El decaimiento de los pesos implica una regularización de la actualización en función a la magnitud de los pesos, siendo una regularización ℓ_2 para un optimizador no adaptativo como *SGD* (Loshchilov and Hutter (2017)).

Los pasos de aprendizaje variables consisten en la definición de una función que determine el valor del paso de aprendizaje en cada momento del entrenamiento. Esta función suele ser simple, como una ley de potencia (Mishra and Sarawadekar (2019)), un ciclo (Smith (2017)), o una corrección de coseno (Loshchilov and Hutter (2016)).

Esta metodología ha demostrado ser muy útil para la convergencia rápida y evitar mínimos locales, pero su implementación no puede ser a ciegas ya que puede resultar en una peor convergencia (Defazio et al. (2023)).

El decaimiento de los pesos mejora la generalización al penalizar el ajuste al ruido de la distribución del conjunto de datos (Krogh and Hertz (1991)). Debido a la regularización en función a la magnitud de los pesos, el modelo resulta en una convergencia a una distribución más generalizada. Esta generalización puede llegar a incurrir en problemas de *over-smoothing* si la penalización es alta o el periodo (*número de épocas*) de aplicación es incorrecto (Golatkar et al. (2019)).

Finalmente, la combinación de pasos de aprendizaje variables y decaimiento de pesos puede ser desfavorable por la doble regularización y la alta parametrización inducida (Lewkowycz and Gur-Ari (2020)). Adicionalmente, la necesidad de utilizar un optimizador adaptativo (Loshchilov and Hutter (2017)) y la computación del decaimiento aumentan el tiempo de entrenamiento inevitablemente.

2.1.2.4. Aprendizaje Federado

Esta es una metodología novedosa y de gran interés para el entrenamiento de redes neuronales descentralizado, distribuido y colaborativo (Bharati et al. (2022)). Permite a un conjunto de entidades compartir un modelo sin compartir los datos y, en los últimos tiempos, ha permitido implementar modelos de alta precisión en campos en los que las entidades por sí mismas no cuentan con la cantidad de datos necesaria para el entrenamiento.

Esta vía de desarrollo es prometedora ya que añade una capa de seguridad en los datos (Buenestado Cortés (2022)) y reduce la huella de carbono al deduplicar modelos.

De forma esquemática, figura 2.7, este procedimiento se implementa de la siguiente manera para N entidades que comparten una misma arquitectura de red neuronal con un mismo propósito:

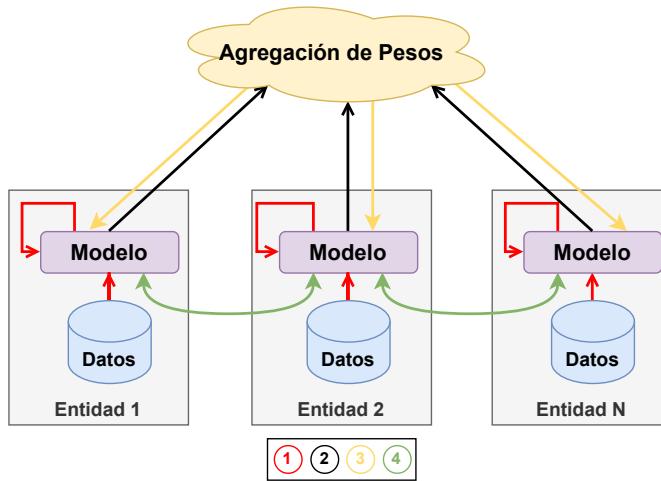


Figura 2.7: Aprendizaje Federado.

1. Cada entidad realiza un entrenamiento *in-situ* con su propio conjunto de datos siguiendo las pautas comunes definidas por la *Federación* (conjunto de entidades).
2. Las entidades envían los pesos actualizados de su modelo entrenado (*encriptados preferiblemente*) a un repositorio común.
3. Un algoritmo de agregación de pesos más o menos sencillo (Moshawrab et al. (2023)) combina las diferentes aportaciones.
4. Cada entidad descarga los pesos agregados de forma que cada entidad tenga una copia exacta del modelo.
5. Se puede repetir del paso 1. al 4. tantas veces como se quiera de forma síncrona o asíncrona.

Los algoritmos de agregación mencionados son uno de los puntos de mayor evolución del campo del Aprendizaje Federado.

En los momentos iniciales de esta metodología obtener la media de los pesos (McMahan et al. (2017)) podía llegar a ser suficiente, pero el descubrimiento de posibles ataques adversarios a la red, la necesidad de recompensas por calidad de aportación (*monetización*) y el aumento de complejidad en cuanto tamaño de la red y número de pesos implicados, han llevado a innovar rápidamente en esta línea. Consecuentemente, se ha evolucionado desde la agregación por la media a algoritmos tan complejos como *RFLPA* (Mai et al. (2024)) en 7 años.

Este emergente campo del aprendizaje máquina promete seguir creciendo para facilitar el entrenamiento colaborativo y responsable con la protección de datos.

2.1.2.5. Afinamiento y Transferencia de Aprendizaje

Se denomina afinamiento, *Fine Tuning* (Hinton and Salakhutdinov (2006)), al procedimiento por el cual se toma un modelo *pre-entrenado* (o parte del mismo), se congelan sus pesos (se impide el aprendizaje) y se le añaden una o más capas antes o después que serán ajustadas a un problema específico.

De forma sutilmente diferente, la transferencia de aprendizaje, *Transfer Learning* (Zhuang et al. (2020)), toma un modelo *pre-entrenado* (o parte del mismo) y se añaden capas al inicio o al final para ajustarse conjuntamente a un nuevo problema variando los nuevos pesos así como los originales.

Esta diferencia entre ajustar levemente los pesos del modelo original o no hacerlo tiene implicaciones de alto grado en cuanto a recursos necesarios y aplicación de la técnica.

La diferencia de aplicabilidad entre estas técnicas viene dada por la finalidad y los recursos disponibles.

En primer lugar, si la tarea objetivo es una subtarea de la dada por el modelo pre-entrenado, se trata de una especialización y el afinamiento será suficiente, en caso contrario será necesario una transferencia de aprendizaje.

En segundo lugar, si no es posible realizar un paso de entrenamiento del modelo pre-entrenado con los recursos disponibles, no será posible la transferencia de aprendizaje.

De esta forma, los campos de aplicación de estas metodologías no suelen ser solapantes.

El afinamiento, es muy utilizado en modelos fundacionales, subsección 2.1.1.2, para acotar la tarea de un modelo. Por ejemplo, *Segment Anything* (Kirillov et al. (2023)) acotado al ámbito de la medicina, MedSAM (Ma et al. (2024a)). A través de esta metodología se obtiene un nuevo modelo con mejores métricas en el caso específico ajustado. Además, utilizar el modelo fundacional a modo de capa de transformación (no aprendible) implica que no requiere de una gran cantidad de datos ni unos requisitos de computación grandes en el entrenamiento ya que la cantidad de pesos a ajustar es muy baja.

La transferencia de aprendizaje, es muy útil en la combinación y la extensión de capacidades de modelos. De nuevo, siguiendo con la analogía, si se quisiera tomar *Segment Anything* (Kirillov et al. (2023)) para describir con texto el contenido de una imagen (tarea fuera de las tareas del modelo original), se añadiría un *Decoder* o *Predictor* que tuviera texto como salida y se entrenaría el conjunto total de pesos para que la transformación de *Segment Anything* variase ligeramente en favor de la nueva tarea.

La transferencia de aprendizaje es más compleja que el afinamiento y requiere de más recursos ya que la propagación hacia atrás se efectuará sobre todo el modelo,

no únicamente en las nuevas capas. Además, el paso de aprendizaje debe ser bajo para no destruir la convergencia previa sino guiarla ligeramente incrementando el número de épocas para llegar a la convergencia.

Ambos recursos son utilizados en la actualidad en multitud de casos de uso por su versatilidad. Asimismo, la simpleza de la idea permite idear algoritmos más complejos que contengan a este como uno de sus pasos internos.

2.2. Segmentación semántica en aplicaciones biomédicas

Existe un número abrumador de aplicaciones de segmentación semántica en biomedicina.

En esta sección se detallarán las principales fuentes de datos existentes y sus tareas asociadas más comunes. Agregando a lo anterior, se presentarán las principales técnicas utilizadas en el contexto del aprendizaje profundo. Finalmente, se expondrán los métodos más actuales del caso de uso de este trabajo y se compararán con desarrollos vistos previamente para determinar diferentes posibles vías de actualización y mejora.

2.2.1. Principales fuentes de datos y tareas

Las principales fuentes de datos están relacionadas con Tomografía Computerizada (CT), Imágenes de Resonancia Magnética (MRI), Imágenes de Ultrasonido (USI), Rayos X, Tomografía de Coherencia Óptica (OCT), Tomografía por Emisión de Positrones (PET), Imágenes de Diapositivas Completas (WSI) y otras (Du et al. (2020); Siddique et al. (2021)). Todas estas técnicas de obtención de imágenes permiten la implementación de sistemas inteligentes para segmentar diferentes tipos de afecciones.

Las tareas más comunes encontradas en las diversas líneas de investigación se exponen a continuación.

Hiperintensiades de masa blanca en el cerebro

Las regiones de hiperintensidad de masa blanca en el cerebro son de interés ya que pueden tener asociado un factor de riesgo para diferentes lesiones como infartos subcorticales, accidentes cerebrovasculares isquémicos o atrofia cerebral (Wardlaw et al. (2015)). Los usualmente datos son tomados por medio de MRI (Coenen et al. (2023)). Estas imágenes llevan un preprocesamiento T1, T2 o FLAIR que anula la señal proveniente del líquido cefalorraquídeo de diferentes formas para mejorar las observaciones (Martorell et al. (2012)) y presentan un

desafío debido a su componente tridimensional al ser cada imagen una capa del volumen total del cerebro.

Tumores cerebrales

Los tumores cerebrales son estudiados a través de MRI y conlleva los mismos desafíos que el caso anterior. La investigación actual en el contexto de la inteligencia artificial se centra principalmente en la detección temprana (Khalighi et al. (2024)) ya que el método de actuación es menos invasivo y conlleva menos riesgos en las primeras fases del desarrollo.

Tumores en mamografías

Los tumores en mamografías hacen uso de USI dado que es la mejor vía de exploración del tejido adiposo subcutáneo. Este campo se explora en mayor profundidad en el subsección 2.2.3 y en las secciones 1.1, 1.2 dado que es la tarea a resolver en el trabajo.

Tumores en ovarios

Este tipo de tumores es uno de los cánceres más diagnosticados a nivel mundial y suele ser ignorado hasta sus fases finales convirtiéndose en uno con las mayores tasas de mortalidad entre las enfermedades ginecológicas (Labidi-Galy et al. (2012)). Utilizando conjuntos de datos como PLCO de USI transvaginales o de CT anotadas por expertos (Kodipalli et al. (2023)) se avanza hacia la detección temprana y categorización de este tipo de tumores (Chen et al. (2023b)).

Lesiones por infarto

El infarto es la segunda causa de muerte a nivel mundial y afecta a 15 millones de personas anualmente (Feigin et al. (2023)). Esta afección genera al $\sim 50\%$ de los supervivientes y sus allegados desfíos económicos, cognitivos y psicológicos a corto y largo plazo (Feigin et al. (2022)). Los métodos de segmentación semántica se proponen como una vía de ayuda en la rehabilitación de los supervivientes para conocer las regiones afectadas, su impacto en la salud del paciente en el futuro y las limitaciones derivadas (Malik et al. (2024)). La mayoría de los conjuntos de datos publicados se corresponden con escáneres de MRI o de CT (Luo et al. (2024)).

Vasos sanguíneos de la retina

La detección y segmentación de arterias y venas de la retina es una tarea fundamental en el análisis de imágenes biomédicas (Galdran et al. (2022)). La oclusión, deterioro o derrame de la arteria central puede conllevar problemas de visión o incluso la pérdida de la misma (Abdushkour et al. (2023)). Sobre esta línea se investiga en segmentación de filamentos finos (vasos sanguíneos) (Cervantes et al. (2023)) utilizando conjuntos de datos OCT (Galdran et al. (2022)).

Células cancerosas

Se utiliza para la detección de cáncer en diversos órganos como pulmones (Primakov et al. (2022)) y colon (Bokhorst et al. (2023)) pero pueden ser utilizados en cualquier tipo en el que se pueda extraer una muestra de tejido (Bhuiyan and Abdullah (2022); Karol et al. (2024)). Los sistemas automáticos implementados en esta vía de desarrollo hacen uso de datos tomados vía CT, OCT, PET y WSI (Bhuiyan and Abdullah (2022)) para segmentar las células cancerosas de un tejido.

Tuberculosis en los pulmones

La tuberculosis es una enfermedad contagiosa que afecta a los pulmones principalmente. En 2020 se estima que 10 millones de personas padecieron tuberculosis en el mundo entero (Organization (2020)). La detección de esta enfermedad ayuda a tener un tratamiento rápido y efectivo además de impedir la propagación. Existen dos tipologías de datos, una basada en CT que es preferida en los últimos tiempos y otra basada en Rayos X que es más asequible y común en los centros médicos (Rajaraman et al. (2021); Kim et al. (2024)).

Además de estos campos existen frentes abiertos en un enorme espectro de aplicaciones que requieren de sistemas automatizados para la segmentación de precisión y eficiente de diversas características (Ronneberger et al. (2015); Siddique et al. (2021); Azad et al. (2024)).

Estos sistemas ayudan al personal sanitario reduciendo su carga de trabajo y permiten la detección temprana de enfermedades. En contraparte, enfrentan desafíos en la accesibilidad, cantidad y calidad del dato, en la inherente complejidad del problema y en el consumo de tiempo y recursos derivados de la automatización.

2.2.2. Principales arquitecturas y soluciones

Los métodos de aprendizaje profundo utilizados para resolver las tareas presentadas en la subsección previa son diversas en función de los objetivos adicionales que se desean cumplir (eficiencia, precisión, velocidad, accesibilidad, etc.).

En cuestión de arquitecturas de red neuronal encontramos una clara prevalencia de U-Nets (Azad et al. (2024)). La U-Net (Ronneberger et al. (2015)), figura 2.8, consiste en dos partes, el *Encoder* que extrae las características de la imagen aumentando los canales y disminuyendo las dimensiones de altura y anchura, y el *Decoder* que toma la salida del *Encoder* y extiende de nuevo las dimensiones con la información dada para obtener el mapa de segmentación. Estas partes están conectados por uno o más atajos que preservan la información a distintos niveles de resolución para mantener el contexto.

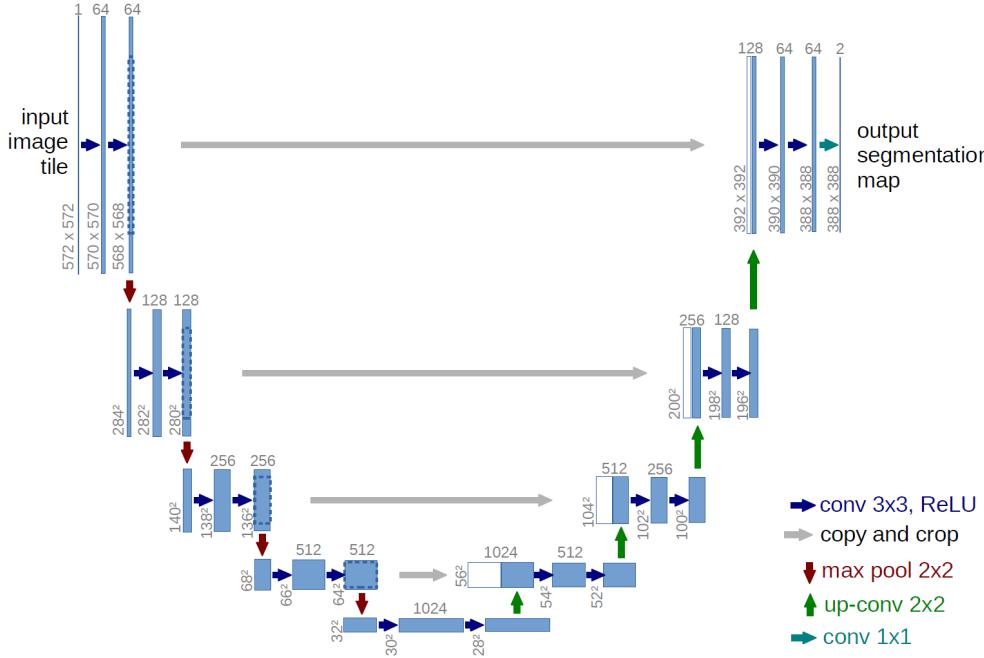


Figura 2.8: Arquitectura U-Net. (*Imagen de PRIP-Ronnenberg*)

Este tipo de arquitectura tiene origen en la CNN clásica (LeCun et al. (1995)) que únicamente contaba con la parte del *Encoder*, y mejoran la precisión de las FCN (Long et al. (2015)) cuyo *Decoder* consistía en una capa densamente conectada.

En los últimos años, las arquitecturas U-Net en el campo de la biomedicina, y en general, ha variado para adaptar diferentes bloques.

Mecanismo de atención

Como se ha comentado en la subsección 2.1.1.1, la atención dota de interpretabilidad y aumenta la precisión de una arquitectura dada a costa de incrementar los requisitos de computación necesarios. Desde la primera implementación de los *Transformers* de Visión (Dosovitskiy et al. (2020)), el número de implementaciones de esta técnica en las arquitecturas U-Net ha incrementado enormemente (Xie et al. (2023)).

Se han realizado variaciones en el *Encoder* y el *Decoder*, por separado y simultáneamente, a nivel espacial y por canales, así como en las *skip connections*. Entre las implementaciones con mejor resultado se encuentran Swin-U-Net (Cao et al. (2023)), UNETR (Hatamizadeh et al. (2022)) y TransUNet (Chen et al. (2021)), que también han experimentado variaciones posteriores.

En redes neuronales que apuestan por la eficiencia se han visto sustituciones de estos bloques de atención por SSSM (subsección 2.1.1.4), i.e. U-Mamba (Ma et al. (2024b)), o con convoluciones dilatadas y agrupadas, i.e. ESPNet (Mehta et al. (2018)), FSNet (Zhang et al. (2019)), EFSNet (Hu and Wang (2020)).

Mejora en las *skip-connections*

Las mejoras vienen en una amplia variedad de formas.

Por un lado, se propone un aumento en el número de *skip-connections* a implementar para explotar las características multi-escala de forma más efectiva (U-Net ++ (Zhou et al. (2019)), U-Net 3+ Huang et al. (2020b)).

Por otro lado, se ha probado efectuar transformaciones en la propia *skip-connection* con mecanismos de atención (Attention U-Net (Oktay et al. (2018))), recurrencia espacial (Cr-UNet Li et al. (2019)) o convoluciones recurrentes bidireccionales (BCDU-Net Azad et al. (2019)).

Cada una de estas implementaciones favorece o explota una característica diferente para ajustar mejor un tipo de tarea resultando en mejoras significativas y abriendo camino a nuevas investigaciones.

Mejora en el vector latente

Se denomina vector latente al vector de salida del *Encoder*. Si bien, en la arquitectura U-Net original se transfiere directamente al *Decoder*, hay nuevos desarrollos que realizan transformaciones sobre este vector para enriquecer el mapa de características. Los avances más significativos se pueden dividir en la introducción de la atención (SA-UNet (Guo et al. (2021))) y la representación multi-escala (ASSP-UNet (Hai et al. (2019))).

En combinación con estas variaciones de las U-Net (y otras muchas) se encuentran los algoritmos utilizados para la optimización de las arquitecturas. Son de especial interés y relevancia, los siguientes.

Redes Generativas Adversarias

La metodología GAN (subsección 2.1.2.2) es ampliamente utilizada en el campo de la biomedicina (Ashok and Gupta (2022); Iqbal et al. (2022)). Esta metodología aborda el problema de la falta de datos y ha demostrado obtener buenos resultados en multitud de tareas. En contraparte, sufre de baja reproducibilidad en comparación con la metodología clásica de optimización (Iqbal et al. (2022)).

Entre las variantes existentes, las más utilizadas son Vanilla-GAN (Goodfellow et al. (2014)), la versión clásica, y conditional-GAN (Mirza and Osindero (2014)), donde el discriminador también observa el valor de entrada del generador.

Difusión

Diversas metodologías se han implementado entre 2023 y 2024 basadas en las técnicas de difusión, subsección 2.1.2.1.

Entre estas se encuentran SDSeg (Lin et al. (2024)) que toma el modelo fundacional de Stable Diffusion (Liu et al. (2024a)) aplicado al vector latente de la U-Net. Utilizando una concatenación de los resultados de los pasos de la difusión obtiene un modelo con inferencia de un único paso que solventa el

problema del tiempo de inferencia de la difusión. Este modelo es capaz de generar predicciones a la altura del estado del arte en diversos conjuntos de datos y tareas.

Otra implementación reciente de esta metodología es MedSegDiff (Wu et al. (2023)) que utiliza una codificación posicional para los bloques de atención dinámica en función del resultado de cada paso de la difusión guiada además de un análisis frecuencial de las características para eliminar el efecto negativo del ruido de alta frecuencia. Esta metodología mejora significativamente los resultados del estado del arte y se posiciona como una de las mejores soluciones hasta la fecha.

La metodología de la difusión aplicada a U-Nets en tareas biomédicas está abierta a desarrollo en todos los aspectos pero se presenta como una de las líneas más prometedoras en los próximos años.

Profesor-Aprendiz

La metodología profesor-aprendiz es un método de destilación de aprendizaje (Abbasi et al. (2019)) que puede utilizarse para mejorar los resultados, simplificar un modelo existente o abordar la limitación de datos en un caso particular. Para ello, se toma un modelo *pre-entrenado* (el profesor) y un modelo a entrenar (el aprendiz) de forma que el aprendiz se entrena para emular (o incluso mejorar) la respuesta que daría el profesor para unos datos de entrada determinados.

Existen diversas implementaciones recientes de esta metodología que son utilizadas para obtener una solución semi-supervisada (Sun et al. (2020); Li et al. (2024b)) o para obtener un modelo ligero (Guo et al. (2023); Lou et al. (2023)).

Tanto el aprendizaje semi-supervisado como la obtención de modelos más ligeros son de gran interés en el campo de la biomedicina por la falta de datos y la optimización de recursos. En consecuencia, se espera que este campo se extienda en los próximos años. El inconveniente reside en la necesidad de un modelo *pre-entrenado*, presumiblemente fundacional con gran número de parámetros, que sirva de modelo profesor con los requisitos que conlleva.

Existen multitud de metodologías de optimización que no se mencionan en este apartado por que ya han sido previamente mencionadas y explicadas o porque se consideran herramientas más asentadas y ampliamente utilizadas, como el decaimiento de pesos.

2.2.3. Detección de tumores en ecografías mamarias

En la detección de tumores en ecografías mamarias utilizando segmentación semántica se puede observar que la principal solución viene dada por las

arquitecturas U-Net (Ilesanmi et al. (2021)), pero, como se ha visto anteriormente, se pueden establecer múltiples variaciones sobre estas.

Mecanismo de atención

La atención, de nuevo, es una de las vías más explotadas en este caso de uso. Entre las arquitecturas utilizadas se encuentran Attention U-Net (Chen et al. (2023c)) y SegNet (Vianna et al. (2021)).

Se observa en este ámbito un desfase temporal entre el origen de los avances y la aplicación a este campo (Attention U-Net: 5 años, SegNet: 4 años) que sugiere una actualización.

Por otro lado, existen avances que emulan la atención utilizando bloques más eficientes como SK-U-Net que implementa los bloques SK combinando bloques convolucionales eficientes (figura C.1a) con y sin dilatación por medio de un bloque densamente conectado y una activación sigmoide.

Avances generales en esta línea que no se han visto en el estado del arte de la segmentación de masas tumorales en ecografías mamarias son la utilización de SSSMs (subsección 2.2.2), o las convoluciones agrupadas.

Guía por Saliencia

El mapa de saliencia o, simplemente, saliencia se obtiene de la ordenación, $S(\cdot)$, valor del gradiente, ∇ , de los pesos de la red neuronal, $f_\theta(\cdot)$, para un input, X , respecto de dicho input al aplicar la función de pérdida, $\mathcal{L}(f_\theta(X), y)$, es decir, $sal = K \cdot S(\nabla_X f_\theta(X))$ donde K es una matriz constante que puede ser la identidad o una máscara (Simonyan et al. (2014)).

El mapa de saliencia se utiliza, principalmente, para dotar a la red de interpretabilidad, pero también se puede utilizar para guiar el entrenamiento (Ismail et al. (2021)). En este contexto se encuentran soluciones como SMU-Net (Ning et al. (2022)) que utiliza una red auxiliar especializada en obtener los mapas de saliencia estableciendo una aproximación inicial con la que guiar la generación de la red principal. Otro ejemplo es U-Net-SA-C (Vakanski et al. (2020)) que integra los mapas de saliencia utilizando bloques de atención como conocimiento a priori del tumor.

Esta línea de investigación ha quedado desplazada en los últimos años por los bloques de atención pero es otra vía de introducir interpretabilidad en redes que prescindan de este mecanismo. Es probable que con los nuevos desarrollos en arquitecturas como las basadas en SSSM vuelvan a tomar peso.

Segmentación Multi-Tarea

La segmentación multi-tarea consiste en construir arquitecturas con dos salidas o más que dividen la tarea general en varias sub-tareas. Por ejemplo, si la tarea es “segmentar y clasificar los tumores de una imagen” se puede convertir en

“segementar en tumor/no-tumor, salida 1, y clasificar los tumores en categorías, salida 2”.

Este abordaje es particular de la segmentación de tumores en ecografías mamarias en el que únicamente hay un objeto a segmentar de una única categoría en cada imagen. Bajo esta premisa, se han desarrollado varias arquitecturas como MT-Net (Cao et al. (2020)), MTL-Net, RMTL-Net ó SHA-MTL (Xu et al. (2023)) que dividen la tarea entre segmentación binaria y clasificación para incrementar la eficiencia, o LightBTSeg (Guo et al. (2023)) que utiliza dos profesores binarios (*maligno* y *benigno*) para destilar el modelo aprendiz final. Como se puede observar, esta es una línea de investigación en desarrollo y es aplicable de multitud de formas.

En una etapa inicial de este trabajo se indagó en esta solución pero la combinación de pérdida de clasificación, \mathcal{L}_c , y de segmentación, \mathcal{L}_s , corrompía el entrenamiento. Debido a que \mathcal{L}_c empezaba a sobreajustar antes de que \mathcal{L}_s convergiera se obtenía una segmentación pobre y una clasificación que no generalizaba fuera de los datos de entrenamiento. Además, la división de la salida en dos aumentaba el número de parámetros, pesos y sesgos, y la latencia de la red neuronal.

Arquitecturas de bajo consumo

Esta vía de desarrollo está tomando un peso relevante en los últimos años debido, principalmente, a la aplicabilidad de las soluciones.

Se pueden observar diferentes variaciones en las arquitecturas U-Net y los módulos de atención con el fin de reducir parámetros y coste computacional.

En primer lugar, el uso de bloques de exprimido y excitación (Hu et al. (2018)) se puede observar en redes como LAED-Net (Zhou et al. (2022b)), RMAU-Net (Yuan et al. (2023)) y LCMU-Net (Zhang and Niu (2023)). Estos bloques consisten en dos operaciones: (i) la operación de exprimido que produce una descripción de los canales agregando los mapas de características en las dimensiones $(H \times W)$, siendo H la altura y W la anchura, y (ii) la operación de excitación que toma la salida de la operación anterior y aplica una función por canal que modula los pesos.

En segundo lugar, se observa el reemplazo de bloques convolucionales por bloques *SCConv* y *ShiftMLP* que aplica SC-UNext (Cai et al. (2024)). Por un lado, *SCConv* consiste en una Unidad de Reconstrucción Espacial que utiliza normalización por grupo y una Unidad de Reconstrucción de Canales que utiliza una estrategia de “división - transformación - fusión” que combinadas secuencialmente abordan las redundancias espaciales y en los canales de forma eficiente. Por otro lado, el bloque *ShiftMLP* está basado en la apoptosis y división celular, donde la apoptosis se utiliza para filtrar elementos del sesgo que

no aporten información y la división se produce eligiendo aleatoriamente los pesos que se actualizarán. La combinación de ambos bloques a modo de regularización interna de la red neuronal durante el entrenamiento mejora la eficiencia reduciendo el número de parámetros.

Finalmente, de los avances generales en eficiencia computacional (subsección 2.1.1.3), se observan diversas implementaciones de: (i) bloques convolucionales eficientes con normalización por lote, figura C.1a, y bloques residuales, figura C.2a, (Zhou et al. (2022b); Yuan et al. (2023); Zhang and Niu (2023)), (ii) bloques desagregados o factorizados, figura C.1b, (Zhang and Niu (2023)).

Se puede comprobar que hay varias técnicas de reducción de parámetros, subsección 2.1.1.3, que no se han implementado aún como son: las convoluciones por grupo con barajado de canales (*que dejan obsoletos los bloques de exprimido y excitación* (Zhang et al. (2018))) y la regularización dropout espacial que aborda el problema de la Unidad de Reconstrucción de Canales del bloque *SCConv* mencionado sin añadir ningún parámetro.

Realizando una comparativa del estado del arte de la segmentación semántica para la detección de tumores en mamografías con el de la segmentación semántica en aplicaciones biomédicas (subsección 2.2.2) y con el del campo de la Visión Artificial (sección 2.1) comprobamos que es necesaria una actualización en varias líneas. Entre estas líneas se encuentran: los SSSM y los MoE (subsección 2.1.1.4), la difusión guiada (subsección 2.1.2.1) y varias técnicas de eficiencia computacional como la regularización dropout espacial y las convoluciones por grupo con barajado de canales (subsección 2.1.1.3).

En consecuencia, se decide realizar una actualización en las arquitecturas de bajo consumo haciendo uso de las técnicas más novedosas con base en la arquitectura EFSNet (Hu and Wang (2020))

Capítulo 3

Materiales y métodos

3.1. Materiales

En esta sección se exponen los materiales utilizados para la elaboración del trabajo. Se divide en tres partes: software en el que se presenta el lenguaje de programación y recursos necesarios, hardware que está dedicado al recurso físico utilizado para realizar el trabajo de computación, y datos que versa sobre el origen, calidad y metodologías correspondientes a las imágenes que han sido usadas.

3.1.1. Software

El software utilizado es el lenguaje de programación Python en la versión *3.10*. Es necesario remarcar este punto ya que versiones posteriores presentan una mejora de eficiencia y velocidad de entre un 10 % y un 60 % (Python Docs) lo que perturbaría las métricas que se presentarán en el capítulo de Resultados (4).

Las librerías: torch (versión *2.1.1+cu*), lightning (versión *2.1.0*) y opencv-python (versión *4.8.1.78*) han sido, principalmente, los módulos de funcionalidades utilizados.

En primer lugar, con torch se ha implementado la red neuronal EFSNet y el preprocesamiento de datos.

Lightning se ha utilizado para facilitar el uso de la red neuronal implementada ya que aúna una gran cantidad de métodos asociados al entrenamiento, validación, guardado y carga de modelos que aumentan la eficiencia del código y su interpretabilidad.

Finalmente, opencv-python ha sido utilizado para la carga de datos y la realización de transformaciones y visualizaciones sencillas.

3.1.2. Hardware

Dado el objetivo del trabajo de facilitar el uso de la herramienta presentada en ordenadores convencionales, se ha hecho uso de uno (aunque optimizado para la ciencia de datos), en concreto, un portátil *Asus Predator* con un procesador *intel-core i7*, memoria RAM de 32Gb y una GPU *NVidia GeForce 4060 RTX* de 8Gb de memoria dedicada y 16Gb de memoria total.

El entrenamiento se ha realizado haciendo uso de la tarjeta gráfica debido a la capacidad de paralelización que presenta. Debido a la configuración interna se puede observar a lo largo de la memoria que los parámetros que afectan a la red neuronal como: dimensiones de las imágenes, tamaño del *batch* o neuronas por capa son potencias de 2 (2^x).

3.1.3. Datos

En esa subsección se presenta el origen de los datos, cualidades y transformaciones efectuadas para el uso en el caso presentado.

3.1.3.1. Origen

El conjunto de datos utilizado se denomina *Dataset of Breast Ultrasound Images* (Al-Dhabyani et al. (2020)), *BUSI*, que es de uso libre, ha sido aceptado por revisión de pares para su utilización con fines de investigación y se puede encontrar en *BUSI-Dataset*. La distribución de categorías se muestra en la tabla 3.1.

Tabla 3.1: Número de imágenes por categoría.

Categoría	Número de imágenes
Normal	133
Benigno	437
Maligno	210
Total	780

Existen cuatro principales datasets públicos de imágenes de ecografías mamarias anotadas por expertos:

1. *BrEaST* (Pawłowska et al. (2024)): se trata de un dataset que contiene las mismas clases, es más novedoso y asegura una calidad de etiquetado excelente, pero cuenta con 256 ejemplos siendo 4 normales, 154 tumores benignos y 98 malignos que pueden resultar insuficientes para el ajuste de

una red neuronal, además esta es la versión mejorada de 2024 del *BrEaST* previo que contenía mayor cantidad de fallos y no estaba disponible cuando se realizó el procedimiento de ingesta de datos (2023).

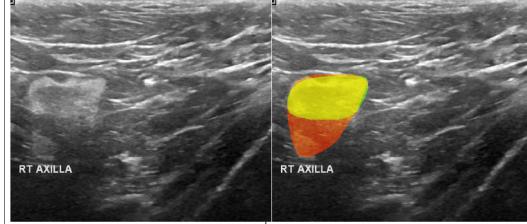
2. *Open Access Series of Breast Ultrasonic Data, OASBUD*, (Piotrzkowska-Wróblewska et al. (2017)): que únicamente cuenta con 100 imágenes siendo 52 casos malignos y 48 benignos correspondientes a 78 mujeres con lo que el problema mencionado previamente es mayor y, además, pueden existir duplicidades.
3. *UDIAT* (Yap et al. (2018)): que cuenta con un total de 163 imágenes, 110 con presencia de tumores benignos y 53 restantes con presencia de tumores malignos. Cada imagen contiene una única masa tumoral con lo que puede incurrir en sesgos.
4. *Dataset of Breast Ultrasound Images* (Al-Dhabyani et al. (2020)), *BUSI*: conjunto de datos compuesto por imágenes en escala de grises de ecografías mamarias asociadas a máscaras binarias y etiquetadas según su categoría correspondiente. Las imágenes corresponden a 600 impresiones de ecografías de mujeres voluntarias de entre 25 y 75 años de edad utilizando los aparatos *LOGIQ E9 ultrasound system* y *LOGIQ E0 Agile ultrasound system*. Se cuenta con 780 imágenes de anchura y altura variable con categorías: normal, benigno y maligno.

Los datasets mencionados no han sido utilizados simultáneamente debido a que las imágenes fueron tomadas con aparatos diferentes y se utilizó un preprocesamiento específico en cada caso. Los aparatos de ultrasonidos tienen, como una de sus características principales, la frecuencia del ultrasonido producido que define la nitidez y profundidad a la que se puede observar. Al no contar con distribuciones de etiquetas ni metodologías similares se ha optado por evitar utilizar datos combinados dado que la red neuronal puede converger al aprendizaje de características de las imágenes que no se correspondan con la determinación de una región tumoral como se ha observado en otros casos relacionados con las técnicas de aprendizaje profundo (Ribeiro et al. (2016)).

La elección del dataset *BUSI* sobre los demás se debe, principalmente, a la cantidad de muestras, pero, también a la alta manejabilidad de las mismas. La etiquetación de imágenes por medio de la organización en carpetas y utilización de títulos permite mayor agilidad dado que la carga de imágenes se realiza a través del enrutamiento que lleva implícita la localización y nombre.

Tabla 3.2: Dimensiones de las imágenes originales.

Dimension	Mínimo	Máximo	Media
Altura	310	719	501.45
Anchura	190	1048	615.68

Figura 3.1: Imagen duplicada con texto (*RT AXILLA*) y discrepancia en las máscaras (*verde, rojo*). (Pawlowska et al. (2023))

3.1.3.2. Cualidades

El conjunto de datos se compone por 780 imágenes en escala de grises de anchura arbitraria, tabla 3.2, y 798 máscaras binarias asociadas por lo que hay imágenes con más de una máscara. En total se cuenta con 252.31 millones de píxeles de los cuales un 7.33 % son correspondientes a tumores, 3.71 % son de la clase *benigno* y 3.62 % son de la clase *maligno*. El formato de las imágenes es png.

Las imágenes están organizadas en carpetas cuyo nombre corresponde con la clase. Cada imagen está nombrada tal que “*clase (id).png*” mientras que las máscaras “*clase (id)_mask.png*” o “*clase (id)_mask_1.png*” de forma que es sencillo concretar qué máscaras se asocian a qué imágenes y su clase. Las máscaras tienen valor 0 o valor 1 para cada uno de sus píxeles donde 0 siempre es *normal* y 1 será *benigno* o *maligno* dependiendo del nombre de la imagen o carpeta en la que se encuentre. No hay imágenes que contengan al mismo tiempo máscaras correspondientes a *benigno* y a *maligno*. Cada imagen está únicamente relacionada a una clase.

Existen aberraciones en las imágenes (véase figura 3.1) como textos, superposiciones, pictogramas, medidas y efecto doppler que pueden perturbar el ajuste de modelos. Además, hay imágenes duplicadas con discrepancias en las máscaras asociadas, otras correspondientes a la axila y otras con objetos como agujas de biopsia (Pawlowska et al. (2023)).

3.1.3.3. Transformaciones iniciales

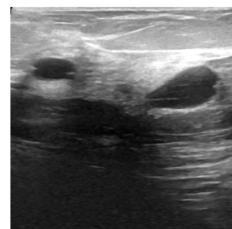
En primer lugar, se toman las máscaras que corresponden a una misma imagen y se suman sus valores formando una nueva máscara, con lo cual se obtiene una

relación uno a uno de imágenes y máscaras, y se aumenta ligeramente la proporción de casos positivos en el conjunto.

Finalmente se redimensionan las imágenes a (512×512) para estandarizar las dimensiones del conjunto y facilitar el entrenamiento de la red neuronal. Este procedimiento se puede encontrar en las figuras 3.2, 3.3.



(a) Imagen original.



(b) Imagen redimensionada.

Figura 3.2: Redimensionado de imagen.

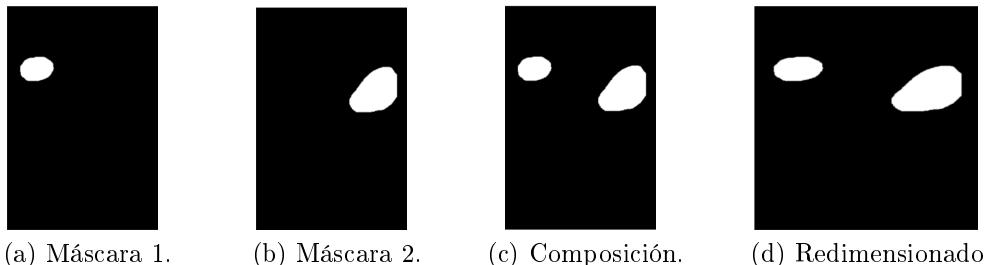


Figura 3.3: Procesamiento de máscaras.

Aplicando estas transformaciones se cuenta con 204.47 millones de píxeles de los cuales un 7.83 % son correspondientes a tumores, 3.86 % son de la clase *benigno* y 3.97 % son de la clase *maligno*. Sigue siendo un conjunto tremadamente desbalanceado de imágenes, pero ya está correctamente estandarizado a nivel de dimensiones y duplicidades.

Es importante unir las máscaras que corresponden a una misma imagen ya que, de otro modo, se pueden incurrir en dos problemas principales: sobrerepresentación de ciertas imágenes y consecuente sobreajuste de las mismas, y ejemplos con discrepancias en el objetivo (un par (imagen, máscara) tiene un conjunto de positivos y la otra otro diferente) que resultan en un mal ajuste para ambos casos o en la obtención del valor medio de ambos.

Durante el entrenamiento y validación de la red neuronal, se aplican distintas transformaciones consecutivas a las imágenes tras estas transformaciones iniciales para realizar un aumento de datos (subsección 3.2.2.1).

3.2. Métodos

Los métodos utilizados se pueden dividir en la red neuronal implementada, EFSNet, que es una función que transforma una imagen en una máscara, y los métodos utilizados para que la función se ajuste al problema definido, i.e. algoritmos de optimización.

3.2.1. EFSNet

La EFSNet, *Efficient Fast Semantic Segmentation Network*, (Hu and Wang (2020)), figura 3.4, es una red neuronal convolucional diseñada para obtener inferencias de alta calidad con bajo coste computacional en el ámbito de la segmentación semántica de imágenes. Los principales puntos para la reducción de número de parámetros son: las convoluciones agrupadas, los bloques factorizados, los bloques de convoluciones barajadas dilatadas consecutivas (*CSDC*) y los módulos de aumento de resolución con uso de ShuffleNet.

Esta arquitectura aporta a la investigación en redes neuronales para segmentación semántica rápida los bloques CSDC y los módulos de aumento de resolución. De manera global, es una arquitectura tipo U-Net (Ronneberger et al. (2015)) con características de la ResNet (He et al. (2016)).

Se puede dividir en: “*Encoder*” en el que se realiza la extracción de características de la imagen por medio de operaciones de convolución consecutivas que reducen la altura y anchura de la imagen y aumentan el número de canales, “*Decoder*” que realiza la operación inversa al “*Encoder*”, y la capa de salida “*Output*” que deviene en la predicción final.

La reducción de parámetros en una red neuronal conlleva asociado el problema de la falta de información debido a la reducción en la extracción de características. Para abordar este problema EFSNet explota varias técnicas de retención de información como son: la dilatación en las convoluciones que aumenta el campo receptivo, la regularización de canales vía *Dropout* (Lee and Lee (2020)) que evita la especialización de canales, la utilización de conexiones residuales que retiene la información original o las *skip-connections* propias de la U-Net que establecen un *shortcut* entre el *Encoder* y el *Decoder*, entre otras que se explican más adelante. Este abordaje conlleva un número de operaciones concatenadas mayor que en las redes neuronales habituales derivando en otro potencial problema que es el desvanecimiento del gradiente. Las conexiones residuales, las convoluciones agrupadas, las capas de normalización por lote (*BatchNorm*) (Ioffe and Szegedy (2015)) seguidas por las activaciones PReLU (He et al. (2015)) alivian este problema (Shah et al. (2016)).

La EFSNet implementada, figura 3.4, tiene 178887 parámetros (suma de

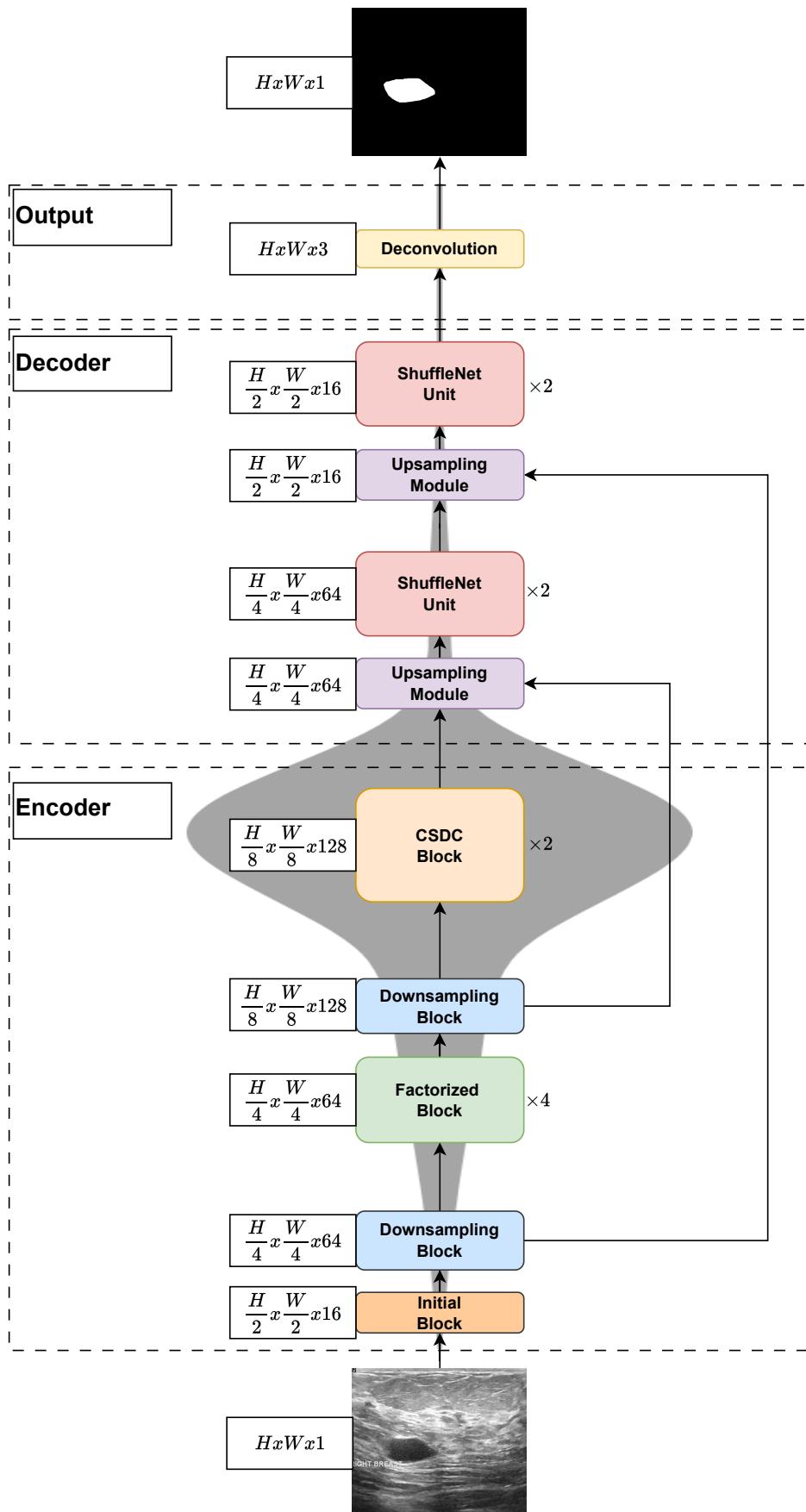


Figura 3.4: EFSNet con parámetros por capa en de fondo en gris.

Tabla 3.3: Parámetros de EFSNet implementada por bloque. Dim se utiliza como abreviatura de dimensiones y params como abreviatura de parámetros. Se omite la dimensión de batch en las columnas de dimensiones por redundancia.

Bloque	Dim. de entrada	Dim. de salida	# params	% params
Initial	(1 × 256 × 256)	(16 × 128 × 128)	166	0.09
Downsampling 1	(16 × 128 × 128)	(64 × 64 × 64)	5699	3.19
Factorized (x4)	(64 × 64 × 64)	(64 × 64 × 64)	15244	8.52
Downsampling 2	(64 × 64 × 64)	(128 × 32 × 32)	30339	16.96
CSDC (x2)	(128 × 32 × 32)	(128 × 32 × 32)	109548	61.26
Upsampling 1	(128 × 32 × 32)	(64 × 64 × 64)	12418	6.94
ShuffleNet 1 (x2)	(64 × 64 × 64)	(64 × 64 × 64)	3232	1.81
Upsampling 2	(64 × 64 × 64)	(16 × 128 × 128)	1314	0.73
ShuffleNet 2 (x2)	(16 × 128 × 128)	(16 × 128 × 128)	456	0.25
ConvTranspose2d	(16 × 128 × 128)	(3 × 256 × 256)	435	0.24
EFSNet Total	(1 × 256 × 256)	(3 × 256 × 256)	178887	100.00

pesos y sesgos de cada capa) totales (tabla 3.3) todos entrenables, para una imagen de dimensiones (256, 256, 1). El cómputo de un paso de inferencia consume 142,67 *Mb* de memoria RAM. El artículo original asegura que esta arquitectura tiene 173*k* parámetros para una imagen de entrada de (1024, 512, 3) (Hu and Wang (2020)) en contraposición con los 179099 parámetros que tendría la arquitectura implementada en este trabajo para dichas dimensiones de entrada. No se ha detectado error en la implementación y tampoco se ha encontrado su implementación para comparar por lo que se considera que esta diferencia se debe a la omisión de información en el artículo y no a mala fe de los investigadores.

En las figuras de las siguientes subsecciones se tomará el siguiente criterio de representación de los *shortcuts* y caminos principales: los *shortcuts*, con trasnformaciones lineales, se presentan en la rama de la izquierda mientras que los caminos principales con mayor número de operaciones se presentarán en la rama derecha. Muchas de las imágenes presentadas están basadas en el artículo de FSSNet (Zhang et al. (2019)) en el que se especifica que se aplica *SpatialDropout* al final de cada rama que realice una transformación no lineal (rama derecha) antes de la operación de combinación, pero no se observa en las imágenes. Se ha decidido mostrar estas capas en este trabajo para facilitar posibles implementaciones.

3.2.1.1. Bloque Convolucional

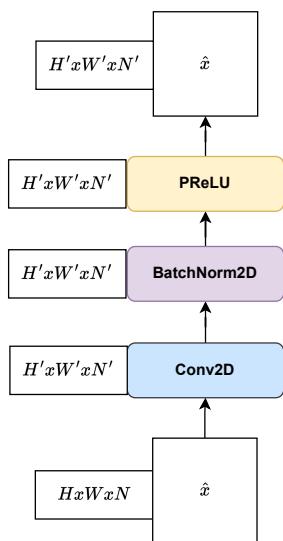


Figura 3.5: Bloque Convolucional.

El bloque convolucional, figura 3.5, es ampliamente utilizado a lo largo de la arquitectura de toda la red. Se verá en el bloque inicial (subsección 3.2.1.2), en el bloque de reducción de resolución (subsección 3.2.1.3), en el bloque factorizado (subsección 3.2.1.4), en el bloque SDC (subsección 3.2.1.6), en el módulo de aumento de resolución (subsección 3.2.1.7) y en la unidad ShuffleNet (subsección 3.2.1.8).

Este bloque se trata de una transformación que disminuye o mantiene la altura y anchura de la imagen y aumenta o mantiene los canales. Está compuesto por el triplete: Convolución2D-BatchNorm2D-PReLU, y tiene tres principales razones de ser así.

En primer lugar, históricamente el diseño de arquitecturas de redes convolucionales han convergido hacia la utilización de bloques definidos por la combinación de capas: *convolucion - normalización - activación - pooling - dropout*, aunque en este caso el pooling no es utilizado y el dropout se realiza fuera del bloque.

En segundo lugar, utilizar *BatchNorm* (Ioffe and Szegedy (2015)) permite prescindir del *bias* de la capa de convolución (sección B.1). Por bloque es despreciable pero en conjunto pueden llegar a ser miles de parámetros de reducción asociados al sesgo.

En tercer lugar, la activación *PReLU* (He et al. (2015)) o *Parametrized ReLU* aborda el problema del desvanecimiento del gradiente y añade un parámetro entrenable que se corresponde con la pendiente de la parte negativa, \mathbb{R}^- , que permite difundir valores negativos con un umbral definido por los requisitos de la propagación hacia atrás del entrenamiento.

3.2.1.2. Bloque Inicial

El Bloque Inicial de la EFSNet es heredado de la ENet (Paszke et al. (2016b)) e inspirado por Inception (Szegedy et al. (2017)). Este bloque se compone por dos ramas que se concatenan al final (3.6).

La rama de la izquierda efectúa una reducción vía *MaxPooling* de los canales de la imagen de entrada para obtener las mayores intensidades de cada conjunto de píxeles en un kernel de dimensión (2×2) , en el caso del trabajo actual únicamente hay un canal mientras que en el orginal esta operación actúa sobre los tres canales

RGB (Hu and Wang (2020)). Esta rama es utilizada para retener información y agilizar el entrenamiento.

Simultáneamente, la rama de la derecha aplica un bloque convolucional (subsección 3.2.1.1) con kernel (3×3) y stride 2 con $N = 16 - C$ filtros con C los canales de entrada. De esta forma se obtiene una salida, \hat{x} , con la mitad del tamaño original y 16 canales tras la concatenación de ambas ramas. Los canales de salida deben ser potencia de 2 para maximizar la potencia de la paralelización utilizando GPU, pero no se ha encontrado una razón de por qué 16 (Paszke et al. (2016b)) y no 4, 8 o 32.

En los artículos en los que se implementa este bloque inicial (Paszke et al. (2016b); Hu and Wang (2020)) las imágenes de entrada tienen tres canales por lo que las intensidades de los canales tendrán mayor peso en el mensaje propagado que en el caso actual que únicamente tiene uno. Para un único canal este bloque es aproximadamente igual a utilizar únicamente la rama dada por el bloque de convolución con 16 canales.

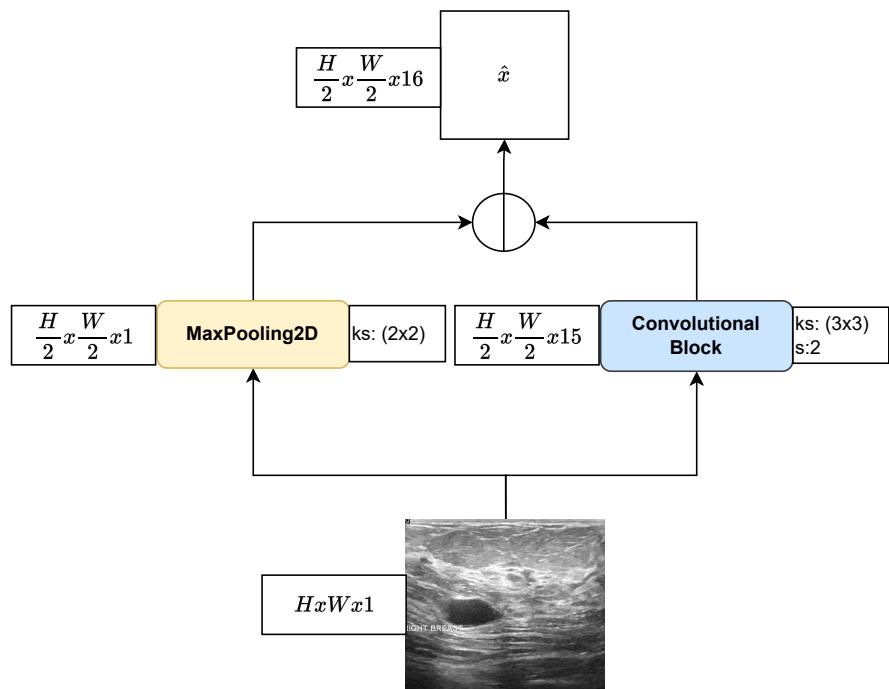


Figura 3.6: Bloque inicial.

3.2.1.3. Bloque *Downsampling*

El Bloque *Downsampling*, o de reducción de resolución, es tomado de la arquitectura FSSNet (Zhang et al. (2019)).

Como se puede observar en la figura 3.7, es un bloque residual formado por una parte lineal, rama izquierda, y una parte no lineal, rama derecha.

La parte lineal se compone por una agregación *MaxPooling* de tamaño de kernel

(2×2) seguida por un bloque convolucional (subsección 3.2.1.1) de tamaño de kernel (1×1) y sin activación.

La rama no lineal utiliza una primera convolución de tamaño de kernel (2×2) con *stride* 2 para reducir la anchura y altura de la entrada a la mitad de forma que la siguiente capa convolucional de kernel (3×3) tenga menos parámetros. Esta segunda capa es la encargada de realizar la extracción de características junto con la tercera convolución de kernel (1×1) y sin activación que multiplica en un factor 4 el tamaño de los canales que obtiene de entrada.

Finalmente, se aplica un *SpatialDropout* a la rama de la derecha, se suman las salidas de ambas ramas y se aplica una activación *PReLU*.

Las dos características principales de este bloque son: la rama no lineal que preserva la información, y los consecutivos bloques convolucionales que primero reducen las dimensiones, después extraen características y terminan por extender los canales reduciendo el coste computacional sobre el denominado *bottleneck* de la ResNet (He et al. (2016)).

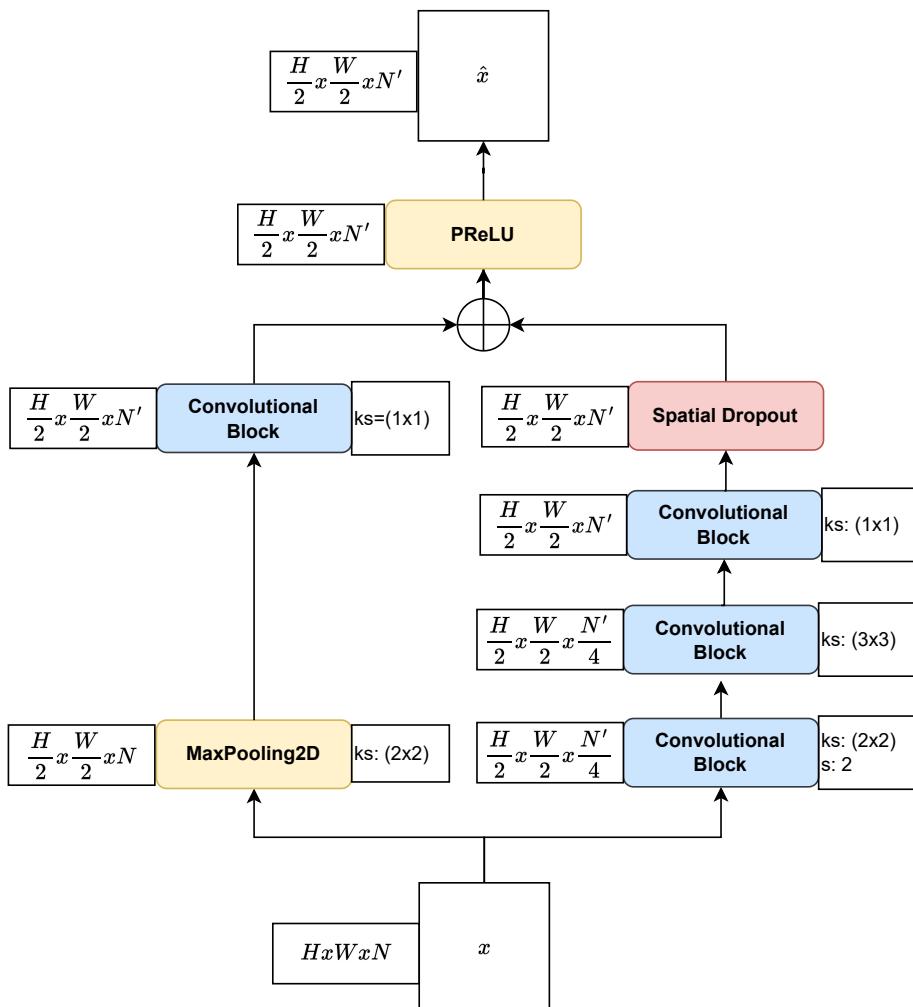


Figura 3.7: Bloque de reducción de dimensionalidad.

3.2.1.4. Bloque Factorizado

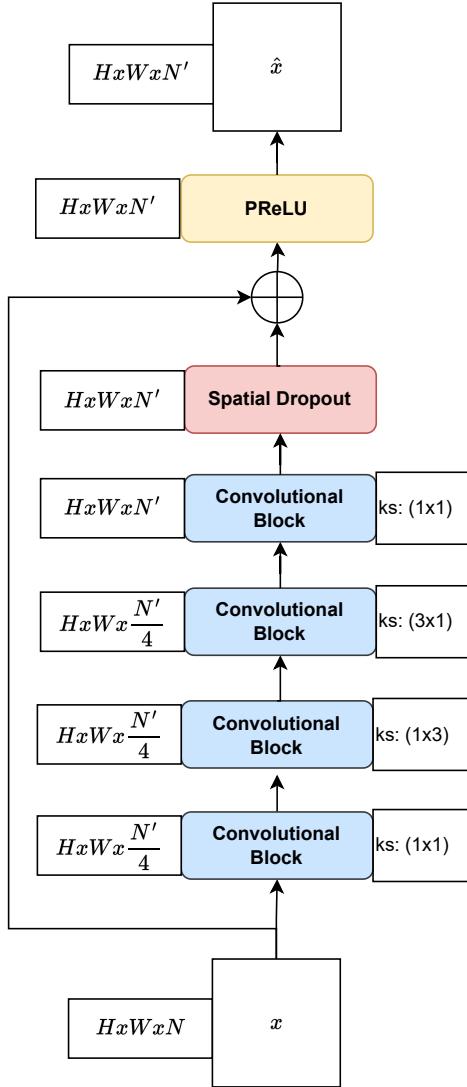


Figura 3.8: Bloque Factorizado.

extiende los canales seguido de un *SpatialDropout*. Como se puede observar en la figura 3.8 la rama lineal no realiza ninguna transformación. Finalmente, se suman ambos caminos.

En FFSNet (Zhang et al. (2019)) y EFSNet (Hu and Wang (2020)) se utilizan cuatro bloques factorizados seguidos que se denominan *Bloques Factorizados Continuos* y sirven para la extracción de características de bajo nivel. La definición de, exactamente, cuatro bloques factorizados no está justificada y se asume por el comentario de los autores respecto a las métricas de validación (Zhang et al. (2019)) que ha sido determinado experimentalmente a fuerza bruta.

El bloque factorizado, figura 3.8, se utiliza para la extracción ágil de características. Utiliza un *truco* similar al Bloque *Downsampling* (subsección 3.2.1.3) en su rama no lineal, pero añade la descomposición del Bloque convolucional de kernel (3×3) en dos bloques convolucionales de kernels (1×3) y (3×1) sin activación entre medias (siendo indiferente el orden). Este añadido aumenta la velocidad de entrenamiento y lleva una operación similar. Esta descomposición en factores es la que le da el nombre al bloque.

Por lo demás, esta rama se comporta igual que la rama no lineal del bloque de reducción de resolución (subsección 3.2.1.3) si mantuviésemos las dimensiones de altura y anchura: el primer bloque convolucional de kernel (1×1) se utiliza para reducir el número de parámetros del siguiente bloque, continua por los mencionados bloques de convolución factorizados, y finaliza con un bloque de convolución de kernel (1×1) sin activación que

3.2.1.5. Bloque CSDC

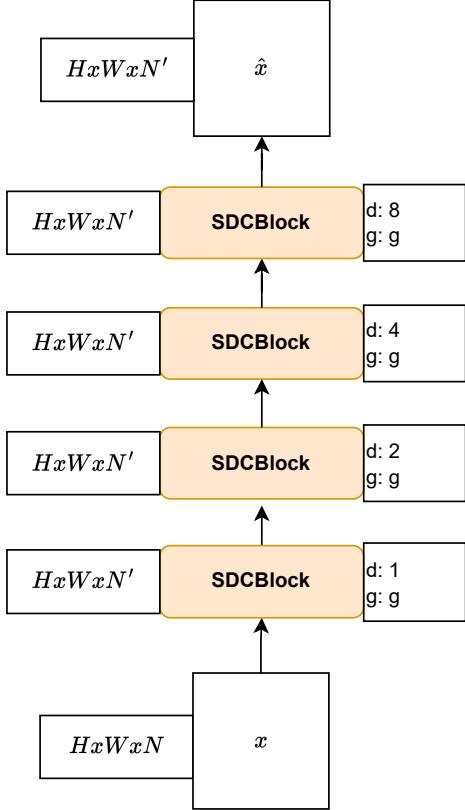


Figura 3.9: Bloque CSDC.

otras arquitecturas de actualidad basadas en transformers este grueso de parámetros lo lleva la capa o capas de atención y la estrategia es similar: obtener información del contexto por medio de aumentar el campo receptivo y establecer relaciones entre zonas alejadas de la imagen.

En el artículo de EFSNet (Hu and Wang (2020)) se obtiene empíricamente la combinación óptima de número de bloques SDC y número de bloques CSDC que son $N_{SDC} = 4$, $N_{CSDC} = 2$ así como el número de grupos que maximizan la precisión, $g = 2$. Más allá del empirismo no hay una razón de por qué se deben utilizar estos parámetros.

El campo receptivo en el último bloque SDC es de (31×31) , el 97% del input para el caso actual con dimensiones de entrada (256×256) , ya que se utiliza una dilatación $d = 8$ con un kernel de tamaño (3×3) sobre la imagen reducida en un factor 8.

El bloque CSDC, *Continuous Shuffled Dilated Convolution*, figura 3.9, se compone por cuatro bloques SDC (subsección 3.2.1.6) y se define por un número de grupos, g . Los bloques SDC internos siguen la relación 2^k , $k \in [0, K - 1]$ para los valores de la dilatación donde K es el número de bloques que componen el bloque CSDC. Este bloque CSDC obtiene información multi-escala basándose en el principio “*reducir - transformar - expandir*” (Hu and Wang (2020)), expandiendo el campo receptivo y disminuyendo el número de parámetros necesarios. Estos bloques representan el grueso de los parámetros utilizados en la arquitectura, un 61.26 %, con lo que es el bloque principal de extracción de información de la arquitectura. En

3.2.1.6. Bloque SDC

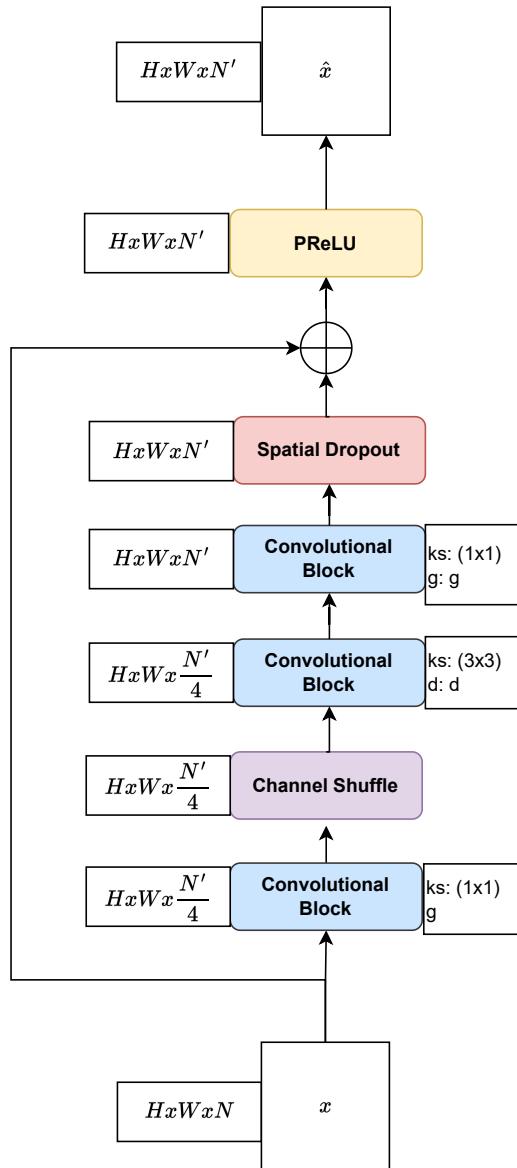


Figura 3.10: Bloque SDC.

El bloque SDC, *Shuffled Dilated Convolution*, es el principal avance propuesto por EFSNet (Hu and Wang (2020)). Se basa en la *ShuffleNet Unit* (Ma et al. (2018)) y continua la idea del bloque *downsampling* (subsección 3.2.1.3) y del bloque factorizado (subsección 3.2.1.4) pero añade tres características: el *barajado* de canales (*Channel-Shuffle*), las convoluciones dilatadas (Yu and Koltun (2015)) y las convoluciones por grupo. Este bloque puede observarse esquemáticamente en la figura 3.10.

Se compone por una rama lineal que sirve de retención de información y una rama no lineal compuesta por un bloque convolucional de grupo con kernel de tamaño (1×1) para aumentar los canales de la entrada seguida de un barajado de canales en g grupos, un bloque convolucional de tamaño (3×3) y dilatación d sin activación que se encarga de la extracción de características, un bloque convolucional de grupo con kernel de tamaño (1×1) sin activación para aumentar los canales y un dropout final en los canales.

En el paso final se suman ambas ramas y se aplica una activación *PReLU*. El motivo de añadir la capa de *Channel Shuffle* es abordar un potencial problema derivado de la convolución de grupo: los canales de cada grupo únicamente estén relacionados con los canales del mismo grupo.

3.2.1.7. Módulo de *Upsampling*

El módulo de *Upsampling*, o de aumento de resolución, es uno de los dos componentes del *Decoder* y es uno de los avances propuestos por EFSNet (Hu and Wang (2020)) para disminuir el número de parámetros sin perder precisión. Se compone de dos ramas con output similares uno procedente del camino principal, rama izquierda en la figura 3.11, y otro procedente de una *skip connection* extraída de un bloque *downsampling* (subsección 3.2.1.3) del *Encoder*, rama derecha en la figura 3.11. El propósito del módulo es aumentar en un factor 2 las dimensiones de altura y anchura de los mapas de entrada.

En la rama de la izquierda, se utiliza un bloque de convolución de kernel (1×1) para reducir el tamaño de los canales del mapa de características, posteriormente se aplica un algoritmo de interpolación bilineal para aumentar en un factor dos la altura y anchura del mapa.

En la rama derecha se aplica un bloque convolucional de kernel (1×1) para disminuir el tamaño de los canales del mapa dado por el bloque *downsampling* y, posteriormente, se aplica una deconvolución de kernel (2×2) y *stride* 2 para aumentar la resolución en un factor 2.

Finalmente, se concatentan las salidas de ambas ramas y se aplica una activación *PReLU* para obtener el nuevo mapa de caracrerísticas. La concatenación de las salidas de ambas ramas previenen la obtención de predicciones con ruido debido a la retención de información de la *skip connection*.

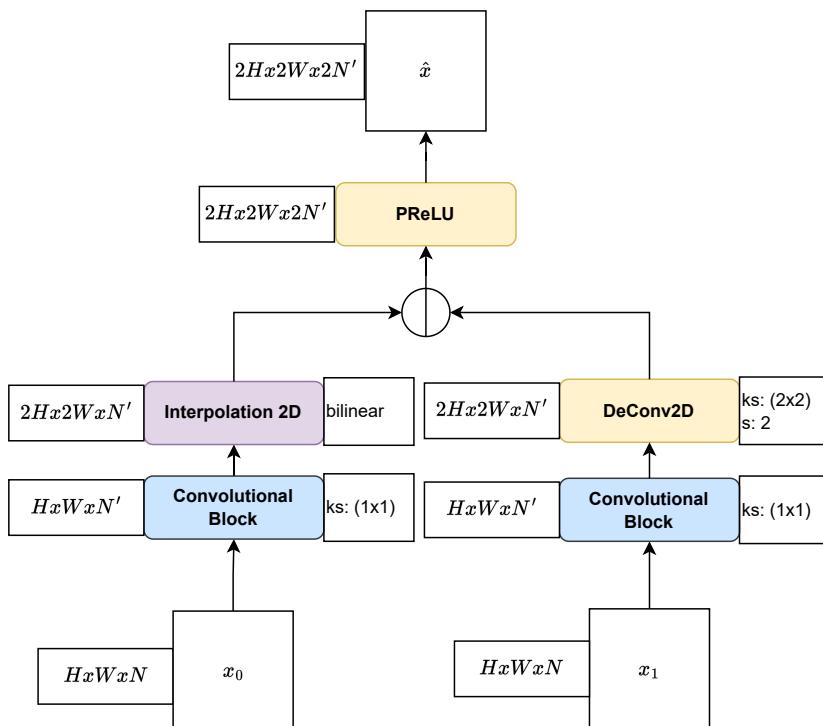


Figura 3.11: Módulo de *Upsampling*.

3.2.1.8. *ShuffleNet Unit*

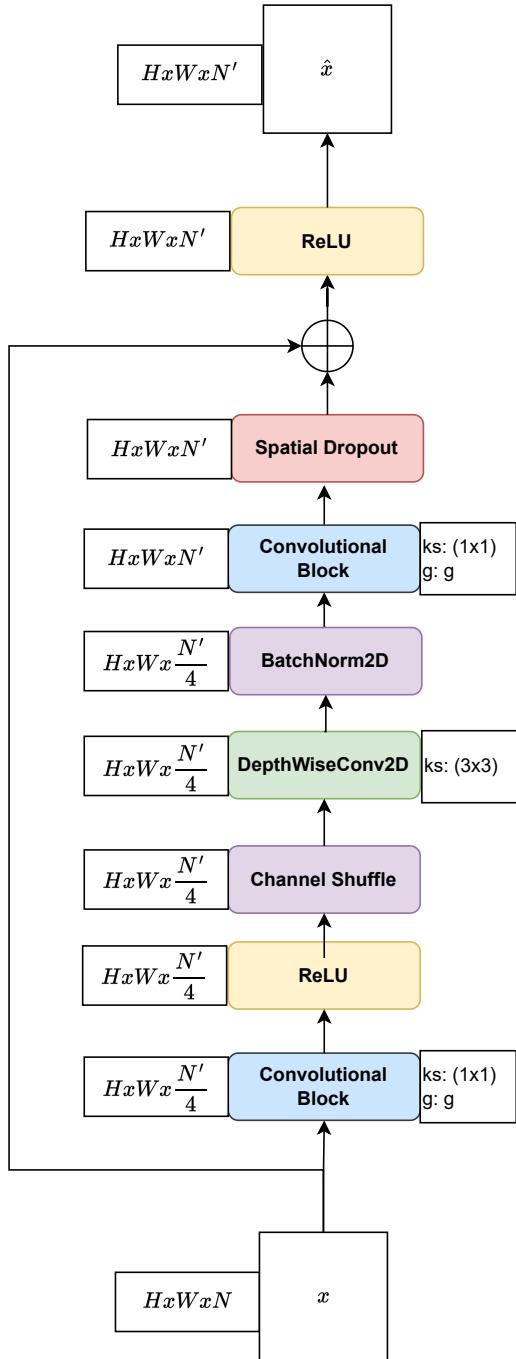


Figura 3.12: *ShuffleNet Unit*.

canales.

La capa *Channel Shuffle* no se utiliza tras el último bloque convolucional dado que no varía la precisión resultante y evitarlo agiliza la inferencia (Hu and Wang (2020)).

La *ShuffleNet Unit* se utiliza tras los pasos de aumento de resolución en el *Decoder*. Es un bloque de tipo residual con una rama sin transformaciones a la izquierda que se suma con la rama no lineal de la derecha. El esquema de este bloque puede observarse en la figura 3.12.

Está basado en la arquitectura *bottleneck unit*(He et al. (2016)) pero añade tres variaciones importantes.

En primer lugar, la primera y última convolución de kernel (1×1) que sirven para variar la dimensión del mapa de características se realizan por grupos para crear mapas más amplios reduciendo el coste.

En segundo lugar, se añade una capa de barajado de canales *Channel Shuffle* tras la primera convolución para relacionar toda la información y no únicamente la de los canales correspondientes a cada grupo.

En tercer lugar, se intercambia la capa convolucional de extracción de características con kernel (3×3) por una capa convolucional en profundidad que reduce el coste computacional.

Adicionalmente, se hace uso de noramlización por lote (*BatchNorm*) para eliminar los parámetros del *bias* de las capas convolucionales, activación *ReLU*, y regularización *dropout* en los

3.2.1.9. Predicción

La predicción, como se observa en el diagrama de la figura 3.4, se trata de una convolución transpuesta o deconvolución de kernel (3×3) con *stride* 2, tres canales de salida, normal, benigno y maligno, sin activación y acotada entre los valores -10 y 10. Con esta capa se obtiene el mapa de probabilidades con el que se obtiene la segmentación final.

Dada la naturaleza del problema, se puede establecer esta capa final, x_s , para obtener dos clases: *normal*, *tumor* y utilizar el vector latente resultante del *Encoder* para realizar la clasificación a nivel de imagen, x_c , entre: *normal*, *benigno* y *maligno*. Dado que x_s tiene dimensiones (H, W) , altura y anchura respectivamente, con valor binario y x_c es un entero la predicción final vendrá dada por $\hat{x} = x_c \cdot x_s$ y tendrá las mismas dimensiones de x_s . Esta forma de predicción añade parámetros pero alivia el problema de la mezcla de clases, i.e. un mismo tumor con regiones pertenecientes a *benigno-maligno*, presente en la segmentación usual. Se ha probado esta vía pero los resultados han sido considerablemente peores ya que x_c converge y se sobreajusta antes de que x_s comience a converger. Este desfase provoca que el ajuste de la segmentación sea corrompido.

3.2.2. Entrenamiento y validación

Esta sección se dedica a definir el método de entrenamiento desde el procesado de los datos a la optimización de la red neuronal.

Para el entrenamiento y validación de la EFSNet se ha dividido el conjunto de 780 imágenes en un conjunto de 680 para entrenamiento y 100 para validación que representan el 87 y el 13 % de la muestra. La división se ha realizado aleatoriamente utilizando *scikit-learn* con una semilla, *seed* = 0. La distribución de imágenes por categoría y conjunto de datos se puede observar en la tabla 3.4.

Tabla 3.4: Distribución de imágenes por categoría y conjunto de datos.

Categoría	Total	Entrenamiento	Validación
Normal	133	115	18
Benigno	437	382	54
Maligno	210	182	28
Total	780	680	100

Es evidente que hay una sobre-representación de imágenes de la clase *benigno*. A nivel de píxel, que es la escala objetivo, la sobre-representación se da en la

clase *normal* manteniéndose un balance entre las clases *benigno* y *maligno*. Esto motiva a pensar que los tumores de tipo *maligno* tienden a ser regiones mayores, mientras que los *benignos* no suponen una proporción muy grande de la imagen. De estas hipótesis se pueden intuir varios *mínimos locales* a los que la red puede converger como: determinación de la clase del tumor en base al área ocupada o preponderancia de la clase *benigno* en las regiones tumorales. Más adelante, se comprobará si se ha convergido alguno de estos puntos.

Para evitar el sobreajuste y lidiar con el desbalance de clases se han utilizado varias técnicas:

- Se han definido un conjunto de transformaciones afines aleatorizadas sobre los datos del conjunto de entrenamiento.
- Se ha pesado la importancia de cada clase en la función de pérdidas en función de su representación.
- Se utiliza un paso de aprendizaje, *learning rate*, variable en función del número de épocas realizadas.

3.2.2.1. Aumento de imágenes

Se han utilizado una serie de transformaciones aleatorias que se aplican a las imágenes al ser suministradas a la red para el entrenamiento, pero no para la validación. Estas transformaciones, figura C.3, en orden de aplicación, son las siguientes:

1. Redimensionado ó recorte elegido aleatoriamente a partir de una distribución unifome con probabilidad $p = 0,5$ a $(256, 256, 1)$.
2. Rotación con respecto al centro con un ángulo aleatorio $\theta \in (-\pi, \pi) rad$.
3. Cambio de brillo en un factor $b \in (0,75, 1,25)$.
4. Cambio de contraste en un factor $c \in (0,75, 1,25)$.
5. Volteado horizontal con una probabilidad de $p_{vh} = 0,25$.
6. Volteado vertical con una probabilidad $p_{vv} = 0,25$.

Sobre las máscaras únicamente se aplica el redimensionado, recorte, rotación y volteado vertical y horizontal por consistencia.

Este método de aumento de imágenes colleva la principal ventaja de engrosar el número de muestras disponibles de forma aleatoria evitando en gran medida un posible problema de *sobre-ajuste*. De igual forma, aporta varias desventajas como:

Tabla 3.5: Pesos CCE.

Categoría	Teórico	Normalizado	Utilizado
Normal	1.51	1.00	1.00
Benigno	17.27	11.45	5.00
Maligno	17.24	11.43	5.00

- El redimensionado de la imagen reduce su resolución eliminando detalles potencialmente valiosos.
- El recorte aleatorio puede dejar fuera las regiones tumorales de interés aumentando el desbalance favorable a la clase *normal*
- La rotación añade zonas totalmente oscuras y planas que siempre pertenecen a la clase *normal* pudiendo sesgar la predicción de zonas similares a esta clase.
- Los cambios de brillo y contraste pueden suavizar detalles útiles, enmascarar bordes de las regiones de interés o amplificar diferencias de color.

3.2.2.2. Función de pérdida

La función de pérdida utilizada es la denominada “*Entropía cruzada categórica*”, *CCE* (ecuación B.8), la extensión a N categorías de la función de “*Entropía cruzada binaria*”, *BCE* (Mannor et al. (2005)), con peso por clase (*torch wCCE*).

Los pesos se han definido basándose en la metodología propuesta por ENet (Paszke et al. (2016a)) tal que

$$\omega_c = \frac{1}{\ln(c + p_{class})} \quad (3.1)$$

donde c es una constante arbitraria que se define en el artículo como 1,02 y p_{class} es la representación de la clase.

En el conjunto de datos se obtienen los siguientes valores al aplicar la fórmula 3.1 sobre el conjunto de entrenamiento sin transformar: (*normal*, *benigno*, *maligno*) = (1.51, 17.27, 17.24) que normalizados por la clase dominante quedan: (*normal*, *benigno*, *maligno*) = (1, 11.45, 11.43). Empíricamente se ha observado que estos valores son excesivamente *agresivos* preponderando las clases menos representadas. Disminuyendo los valores gradualmente se ha llegado a los pesos (*normal*, *benigno*, *maligno*) = (1, 5, 5) utilizados finalmente, tabla 3.5. Esta diferencia se puede deber a la aumentación de imágenes vista en la subsección anterior (3.2.2.1).

3.2.2.3. Optimización

En esta subsección se detalla el algoritmo utilizado para la optimización de los pesos de la EFSNet. Se sigue el procedimiento mostrado en el artículo de referencia (Hu and Wang (2020)). Se utiliza un optimizador Adam (Kingma and Ba (2014)) con ratio de aprendizaje (*learning rate*) inicial $lr_0 = 0,0005$, decaimiento exponencial del momento de primer orden $\beta_1 = 0,9$ y de segundo orden $\beta_2 = 0,999$, epsilon $\varepsilon = 1 \cdot 10^{-8}$ por estabilidad numérica, decaimiento en los pesos $w_d = 0$ y sin variante *AMSGRAD* (Reddi et al. (2019)). Los pesos de la red se inicializan con la función Kaiming Uniform, también llamada He Uniform (He et al. (2015)), y se establece la probabilidad de *dropout* $p_{dr} = 0,2$.

En comparación con la propuesta original del artículo (Hu and Wang (2020)) aquí se especifican β_2 y ε . Además, en este trabajo se ha decidido utilizar $w_d = 0$ y no el valor propuesto $w_d = 0,0004$ dado que realentiza significativamente el entrenamiento y puede producir resultados negativos con un ratio de aprendizaje variable (Lewkowycz and Gur-Ari (2020)).

Se define un decaimiento del ratio de aprendizaje polinómico, *PolynomialLR*, con *power* = 0,9 y número máximo de épocas $e_{max} = 5000$. Adicionalmente, se establece un *checkpoint* basado en el mínimo de la función de pérdida en validación que, en este caso, se obtuvo en la época 312.

Las primeras 1000 épocas de entrenamiento se pueden visualizar en la figura 3.13, a partir de dicho punto la pérdida en el conjunto de entrenamiento y validación convergen a un valor “*estable*”. De la figura 3.13 no se puede extraer una conclusión sobre el *sobre-ajuste* ya que las imágenes del conjunto de entrenamiento han sido aumentadas artificialmente mientras que las del conjunto de validación no. Por este motivo no podemos afirmar que, aproximadamente, en la época 450 el modelo empiece a sobre-ajustar. Además esta diferencia en las transformaciones de datos explica la diferencia en la velocidad de convergencia entre los conjuntos. El modelo termina con un valor de la función de pérdida $CCE_w = 0,2734$.

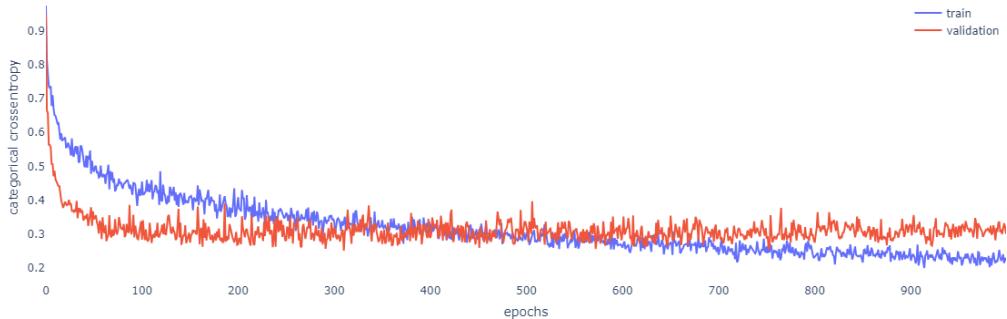


Figura 3.13: Entrenamiento de EFSNet. Se muestran las primeras 1000 épocas.

Capítulo 4

Resultados

En este capítulo se realiza un análisis de resultados.

En primer lugar, se mostrarán las métricas obtenidas en *accuracy* (ecuación B.9), *dsc* (*Dice Score Coefficient*) (ecuación B.10) y *mIoU* (*mean Intersection over Union*) (ecuación B.11) para cada subconjunto así como las matrices de confusión asociadas a los conjuntos de entrenamiento y validación. Con este análisis se comprobará el sobre-ajuste obtenido y una visión general de la eficiencia del modelo.

En segundo lugar, se hará un análisis visual de los aciertos y errores más comunes en el conjunto de validación detallando las características compartidas en cada caso y tratando de razonar las causas de estos.

Finalmente, se comprueba la viabilidad de utilizar esta red neuronal para inferencia en tiempo real.

4.1. Métricas

En las tablas de métricas, tabla 4.1, se puede observar un ligero sobre-ajuste del modelo en el conjunto de entrenamiento aunque es tan leve que puede deberse a una mera fluctuación estadística. Por otro lado, observamos un buen ajuste siendo casi perfecto para la clase normal y mejor para la clase *benigno* que para la clase *maligno*.

Comparando la tabla para las clases: normal, benigno y maligno, tabla 4.1a, con la tabla con la agrupación de benigno y maligno como *tumor*, tabla 4.1b se puede observar una mejora en la segunda. Esta diferencia se debe, principalmente, a tumores bien segmentados pero clasificados incorrectamente.

En las tablas 4.2a y 4.2b se pueden comparar los resultados con otras redes neuronales utilizadas para segmentación semántica *normal/tumor* ya que en la combinación de *normal/benigno/maligno* no hay unicidad o fiabilidad en las

Tabla 4.1: Tablas de métricas.

(a) Clases: normal, benigno, maligno.

Conjunto	\bar{dsc}	$accuracy$	$mIoU$	IoU_{normal}	$IoU_{benigno}$	$IoU_{maligno}$
Entrenamiento	71.42	94.24	59.13	94.69	43.06	39.66
Validación	70.44	94.06	58.14	94.22	43.45	36.74

(b) Clases: normal, tumor.

Conjunto	dsc_T	$accuracy_T$	$mIoU_T$	IoU_{normal}	IoU_{tumor}
Entrenamiento	82.53	94.97	73.15	94.68	51.63
Validación	79.72	94.49	69.79	94.22	45.36

métricas de evaluación utilizadas en los artículos (Xu et al. (2021); Cho et al. (2022); Bansal (2023)). Se expone también el tamaño de los modelos debido a que en este trabajo se ha sacrificado precisión por velocidad y se asumía de principio un peor rendimiento.

En la comparativa observamos que teniendo entre dos y tres órdenes de magnitud menos en el número de parámetros respecto a los demás modelos se obtienen resultados similares. El mayor desvío se encuentra en $accuracy$ (ecuación B.9), mientras que el coeficiente de similaridad de Dice, \bar{dsc} (ecuación B.10), y la intersección sobre unidad, $mIoU$ (ecuación B.11), incluso mejoran modelos como Deeplabv3+ (Chen et al. (2018)), UResNet (Ronneberger et al. (2015)), FCN (Long et al. (2015)) o MTL-Net (Xu et al. (2023)). El alto valor en DSC e IoU indican que, si se detecta el tumor la forma resultante de la máscara se ajusta correctamente a la región tumoral (con respecto a los demás modelos).

Cabe destacar que no existe un criterio de evaluación unificado sobre el conjunto de datos a predecir ni sobre las métricas a utilizar. Debido a la disparidad entre evaluar el modelo únicamente en casos con tumor o en todos los casos los resultados se dividen en dos tablas:

1. Tabla 4.2a: correspondiente a un conjunto de imágenes con y sin tumores presentes.
2. Tabla 4.2b: correspondiente al subconjunto de imágenes que contienen tumores.

Personalmente, opino que es necesaria la introducción de imágenes sin tumores en la evaluación ya que representan la mayoría de los casos a los que se expondrá el modelo si se utiliza en tiempo real.

En las tablas 4.2a y 4.2b se presentan las métricas más comúnmente encontradas aplicadas sobre el conjunto de validación (definido en la subsección 3.2.2). Se asume una distribución estadística similar de categorías en los conjuntos de datos utilizados en los diversos artículos utilizados para realizar la que se observan en la comparativa.

Observando las tablas 4.2a y 4.2b se puede comprobar que hay una gran disparidad en los resultados para una misma arquitectura. Se obtienen peores métricas si hay imágenes sin tumores (*véase DeepLabv3+, PSPNet, FCN y EFSNET(nuestro)*). Esto sugiere que los modelos presentados en la tabla 4.2b “esperan” que exista tumor en la imagen.

En arquitecturas pesadas ($Mparams > 10$) es necesario el análisis previo de un experto y, una vez detectado el tumor, realizar la segmentación de su forma o se podrá incurrir en un falso positivo, que no es de tal gravedad como un falso negativo. En contraposición, en redes livianas como EFSNet, LCMUNet, UneXt y SC-UNeXt no es necesario un análisis previo pero sí una evaluación continua de las predicciones. Por ende, las redes neuronales pesadas y livianas no deben utilizarse con el mismo propósito. Las arquitecturas pesadas son excelentes para definir la forma de un tumor previamente detectado y agilizar el trabajo de enmascarado de la imagen. Las arquitecturas livianas son ideales para la detección automática de tumores. En ningún caso, estas arquitecturas pueden ser utilizadas sin supervisión de un experto.

Por otro lado, observamos que EFSNet es una arquitectura muy competente en comparación con modelos varios órdenes de magnitud mayores que explotan otros avances, a priori más potentes, del estado del arte como los bloques de atención. Además nuestra arquitectura mantiene mejor las métricas entre los conjuntos con y sin imágenes sin tumores presentes (tablas 4.2a y 4.2b) que otras arquitecturas (DeepLabv3+, PSPNet, FCN).

Se presentan en la figura C.4 las matrices de confusión para los conjuntos de entrenamiento y validación. Se han dividido adicionalmente en predicción binaria y multiclas. De esta forma se pueden extraer otras métricas en caso de ser necesario.

Los resultados numéricos obtenidos concuerdan con las suposiciones. Se extrae del análisis y la comparación que EFSNet es una red neuronal de bajo coste computacional con una capacidad comparable a las arquitecturas más relevantes en el estado del arte de detección de tumores en ecografías mamarias vía segmentación semántica. Además, el reducido número de parámetros permite inferencia en tiempo real con baja latencia evitando la necesidad de una pre-selección de imágenes y permitiendo el acceso a cualquier individuo o entidad.

Tabla 4.2: Comparación de modelos en las clases: *normal*, *tumor* (Xu et al. (2021); Cho et al. (2022); Bansal (2023); Cai et al. (2024)). Se resaltan en negrita los mejores valores del total. Los guiones indican que no se ha encontrado el valor correspondiente para la arquitectura.

(a) Imágenes con y sin tumores presentes.

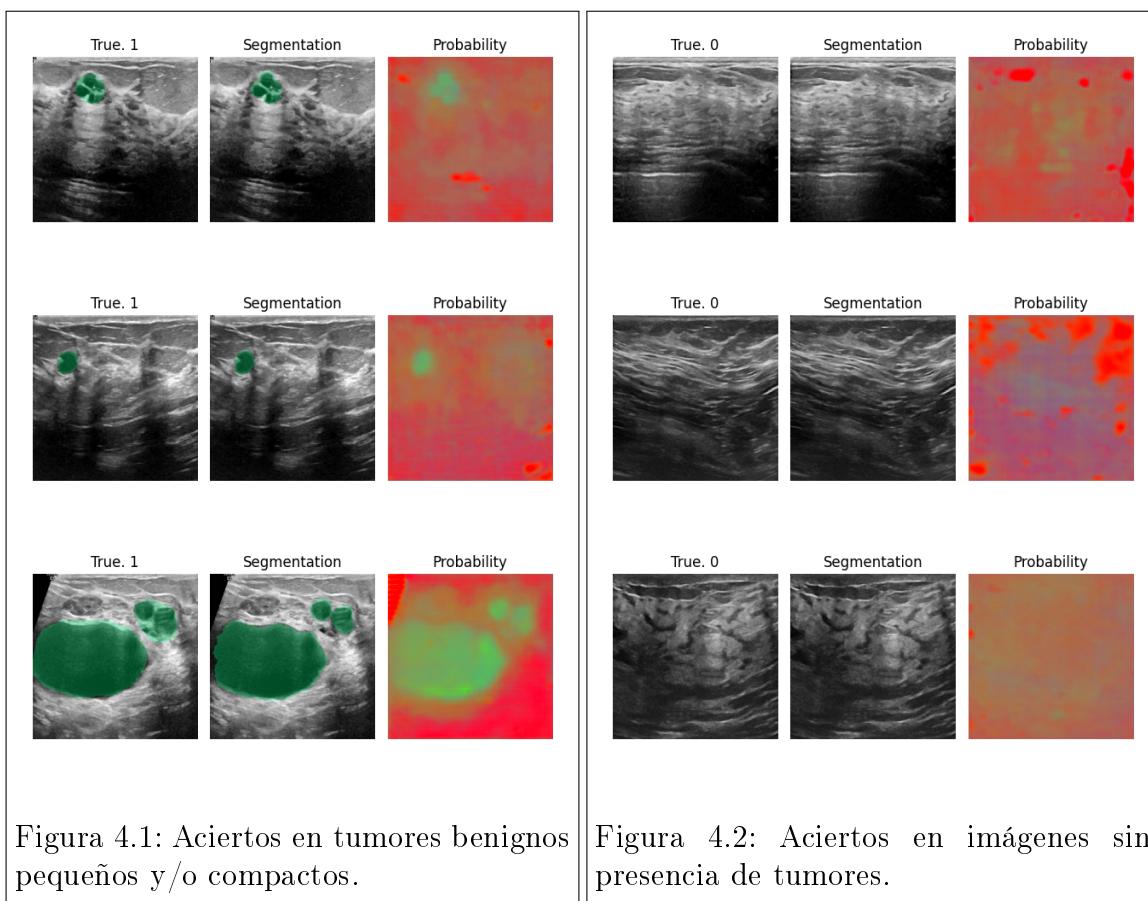
Arquitectura	\bar{dsc}_T	$\bar{accuracy}_T$	\bar{mIoU}_T	# params [M]
EFSNet (nuestro)	79.72	94.49	69.79	0.179
SC-UNeXt (Cai et al. (2024))	75.29	97.09	63.66	1.460
UNeXt (Valanarasu and Patel (2022))	72.97	95.29	61.15	1.470
SegNet (Badrinarayanan et al. (2017))	68.84	96.08	60.64	14.700
UNet (Ronneberger et al. (2015))	69.05	96.17	61.49	34.000
Deeplabv3+ (Chen et al. (2018))	72.25	96.26	64.88	59.339
TransUnet (Chen et al. (2021))	68.94	93.97	57.79	66.810
PSPNet (Zhao et al. (2017))	73.12	96.22	65.23	70.296
FCN (Long et al. (2015))	68.12	95.85	59.33	134.270
Attention U-Net (Oktay et al. (2018))	71.71	96.20	64.42	-
RFS-UNet (Cho et al. (2022))	77.04	96.77	70.47	-

(b) Imágenes únicamente con presencia de tumores (82/100).

Arquitectura	\bar{dsc}_T	$\bar{accuracy}_T$	\bar{mIoU}_T	# params [M]
EFSNet (nuestro)	80.83	94.99	70.86	0.179
LCMUNet (Zhang and Niu (2023))	81.43	-	63.99	1.490
Deeplabv3+ (Chen et al. (2018))	79.14	96.37	70.51	59.339
PSPNet (Zhao et al. (2017))	78.48	96.31	70.11	70.296
UResNet (Ronneberger et al. (2015))	77.98	96.06	69.65	93.501
MTL-Net (Xu et al. (2023))	77.76	96.18	69.33	93.505
RMIL-Net (Xu et al. (2023))	80.04	96.41	71.93	93.506
MTL-COSA (Xu et al. (2023))	78.90	96.35	70.65	109.241
FCN (Long et al. (2015))	76.87	96.08	67.05	134.270
SHA-MTL (Xu et al. (2023))	81.42	95.56	-	$\mathcal{O}(100)$
Residual U-Net (He et al. (2016))	84.81	88.08	-	$\mathcal{O}(100)$
SK-U-Net (Byra et al. (2020))	81.20	94.44	-	$\mathcal{O}(100)$
ESTAN (Shareef et al. (2022))	78.00	-	70.00	-
SaTransformer (Zhang et al. (2023))	86.34	-	83.12	-

4.2. Aciertos

En esta sección se visualizan conjuntos de imágenes con características compartidas cuyas máscaras predichas se han acercado más a las originales. Estos conjuntos son: imágenes con tumores benignos pequeños y/o compactos (figura 4.1) e imágenes sin presencia de tumores (figura 4.2). Evidentemente no todas las imágenes de estos conjuntos han sido predichas a la perfección pero sí una gran proporción de las mismas.



Con base en el mapa de probabilidad de las figuras 4.1 y 4.2 donde (*RGB*) viene dado por el valor que da la red neuronal a las clases *normal*, *benigno* y *maligno*, respectivamente, se puede extraer que la probabilidad asignada es una combinación de todos los estados en su base (*véase el caso de imágenes sin presencia de tumores, figura 4.2*) y únicamente se determina en pequeñas regiones de interés con forma de círculo o filamentos generalmente oscuros en el interior e iluminados en el exterior.

Se puede concebir conceptualmente esta red como un proceso por el cuál se extraen las regiones de interés y, posteriormente, se clasifican. Por lo tanto, EFSNet durante el entrenamiento ha convergido a buscar la forma o formas más habituales de un tumor en una ecografía mamaria y, posteriormente, evaluarla dejando el

resto de la imagen en un estado combinado de clases con predominancia de la clase *normal*. Este proceso es el que se busca con la introducción del Bloque CSDC (subsección 3.2.1.5) que está diseñado para emular un mecanismo de atención. Al enfocar la atención en puntos específicos se facilita la tarea del *Decoder* permitiendo reducir el número de parámetros, pero, por otro lado se añaden errores adicionales como encontrar regiones de interés que no lo son o no encontrar las regiones de interés reales. Además, dado que las redes neuronales tienden a la media en sus predicciones es complejo generalizar la forma de los tumores y, por tanto, predecir filamentos y bordes irregulares.

4.3. Errores

En cuanto a errores comunes hay también una serie de conjuntos de imágenes cuyas inferencias son erróneas por motivos similares. Destacaremos tres de ellas: agujeros y derrames en la inferencia de tumores muy grandes y/u homogéneos, mezcla e intercambio de clases *benigno*, *maligno* en la máscara final y falsos errores.

Los tumores muy grandes y los tumores cuya textura es muy plana (homogéneos) determinan un desafío para EFSNet. La convergencia de la red neuronal determina que las zonas oscuras y planas, aquellas con valores cercanos a 0 en *uint8* en amplias regiones, son de la clase *normal*. Este hecho puede estar asociado a las esquinas generadas por la función de rotación en el aumento de datos (subsección 3.2.2.1) ya que siempre son píxeles de valor 0 y de la clase *normal*, tal y como se ve en la tercera imagen de la figura 4.3. Por otro lado, pese a que el bloque CSDC (subsección 3.2.1.5) emula la atención, no la implementa, amplia el campo receptivo si la convergencia así lo determina y, como se ha visto anteriormente (subsección 3.2.1.5) puede llegar a cubrir el 97% del mapa de características, pero igualmente puede converger a introducir más peso en las *skip-connection* de las SCD (subsección 3.2.1.6) que lo componen. Como se ha visto en la sección de aciertos, 4.2, EFSNet toma regiones de interés con el tamaño medio de un tumor benigno y compacto por lo que en tumores grandes, tomará varias de estas regiones. Se crearán agujeros si una de las regiones cae en una zona oscura y homogénea dentro de un tumor significativamente mayor que el tumor benigno medio, mientras que, se derramará la forma de la máscara si no encuentra un borde, figura 4.4.

Otro error común que disminuye enormemente el valor de las métricas observadas es la mezcla e intercambio de clases en las regiones de interés, figura 4.5. En este error se determina bien la región del tumor pero se realiza una mala predicción del mismo. Por lo tanto, si antes se achacaba el fallo al bloque CSDC,

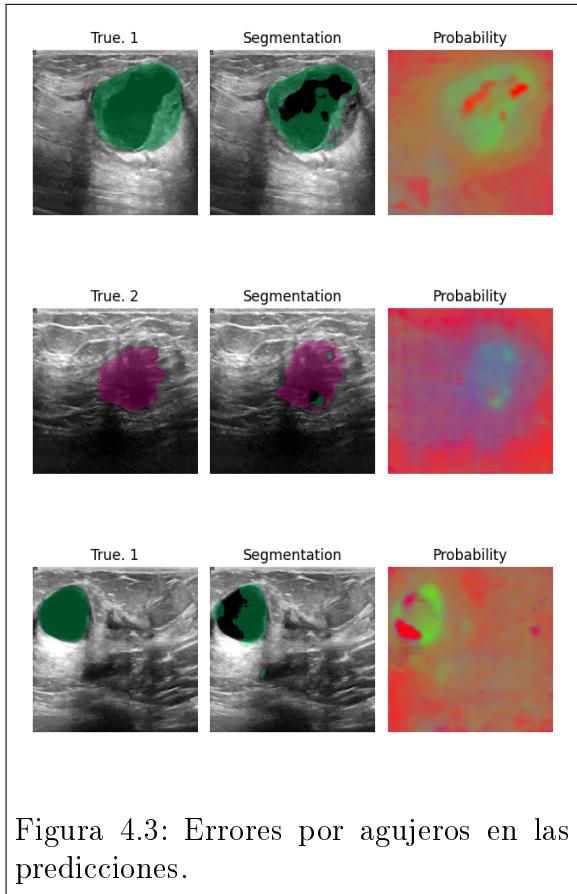


Figura 4.3: Errores por agujeros en las predicciones.

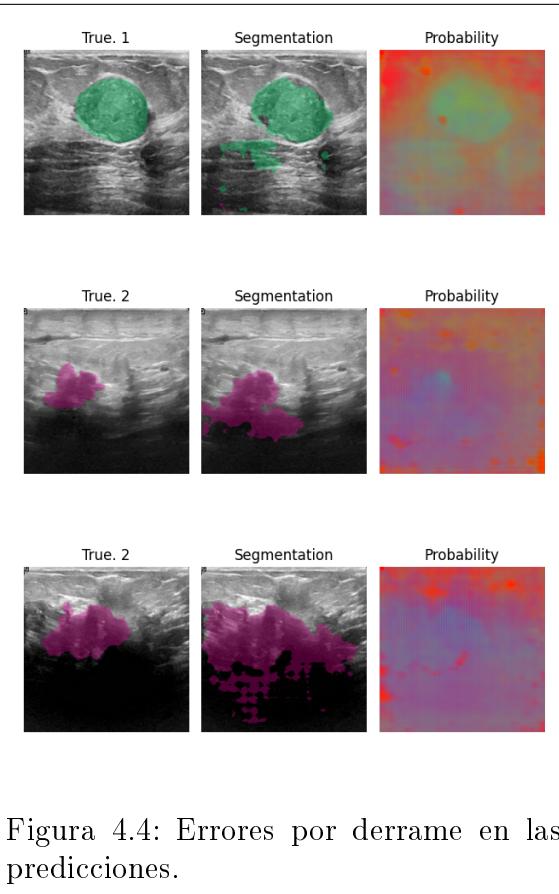
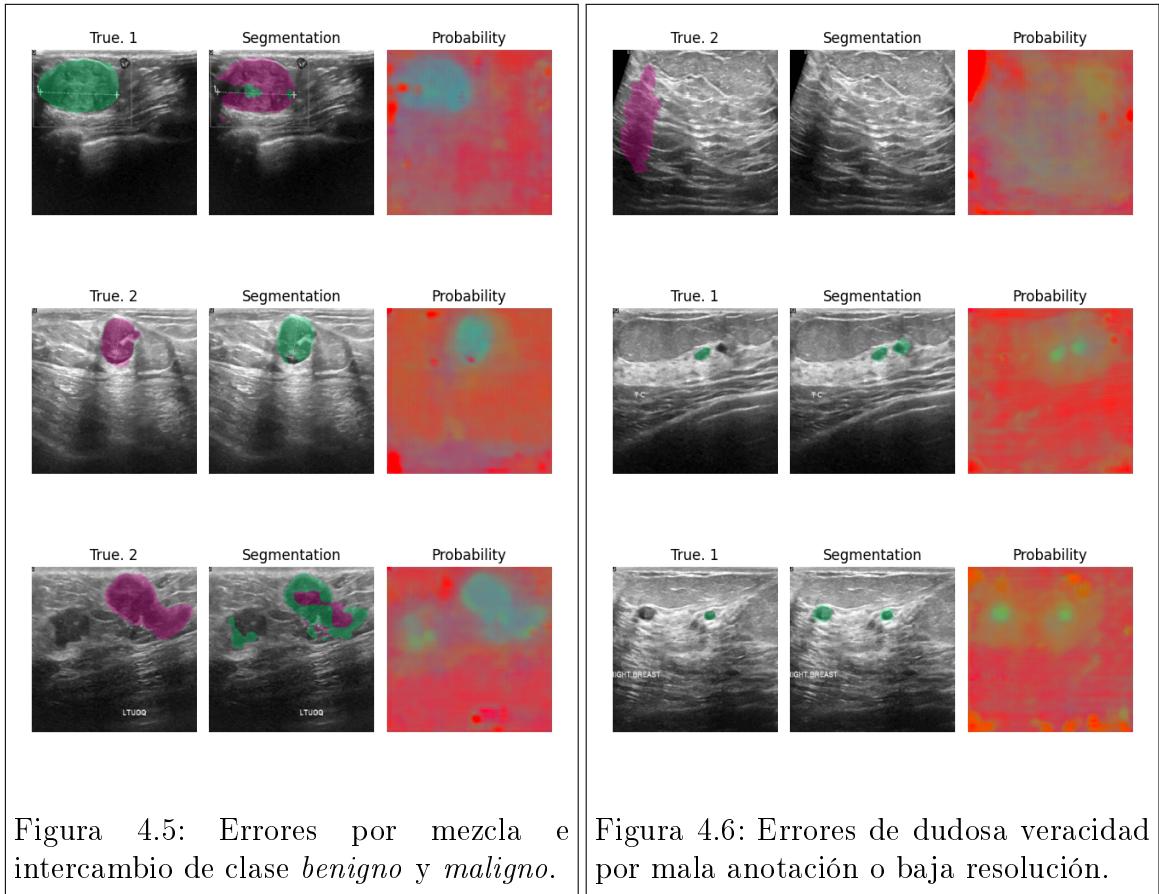


Figura 4.4: Errores por derrame en las predicciones.

ahora se debe adjudicar al *Decoder*. Estas imágenes son las que promueven la diferencia en el valor de las métricas obtenido diferenciando entre *benigno* y *maligno*, y utilizando ambos como *tumor*. En la primera sub-imagen de la figura 4.5 se observa que el campo receptivo efectivo no es grande ni se solapan regiones de interés dado que no hay imagen que contenga al mismo tiempo las tres clases. Se deriva que cada región de interés se evalúa independientemente.

Finalmente, existen errores de dudosa veracidad, figura 4.6. Estos errores se presentan en imágenes en la que no se aprecia una relación clara entre la máscara y los cambios en la textura y en imágenes con dos regiones similares cercanas donde se ha definido como masa tumoral únicamente una de ambas. Dadas las acusaciones vertidas sobre el Dataset BUSI (Pawlowska et al. (2023)) es posible que estos errores no lo sean realmente y añadan error a las métricas vistas. Independientemente de si la realidad es ésta o no, se ha detallado previamente un procedimiento conceptual de la EFSNet que explica correctamente estos errores.



4.4. Tiempo Real

La prueba de inferencia realizada para medir la velocidad de inferencia de EFSNet consiste en la predicción secuencial de 1000 tensores aleatorios de tamaño (1, 1, 256, 256) (siendo la primera dimensión el tamaño de batch) en 10 rondas independientes. En la tabla 4.3 se muestran los resultados.

Tabla 4.3: Tiempo y frecuencia de inferencia.

Procesador	Tiempo por imagen [s]	Imágenes por segundo [fps]
NVidia RTX 4060 (GPU)	0.01234 ± 0.00103	81.621 ± 5.906
Intel Core i7 (CPU)	0.09438 ± 0.00403	10.615 ± 0.458

Además de las métricas presentadas en la tabla 4.3 el artículo de EFSNet (Hu and Wang (2020)) asegura una inferencia de 107 fps utilizando una GPU NVidia Titan Xp sobre los conjuntos de datos CamVid (Brostow et al. (2009)) y CityScapes (Cordts et al. (2016)). Se puede asegurar que EFSNet es un modelo perfectamente válido para tiempo real. Si se utiliza GPU la latencia es despreciable, mientras que si se utiliza CPU se puede tener baja latencia aunque no se “sentirá” instantáneo.

Se debe tener en cuenta que, generalmente, un centro médico suele contar con un clúster de computación o sistemas de computación en la nube con acceso a una o más GPUs y CPUs de mayor potencia que las aquí utilizadas y el rendimiento será aún mayor (aunque también innecesario).

Capítulo 5

Conclusiones y trabajos futuros

A la vista de los resultados se concluye que se han cumplido los objetivos deseados. La implementación de EFSNet para la detección de tumores en ecografías mamarias obtiene métricas comparables a modelos del estado del arte reduciendo el número de parámetros en varios órdenes de magnitud (tablas 4.2a, 4.2b). También se cumple la apuesta por la inferencia en tiempo real, tabla 4.3. Adicionalmente, se han estudiado los errores y aciertos más comunes siendo todos ellos explicables.

La herramienta presentada puede ser de gran utilidad en manos de un experto para guiar la búsqueda o alertar de la presencia de posibles tumores. Se demuestra que esta arquitectura brilla en tumores benignos pequeños que son los correspondientes a los primeros estadíos de la afección. Es principalmente por este motivo que el modelo presenta una solución muy útil en detección temprana de la enfermedad. La inferencia en tiempo real unida a la precisión en la detección temprana dota al modelo de robustez eliminando las aberraciones al permitir observar el mismo punto desde distintos ángulos tantas veces como se requiera.

Se debe mencionar de nuevo en esta sección el mínimo coste computacional requerido, $\simeq 143Mb$ de memoria RAM, que permite no solo realizar inferencia en un ordenador convencional sino en el propio dispositivo de ultrasonidos de forma que se pueda combinar la máscara con la imagen al mismo tiempo que se realiza la revisión médica. Además de la instantaneidad que aporta el modelo, la sencillez algorítmica de su optimización y la reducida cantidad de datos necesaria para su ajuste permite extender esta solución a otros campos de la biomedicina como la detección de hiperintensidades en la masa cerebral, detección de tumores en otros órganos vía ecografía o resonancia magnética, segmentación de células por microscopía o cualquier otra.

En conjunto, esta solución permite a un centro médico utilizar datos de los pacientes en multitud de casos de uso sin violar las restricciones de protección de datos ya que no es necesaria la utilización de computación externa (soluciones

cloud) para el entrenamiento o la inferencia.

Se deben comentar también los inconvenientes de la solución comentada.

En primer lugar, la red puede sufrir en alucinaciones si la imagen de referencia contiene regiones con diferente brillo o contraste. Se aconseja la inferencia continua, en tiempo real, acompañada del criterio de un experto para descartar posibles errores y afianzar aciertos.

En segundo lugar, se ha observado una predilección por tumores de cierto tamaño y se ha comentado la posibilidad de haber “ajustado” el campo receptivo a la media de tamaño. Esta es la principal diferencia entre el bloque CSDC (subsección 3.2.1.5) y un bloque de atención usual, y se debe a la componente residual. Las aberraciones en la máscara que genera este hecho pueden reducir la confianza del experto en las predicciones por lo que es necesaria una explicación detallada de las características del modelo añadiendo carga de trabajo al personal sanitario previa a la utilización del modelo.

Finalmente, los intercambios de clase entre *benigno* y *maligno* pueden ser perjudiciales para el paciente (más si el tumor es maligno y se detecta como benigno) por lo que se recomienda que esta información se contrasta con conocimiento experto y no se deposite toda confianza en la predicción. Es preferible que se utilice el modelo para segmentar *normal*, *tumor*, y, ante la duda apoyarse en la predicción de la clase del tumor ya que de otra forma puede llegar a sesgar la intuición del profesional al cargo.

5.1. Trabajo futuro

A lo largo de la memoria se han comentado varios puntos en los que se podría implementar una mejora. Esta sección sirve como un resumen de todas estas ideas que han podido perderse durante la lectura.

1. Aprendizaje semi-supervisado utilizando *GAN*: utilización de un modelo adversario para realizar aprendizaje semi-supervisado utilizando EFSNet como modelo generador. Esta técnica permite obtener un modelo con menor incertidumbre en el mapa de probabilidades de la segmentación deviniendo en unas predicciones más detalladas y sólidas. En las últimas semanas de desarrollo del proyecto se ha indagado en este campo. En el apéndice A se puede encontrar un conato de ampliación del trabajo siguiendo el artículo *Adversarial Learning with Semi-Supervised Semantic Segmentation* (Hung et al. (2018)).
2. Generalización de tareas y despliegue de modelos: ajustar esta arquitectura a diferentes tareas de segmentación del campo de la biomedicina de forma

independiente. Aprovechar el bajo coste computacional y la alta velocidad de esta arquitectura para desplegar modelos en dispositivos móviles o aparatos médicos de forma que este trabajo teórico vea una o varias aplicaciones prácticas que combinen calidad y accesibilidad.

3. Combinación de conjuntos de datos: una potencial mejora es la combinación de los conjuntos de datos BUSI, UDIAT, BrEaST y OASBUD que son de carácter público para engrosar el número de muestras. Esta vía no es trivial, como se ha comentado cada conjunto de imágenes se ha tomado con un dispositivo diferente y puede resultar complejo homogeneizar el resultado final.
4. Variación del aumento de datos: se ha comentado en la subsección de Aumento de Datos, 3.2.2.1, que la rotación genera zonas negras que son directamente asociadas a la clase normal. Esta característica se propaga a las zonas centrales de los tumores sin textura ni iluminación provocando “agujeros” en la segmentación, figura 4.3. Se debe probar si, al cambiar estas zonas negras por ruido blanco u otra distribución se pueden obtener mejores resultados. Adicionalmente, se pueden variar los parámetros de aumentación de imágenes, el orden en que se aplican o los tipos que se han escogido.
5. Decaimiento de pesos y *Dice Loss*: para aliviar el problema del desbalance de clases se puede añadir el decaimiento de pesos que sugiere el artículo de EFSNet (Hu and Wang (2020)), $w_d = 4 \cdot 10^{-4}$. Además se propone la utilización de una pérdida tal que $l = 0,5 \cdot dice + 0,5 \cdot cce$ que se utiliza en multitud de artículos relacionados con el conjunto de datos BUSI.
6. Segmentación multitarea: se ha comentado en la subsección de predicción, 3.2.1.9, un método para combinar una segmentación binaria junto con un clasificador de clase. El abordaje naïve que fue implementado deviene en errores pero artículos relacionados con este tipo de solución (MTL-Net y variantes, Xu et al. (2023)) implementan técnicas de regularización en la función de pérdida sencillas y efectivas que pueden aplicarse a este modelo.

La implementación de los puntos mencionados anteriormente a modo de análisis univariante sobre la solución expuesta en la memoria se presentan como vías de desarrollo potencialmente beneficiosas en la detección de tumores en ecografías mamarias y otras tareas de segmentación semántica en el campo de la biomedicina.

Bibliografía

Rebecca L. Siegel, Kimberly D. Miller, and Ahmedin Jemal. Cancer statistics, 2019. *CA: A Cancer Journal for Clinicians*, 69(1):7–34, 2019. doi: <https://doi.org/10.3322/caac.21551>. URL <https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.3322/caac.21551>. 1.1

Red Española de Registros del Cáncer. Estimaciones de la incidencia del cáncer en España 2019. *Redecan*, 19:1–14, 2019. 1.1

David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*, 2013. 1.1

Baohao Liao, Yan Meng, and Christof Monz. Parameter-efficient fine-tuning without introducing new latency. *arXiv preprint arXiv:2305.16742*, 2023. 1.1

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 1.1

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. Bitnet: Scaling 1-bit transformers for large language models. *arXiv preprint arXiv:2310.11453*, 2023. 1.1

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024. 1.1

Dong Chen, Ning Liu, Yichen Zhu, Zhengping Che, Rui Ma, Fachao Zhang, Xiaofeng Mou, Yi Chang, and Jian Tang. Epsd: Early pruning with self-distillation for efficient model compression. *arXiv preprint arXiv:2402.00084*, 2024. 1.1

ZhaoJing Zhou, Yun Zhou, Zhuqing Jiang, Aidong Men, and Haiying Wang. An efficient method for model pruning using knowledge distillation with few

- samples. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2515–2519, 2022a. doi: 10.1109/ICASSP43922.2022.9746024. 1.1
- Jinhyuk Park and Albert No. Prune your model before distill it. In *European Conference on Computer Vision*, pages 120–136. Springer, 2022. 1.1
- Peter Seele and Mario D Schultz. From greenwashing to machinewashing: a model and future directions derived from reasoning by analogy. *Journal of Business Ethics*, 178(4):1063–1089, 2022. 1.1
- Xuegang Hu and Haibo Wang. Efficient fast semantic segmentation using continuous shuffle dilated convolutions, 2020. 1.1, 1.2, 2.1.1.3, 2.2.2, 2.2.3, 3.2.1, 3.2.1, 3.2.1.2, 3.2.1.4, 3.2.1.5, 3.2.1.6, 3.2.1.7, 3.2.1.8, 3.2.2.3, 4.4, 5, A.1, B.2
- Walid Al-Dhabyani, Mohammed Gomaa, Hussien Khaled, and Aly Fahmy. Dataset of breast ultrasound images. *Data in brief*, 28:104863, 2020. 1.2, 3.1.3.1, 4
- Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 1.2
- Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017. 1.2, 2.1.1.3, 3.2.1.2
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 1.2
- Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 1.2, 3.2.1.6
- Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 1.2, 3.2.1, 3.2.1.1, B.1
- Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *Proceedings of the european conference on computer vision (ECCV)*, pages 552–568, 2018. 1.2, 2.2.2

Xuetao Zhang, Zhenxue Chen, Q. M. Jonathan Wu, Lei Cai, Dan Lu, and Xianming Li. Fast semantic segmentation for scene perception, Feb 2019. ISSN 1941-0050. 1.2, 2.2.2, 3.2.1, 3.2.1.3, 3.2.1.4

Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation, 2016a. 1.2, 3.2.2.2

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1.2

Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989. 2.1

Yann LeCun, Lawrence D Jackel, Léon Bottou, Corinna Cortes, John S Denker, Harris Drucker, Isabelle Guyon, Urs A Muller, Eduard Sackinger, Patrice Simard, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, 261(276):2, 1995. 2.1, 2.2.2

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 2.1

Olaf Ronneberger, Philipp Fischer, and Thomas 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, proceedings, part III* 18, pages 234–241. Springer, 2015. 2.1, 2.1.1.3, 2.2.1, 2.2.2, 3.2.1, 4.1, 4.2a, 4.2b

Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 06(02):107–116, 1998. doi: 10.1142/S0218488598000094. URL <https://doi.org/10.1142/S0218488598000094>. 2.1, 2.1.1.3

Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Matthias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018. 2.1, 2.2.2, 4.2a

Weimin Tan, Siyuan Chen, and Bo Yan. Diffss: Diffusion model for few-shot semantic segmentation. *arXiv preprint arXiv:2307.00773*, 2023. 2.1, 2.1.2.1

Md Mostafa Kamal Sarker, Hatem A Rashwan, Farhan Akram, Vivek Kumar Singh, Syeda Furruka Banu, Forhad UH Chowdhury, Kabir Ahmed Choudhury, Sylvie Chambon, Petia Radeva, Domenec Puig, et al. Slsnet: Skin lesion segmentation using a lightweight generative adversarial network. *Expert Systems with Applications*, 183:115433, 2021. 2.1, 2.1.2.2

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2.1.1.1

Hanchao Li, Pengfei Xiong, Jie An, and Lingxue Wang. Pyramid attention network for semantic segmentation. *arXiv preprint arXiv:1805.10180*, 2018. 2.1.1.1

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2.1.1.1, 2.2.2

Enze Xie, Wenhui Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021. 2.1.1.1

Ali Hatamizadeh, Yucheng Tang, Vishwesh Nath, Dong Yang, Andriy Myronenko, Bennett Landman, Holger R Roth, and Daguang Xu. Unetr: Transformers for 3d medical image segmentation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 574–584, 2022. 2.1.1.1, 2.2.2

Ali Hatamizadeh, Greg Heinrich, Hongxu Yin, Andrew Tao, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. Fastervit: Fast vision transformers with hierarchical attention. *arXiv preprint arXiv:2306.06189*, 2023. 2.1.1.1, 2.1.1.3

Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In Leonid Karlinsky, Tomer Michaeli, and Ko Nishino, editors, *Computer Vision – ECCV 2022 Workshops*, pages 205–218, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-25066-8. 2.1.1.1, 2.1.1.4, 2.2.2

Jie Zhang, Zhichao Zhang, Hua Liu, and Shiqiang Xu. Satrtransformer: Semantic-aware transformer for breast cancer classification and segmentation. *IET Image Processing*, 17(13):3789–3800, 2023. doi: <https://doi.org/10.1049/ipr2.12897>. URL <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/ipr2.12897>. 2.1.1.1, 4.2b

Weiyi Wu, Chongyang Gao, Xinwen Xu, Siting Li, and Jiang Gui. Memory-efficient sparse pyramid attention networks for whole slide image analysis. *arXiv preprint arXiv:2406.09333*, 2024. 2.1.1.1

Bingyan Liu, Chengyu Wang, Tingfeng Cao, Kui Jia, and Jun Huang. Towards understanding cross and self-attention in stable diffusion for text-guided image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7817–7826, 2024a. 2.2, 2.2.2

Dillon Reis, Jordan Kupec, Jacqueline Hong, and Ahmad Daoudi. Real-time flying object detection with yolov8. *arXiv preprint arXiv:2305.09972*, 2023. 2.1.1.2

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2.1.1.2

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023a. 2.1.1.2

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 2.1.1.2, 2.1.2.5

Zihao Li, Zhuoran Yang, and Mengdi Wang. Reinforcement learning with human feedback: Learning dynamic choices via pessimism. *arXiv preprint arXiv:2305.18438*, 2023b. 2.1.1.2

Hoo-Chang Shin, Holger R. Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel J. Mollura, and Ronald M. Summers. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *CoRR*, abs/1602.03409, 2016. URL <http://arxiv.org/abs/1602.03409>. 2.1.1.2

- Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020. 2.1.1.2, 2.1.2.5
- Shih-Cheng Huang, Anuj Pareek, Saeed Seyyedi, Imon Banerjee, and Matthew P Lungren. Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines. *NPJ digital medicine*, 3(1):136, 2020a. 2.4a, 2.1.1.2
- Yasha Ektefaie, George Dasoulas, Ayush Noori, Maha Farhat, and Marinka Zitnik. Multimodal learning with graphs. *Nature Machine Intelligence*, 5(4):340–350, 2023. 2.4b, 2.1.1.2
- Rui-Jie Zhu, Yu Zhang, Ethan Siferman, Tyler Sheaves, Yiqiao Wang, Dustin Richmond, Peng Zhou, and Jason K Eshraghian. Scalable matmul-free language modeling. *arXiv preprint arXiv:2406.02528*, 2024. 2.1.1.3
- Junyan Li, Delin Chen, Tianle Cai, Peihao Chen, Yining Hong, Zhenfang Chen, Yikang Shen, and Chuang Gan. Flexattention for efficient high-resolution vision-language models. *arXiv preprint arXiv:2407.20228*, 2024a. 2.1.1.3
- Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3531–3539, 2021. 2.1.1.3
- Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 122–138, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01264-9. 2.1.1.3, 3.2.1.6
- Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 2.1.1.3, 2.2.3
- Sanghun Lee and Chulhee Lee. Revisiting spatial dropout for regularizing convolutional neural networks. *Multimedia Tools and Applications*, 79(45):34195–34207, 2020. 2.1.1.3, 3.2.1
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016. doi: 10.1109/CVPR.2016.90. 2.1.1.3, 3.2.1, 3.2.1.3, 3.2.1.8, 4.2b

- Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016b. 2.1.1.3, 3.2.1.2
- Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, June 2018. doi: 10.1109/CVPR.2018.00716. 2.1.1.3, 2.2.3
- Tianlin Liu, Mathieu Blondel, Carlos Riquelme, and Joan Puigcerver. Routers in vision mixture of experts: An empirical study. *arXiv preprint arXiv:2401.15969*, 2024b. 2.1.1.4
- Svetlana Pavlitskaya, Christian Hubschneider, Michael Weber, Ruby Moritz, Fabian Huger, Peter Schlicht, and Marius Zollner. Using mixture of expert models to gain insights into semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 342–343, 2020. 2.1.1.4
- Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34:8583–8595, 2021. 2.1.1.4
- K Santle Camilus and VK Govindan. A review on graph based segmentation. *International Journal of Image, Graphics and Signal Processing*, 4(5):1, 2012. 2.1.1.4
- Maciej Krzywda, Szymon Łukasik, and Amir H Gandomi. Graph neural networks in computer vision-architectures, datasets and common approaches. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2022. 2.1.1.4
- Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision gnn: An image is worth graph of nodes. *Advances in neural information processing systems*, 35:8291–8303, 2022. 2.1.1.4
- Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2.1.1.4
- Alain Bretto and Luc Gillibert. Hypergraph-based image representation. volume 3434, pages 1–11, 04 2005. ISBN 978-3-540-25270-2. doi: 10.1007/978-3-540-31988-7_1. 2.1.1.4

Song Bai, Feihu Zhang, and Philip HS Torr. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110:107637, 2021. 2.1.1.4

Srinivas C Turaga, Joseph F Murray, Viren Jain, Fabian Roth, Moritz Helmstaedter, Kevin Briggman, Winfried Denk, and H Sebastian Seung. Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural computation*, 22(2):511–538, 2010. 2.1.1.4

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020. ISSN 2666-6510. doi: <https://doi.org/10.1016/j.aiopen.2021.01.001>. URL <https://www.sciencedirect.com/science/article/pii/S2666651021000012>. 2.1.1.4

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 2.1.1.4

Xiao Liu, Chenxu Zhang, and Lei Zhang. Vision mamba: A comprehensive survey and taxonomy. *arXiv preprint arXiv:2405.04404*, 2024c. 2.1.1.4

Weibin Liao, Yinghao Zhu, Xinyuan Wang, Cehngwei Pan, Yasha Wang, and Liantao Ma. Lightm-unet: Mamba assists in lightweight unet for medical image segmentation. *arXiv preprint arXiv:2403.05246*, 2024. 2.1.1.4

Jiarun Liu, Hao Yang, Hong-Yu Zhou, Yan Xi, Lequan Yu, Yizhou Yu, Yong Liang, Guangming Shi, Shaoting Zhang, Hairong Zheng, et al. Swin-umamba: Mamba-based unet with imagenet-based pretraining. *arXiv preprint arXiv:2402.03302*, 2024d. 2.1.1.4

Zi Ye and Tianxiang Chen. P-mamba: Marrying perona malik diffusion with mamba for efficient pediatric echocardiographic left ventricular segmentation. *arXiv preprint arXiv:2402.08506*, 2024. 2.1.1.4

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2.1.2.1

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn.openai.com/papers/dall-e-3.pdf*, 2(3):8, 2023. 2.1.2.1

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al.

- Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. 2.1.2.1
- Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 19830–19843, 2023a. 2.5
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2.1.2.2, 2.2.2
- Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. *arXiv preprint arXiv:1802.07934*, 2018. 2.1.2.2, 1, A, A.1, A.2, A.2, A.2
- Brian Dolhansky and Cristian Canton Ferrer. Eye in-painting with exemplar generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7902–7911, 2018. 2.1.2.2
- Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018. 2.1.2.2
- Zhaqing Pan, Weijie Yu, Xiaokai Yi, Asifullah Khan, Feng Yuan, and Yuhui Zheng. Recent progress on generative adversarial networks (gans): A survey. *IEEE access*, 7:36322–36333, 2019. 2.1.2.2
- Abdul Jabbar, Xi Li, and Bourahla Omar. A survey on generative adversarial networks: Variants, applications, and training. *ACM Computing Surveys (CSUR)*, 54(8):1–49, 2021. 2.1.2.2
- Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, and Jieping Ye. A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE transactions on knowledge and data engineering*, 35(4):3313–3332, 2021. 2.1.2.2
- Aaron Defazio, Ashok Cutkosky, Harsh Mehta, and Konstantin Mishchenko. When, why and how much? adaptive learning rate scheduling by refinement. *arXiv preprint arXiv:2310.07831*, 2023. 2.1.2.3
- Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991. 2.1.2.3
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2.1.2.3

Purnendu Mishra and Kishor Sarawadekar. Polynomial learning rate policy with warm restart for deep neural network. In *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, pages 2087–2092, 2019. doi: 10.1109/TENCON.2019.8929465. 2.1.2.3

Leslie N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472, 2017. doi: 10.1109/WACV.2017.58. 2.1.2.3

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 2.1.2.3

Aditya Sharad Golatkar, Alessandro Achille, and Stefano Soatto. Time matters in regularizing deep networks: Weight decay and data augmentation affect early learning dynamics, matter little near convergence. *Advances in Neural Information Processing Systems*, 32, 2019. 2.1.2.3

Aitor Lewkowycz and Guy Gur-Ari. On the training dynamics of deep networks with l_2 regularization. *Advances in Neural Information Processing Systems*, 33:4790–4799, 2020. 2.1.2.3, 3.2.2.3

Subrato Bharati, M Mondal, Prajjoy Podder, and VB Prasath. Federated learning: Applications, challenges and future directions. *International Journal of Hybrid Intelligent Systems*, 18(1-2):19–35, 2022. 2.1.2.4

Miguel Buenestado Cortés. Aprendizaje federado aplicado al diagnóstico de tumores mamarios en imágenes de ultrasonido. *UNED. Trabajo Fin de Master.*, February 2022. 2.1.2.4

Mohammad Moshawrab, Mehdi Adda, Abdenour Bouzouane, Hussein Ibrahim, and Ali Raad. Reviewing federated learning aggregation algorithms; strategies, contributions, limitations and future perspectives. *Electronics*, 12(10), 2023. ISSN 2079-9292. doi: 10.3390/electronics12102287. URL <https://www.mdpi.com/2079-9292/12/10/2287>. 3

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 2.1.2.4

Peihua Mai, Ran Yan, and Yan Pang. Rflpa: A robust federated learning framework against poisoning attacks with secure aggregation. *arXiv preprint arXiv:2405.15182*, 2024. 2.1.2.4

G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, July 2006. ISSN 1095-9203. doi: 10.1126/science.1127647. 2.1.2.5

Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. *Nature Communications*, 15(1):654, 2024a. 2.1.2.5

Getao Du, Xu Cao, Jimin Liang, Xueli Chen, and Yonghua Zhan. Medical image segmentation based on u-net: A review. *Journal of Imaging Science & Technology*, 64(2), 2020. 2.2.1

Nahian Siddique, Sidike Paheding, Colin P. Elkin, and Vijay Devabhaktuni. U-net and its variants for medical image segmentation: A review of theory and applications. *IEEE Access*, 9:82031–82057, 2021. doi: 10.1109/ACCESS.2021.3086020. 2.2.1

Joanna M. Wardlaw, Maria C. Valdés Hernández, and Susana Muñoz-Maniega. What are white matter hyperintensities made of?: Relevance to vascular cognitive impairment. *Journal of the American Heart Association*, 4(6), June 2015. ISSN 2047-9980. doi: 10.1161/jaha.114.001140. 2.2.1

Mirthe Coenen, Geert Jan Biessels, Charles DeCarli, Evan F. Fletcher, Pauline M. Maillard, Frederik Barkhof, Josephine Barnes, Thomas Benke, Jooske M.F. Boomsma, Christopher P.L.H. Chen, Peter Dal-Bianco, Anna Dewenter, Marco Duering, Christian Enzinger, Michael Ewers, Lieza G. Exalto, Nicolai Franzmeier, Onno Groeneweld, Saima Hilal, Edith Hofer, Huiberdina L. Koek, Andrea B. Maier, Cheryl R. McCreary, Janne M. Papma, Ross W. Paterson, Yolande A.L. Pijnenburg, Anna Rubinski, Reinhold Schmidt, Jonathan M. Schott, Catherine F. Slattery, Eric E. Smith, Carole H. Sudre, Rebecca M.E. Steketee, Esther van den Berg, Wiesje M. van der Flier, Narayanaswamy Venkatasubramanian, Meike W. Vernooij, Frank J. Wolters, Xu Xin, J. Matthijs Biesbroek, and Hugo J. Kuijf. Spatial distributions of white matter hyperintensities on brain mri: A pooled analysis of individual participant data from 11 memory clinic cohorts. *NeuroImage: Clinical*, 40:103547, 2023. ISSN 2213-1582. doi: <https://doi.org/10.1016/j.nicl.2023.103547>. URL <https://www.sciencedirect.com/science/article/pii/S2213158223002383>. 2.2.1

S Medrano Martorell, M Cuadrado Blázquez, D García Figueredo, S González Ortiz, and J Capellades Font. Imágenes puntiformes hiperintensas en la sustancia blanca: una aproximación diagnóstica. *Radiología*, 54(4):321–335, 2012. 2.2.1

Sirvan Khalighi, Kartik Reddy, Abhishek Midya, Krunal Balvantbhai Pandav, Anant Madabhushi, and Malak Abedalthagafi. Artificial intelligence in neuro-oncology: advances and challenges in brain tumor diagnosis, prognosis, and precision treatment. *NPJ Precision Oncology*, 8(1):80, 2024. 2.2.1

Sana Intidhar Labidi-Galy, Isabelle Treilleux, Sophie Goddard-Leon, Jean-Damien Combes, Jean-Yves Blay, Isabelle Ray-Coquard, Christophe Caux, and Nathalie Bendriss-Vermare. Plasmacytoid dendritic cells infiltrating ovarian cancer are associated with poor prognosis. *Oncoimmunology*, 1(3):380–382, 2012. 2.2.1

Ashwini Kodipalli, Steven L. Fernandes, Vaishnavi Gururaj, Shriya Varada Rameshbabu, and Santosh Dasar. Performance analysis of segmentation and classification of ct-scanned ovarian tumours using u-net and deep convolutional neural networks. *Diagnostics*, 13(13), 2023. ISSN 2075-4418. doi: 10.3390/diagnostics13132282. URL <https://www.mdpi.com/2075-4418/13/13/2282>. 2.2.1

Lijiang Chen, Changkun Qiao, Meijing Wu, Linghan Cai, Cong Yin, Mukun Yang, Xiubo Sang, and Wenpei Bai. Improving the segmentation accuracy of ovarian-tumor ultrasound images using image inpainting. *Bioengineering*, 10(2), 2023b. ISSN 2306-5354. doi: 10.3390/bioengineering10020184. URL <https://www.mdpi.com/2306-5354/10/2/184>. 2.2.1

Valery L Feigin, Mayowa O Owolabi, Foad Abd-Allah, Rufus O Akinyemi, Natalia V Bhattacharjee, Michael Brainin, Jackie Cao, Valeria Caso, Bronte Dalton, Alan Davis, et al. Pragmatic solutions to reduce the global burden of stroke: a world stroke organization–lancet neurology commission. *The Lancet Neurology*, 22(12):1160–1206, 2023. 2.2.1

Valery L Feigin, Michael Brainin, Bo Norrvig, Sheila Martins, Ralph L Sacco, Werner Hacke, Marc Fisher, Jeyaraj Pandian, and Patrice Lindsay. World stroke organization (wso): global stroke fact sheet 2022. *International Journal of Stroke*, 17(1):18–29, 2022. 2.2.1

Mishaim Malik, Benjamin Chong, Justin Fernandez, Vickie Shim, Nikola Kirilov Kasabov, and Alan Wang. Stroke lesion segmentation and deep learning: A comprehensive review. *Bioengineering*, 11(1), 2024. ISSN 2306-5354. doi: 10.3390/bioengineering11010086. URL <https://www.mdpi.com/2306-5354/11/1/86>. 2.2.1

Jialin Luo, Peishan Dai, Zhuang He, Zhongchao Huang, Shenghui Liao, and Kun Liu. Deep learning models for ischemic stroke lesion segmentation

in medical images: A survey. *Computers in Biology and Medicine*, 175: 108509, 2024. ISSN 0010-4825. doi: <https://doi.org/10.1016/j.compbiomed.2024.108509>. URL <https://www.sciencedirect.com/science/article/pii/S0010482524005936>. 2.2.1

Adrian Galdran, André Anjos, José Dolz, Hadi Chakor, Hervé Lombaert, and Ismail Ben Ayed. State-of-the-art retinal vessel segmentation with minimalistic models. *Scientific Reports*, 12(1):6174, 2022. 2.2.1

Hesham Abdushkour, Toufique A Soomro, Ahmed Ali, Fayyaz Ali Jandan, Herbert Jelinek, Farida Memon, Faisal Althobiani, Saleh Mohammed Ghonaim, and Muhammad Irfan. Enhancing fine retinal vessel segmentation: Morphological reconstruction and double thresholds filtering strategy. *PloS One*, 18(7): e0288792, 2023. 2.2.1

Jair Cervantes, Jared Cervantes, Farid García-Lamont, Arturo Yee-Rendon, Josué Espejel Cabrera, and Laura Domínguez Jalili. A comprehensive survey on segmentation techniques for retinal vessel segmentation. *Neurocomputing*, 556:126626, 2023. ISSN 0925-2312. doi: <https://doi.org/10.1016/j.neucom.2023.126626>. URL <https://www.sciencedirect.com/science/article/pii/S092523122300749X>. 2.2.1

Sergey P Primakov, Abdalla Ibrahim, Janita E van Timmeren, Guangyao Wu, Simon A Keek, Manon Beuque, Renée WY Granzier, Elizaveta Lavrova, Madeleine Scrivener, Sebastian Sanduleanu, et al. Automated detection and segmentation of non-small cell lung cancer computed tomography images. *Nature communications*, 13(1):3423, 2022. 2.2.1

John-Melle Bokhorst, Iris D Nagtegaal, Filippo Fraggetta, Simona Vatrano, Wilma Mesker, Michael Vieth, Jeroen van der Laak, and Francesco Ciompi. Deep learning for multi-class semantic segmentation enables colorectal cancer detection and classification in digital pathology images. *Scientific Reports*, 13(1):8398, 2023. 2.2.1

Md Roman Bhuiyan and Junaidi Abdullah. Detection on cell cancer using the deep transfer learning and histogram based image focus quality assessment. *Sensors*, 22(18), 2022. ISSN 1424-8220. doi: 10.3390/s22187007. URL <https://www.mdpi.com/1424-8220/22/18/7007>. 2.2.1

Michał Karol, Martin Tabakov, Urszula Markowska-Kaczmar, and Lukasz Fulawka. Deep learning for cancer cell detection: do we need dedicated models? *Artificial Intelligence Review*, 57(3):53, 2024. 2.2.1

World Health Organization. *Global tuberculosis report 2020*. World Health Organization, 2020. 2.2.1

Sivaramakrishnan Rajaraman, Les R. Folio, Jane Dimperio, Philip O. Alderson, and Sameer K. Antani. Improved semantic segmentation of tuberculosis—consistent findings in chest x-rays using augmented training of modality-specific u-net models with weak localizations. *Diagnostics*, 11(4), 2021. ISSN 2075-4418. doi: 10.3390/diagnostics11040616. URL <https://www.mdpi.com/2075-4418/11/4/616>. 2.2.1

Tae Hoon Kim, Moez Krichen, Stephen Ojo, Meznah A. Alamro, and Gabriel Avelino Sampedro. Tssg-cnn: A tuberculosis semantic segmentation-guided model for detecting and diagnosis using the adaptive convolutional neural network. *Diagnostics*, 14(11), 2024. ISSN 2075-4418. doi: 10.3390/diagnostics14111174. URL <https://www.mdpi.com/2075-4418/14/11/1174>. 2.2.1

Reza Azad, Ehsan Khodapanah Aghdam, Amelie Rauland, Yiwei Jia, Atlas Haddadi Avval, Afshin Bozorgpour, Sanaz Karimijafarbigloo, Joseph Paul Cohen, Ehsan Adeli, and Dorit Merhof. Medical image segmentation review: The success of u-net. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2.2.1, 2.2.2

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2.2.2, 4.1, 4.2a, 4.2b, A.1

Yutong Xie, Bing Yang, Qingbiao Guan, Jianpeng Zhang, Qi Wu, and Yong Xia. Attention mechanisms in medical image segmentation: A survey. *arXiv preprint arXiv:2305.17937*, 2023. 2.2.2

Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021. 2.2.2, 4.2a

Jun Ma, Feifei Li, and Bo Wang. U-mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*, 2024b. 2.2.2

Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE transactions on medical imaging*, 39(6):1856–1867, 2019. 2.2.2

Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen-Wei Chen, and Jian Wu. Unet 3+: A full-scale connected unet for medical image segmentation. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 1055–1059. IEEE, 2020b. 2.2.2

Haoming Li, Jinghui Fang, Shengfeng Liu, Xiaowen Liang, Xin Yang, Zixin Mai, Manh The Van, Tianfu Wang, Zhiyi Chen, and Dong Ni. Cr-unet: A composite network for ovary and follicle segmentation in ultrasound images. *IEEE journal of biomedical and health informatics*, 24(4):974–983, 2019. 2.2.2

Reza Azad, Maryam Asadi-Aghbolaghi, Mahmood Fathy, and Sergio Escalera. Bi-directional convlstm u-net with densely connected convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019. 2.2.2

Changlu Guo, Márton Szemenyei, Yugen Yi, Wenle Wang, Buer Chen, and Changqi Fan. Sa-unet: Spatial attention u-net for retinal vessel segmentation. In *2020 25th international conference on pattern recognition (ICPR)*, pages 1236–1242. IEEE, 2021. 2.2.2

Jinjin Hai, Kai Qiao, Jian Chen, Hongna Tan, Jingbo Xu, Lei Zeng, Dapeng Shi, and Bin Yan. Fully convolutional densenet with multiscale context for automated breast tumor segmentation. *Journal of healthcare engineering*, 2019(1):8415485, 2019. 2.2.2

Malvika Ashok and Abhishek Gupta. Comparative study of trans - gan architecture for bio-medical image semantic segmentation. In *2022 7th International Conference on Communication and Electronics Systems (ICCES)*, pages 1564–1570, 2022. doi: 10.1109/ICCES54183.2022.9835992. 2.2.2

Ahmed Iqbal, Muhammad Sharif, Mussarat Yasmin, Mudassar Raza, and Shabib Aftab. Generative adversarial networks and its applications in the biomedical image segmentation: a comprehensive survey. *International Journal of Multimedia Information Retrieval*, 11(3):333–368, 2022. ISSN 2192-662X. doi: 10.1007/s13735-022-00240-x. URL <https://doi.org/10.1007/s13735-022-00240-x>. 2.2.2

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. URL <http://arxiv.org/abs/1411.1784>. 2.2.2

Tianyu Lin, Zhiguang Chen, Zhonghao Yan, Weijiang Yu, and Fudan Zheng. Stable diffusion segmentation for biomedical images with single-step reverse process, 2024. URL <https://arxiv.org/abs/2406.18361>. 2.2.2

Junde Wu, Rao Fu, Huihui Fang, Yu Zhang, Yehui Yang, Haoyi Xiong, Huiying Liu, and Yanwu Xu. Medsegdiff: Medical image segmentation with diffusion probabilistic model, 2023. URL <https://arxiv.org/abs/2211.00611>. 2.2.2

Sajjad Abbasi, Mohsen Hajabdollahi, Nader Karimi, and Shadrokh Samavi. Modeling teacher-student techniques in deep neural networks for knowledge distillation. *CoRR*, abs/1912.13179, 2019. URL <http://arxiv.org/abs/1912.13179>. 2.2.2

Liyan Sun, Jianxiong Wu, Xinghao Ding, Yue Huang, Guisheng Wang, and Yizhou Yu. A teacher-student framework for semi-supervised medical image segmentation from mixed supervision. *CoRR*, abs/2010.12219, 2020. URL <https://arxiv.org/abs/2010.12219>. 2.2.2

Boliang Li, Yan Wang, Yaming Xu, and Chen Wu. Dsst: A dual student model guided student–teacher framework for semi-supervised medical image segmentation. *Biomedical Signal Processing and Control*, 90: 105890, 2024b. ISSN 1746-8094. doi: <https://doi.org/10.1016/j.bspc.2023.105890>. URL <https://www.sciencedirect.com/science/article/pii/S174680942301323X>. 2.2.2

Hongjiang Guo, Shengwen Wang, Hao Dang, Kangle Xiao, Yaru Yang, Wenpei Liu, Tongtong Liu, and Yiying Wan. Lightbtseg: A lightweight breast tumor segmentation model using ultrasound images via dual-path joint knowledge distillation. In *2023 China Automation Congress (CAC)*, pages 3841–3847. IEEE, 2023. 2.2.2, 2.2.3

Ange Lou, Shuyue Guan, and Murray Loew. Cfpnet-m: A light-weight encoder-decoder based network for multimodal biomedical image real-time segmentation. *Computers in Biology and Medicine*, 154:106579, 2023. ISSN 0010-4825. doi: <https://doi.org/10.1016/j.compbiomed.2023.106579>. URL <https://www.sciencedirect.com/science/article/pii/S0010482523000446>. 2.2.2

Ademola E. Ilesanmi, Utairat Chaumrattanakul, and Stanislav S. Makhanov. Methods for the segmentation and classification of breast ultrasound images: a review. *Journal of Ultrasound*, 24(4):367–382, 2021. ISSN 1876-7931. doi: 10.1007/s40477-020-00557-5. URL <https://doi.org/10.1007/s40477-020-00557-5>. 2.2.3

Gongping Chen, Lei Li, Yu Dai, Jianxun Zhang, and Moi Hoon Yap. Aau-net: An adaptive attention u-net for breast lesions segmentation in ultrasound images. *IEEE Transactions on Medical Imaging*, 42(5):1289–1300, 2023c. doi: 10.1109/TMI.2022.3226268. 2.2.3

P. O. Vianna, R. Farias, and W. C. A. Pereira. Performance of the segnet in the segmentation of breast ultrasound lesions. In *2021 Global Medical Engineering Physics Exchanges/Pan American Health Care Exchanges (GMEPE/PAHCE)*, pages 1–4, 2021. doi: 10.1109/GMEPE/PAHCE50215.2021.9434877. 2.2.3

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2014. URL <https://arxiv.org/abs/1312.6034>. 2.2.3

Aya Abdelsalam Ismail, Héctor Corrada Bravo, and Soheil Feizi. Improving deep learning interpretability by saliency guided training. *CoRR*, abs/2111.14338, 2021. URL <https://arxiv.org/abs/2111.14338>. 2.2.3

Zhenyuan Ning, Shengzhou Zhong, Qianjin Feng, Wufan Chen, and Yu Zhang. Smu-net: Saliency-guided morphology-aware u-net for breast lesion segmentation in ultrasound image. *IEEE Transactions on Medical Imaging*, 41(2):476–490, 2022. doi: 10.1109/TMI.2021.3116087. 2.2.3

Aleksandar Vakanski, Min Xian, and Phoebe E. Freer. Attention-enriched deep learning model for breast tumor segmentation in ultrasound images. *Ultrasound in Medicine & Biology*, 46(10):2819–2833, 2020. ISSN 0301-5629. doi: <https://doi.org/10.1016/j.ultrasmedbio.2020.06.015>. URL <https://www.sciencedirect.com/science/article/pii/S0301562920302878>. 2.2.3

Haichao Cao, Shiliang Pu, Wenming Tan, Junyan Tong, and Di Zhang. Multi-tasking u-shaped network for benign and malignant classification of breast masses. *IEEE Access*, 8:223396–223404, 2020. ISSN 2169-3536. doi: 10.1109/access.2020.3042889. 2.2.3

Meng Xu, Kuan Huang, and Xiaojun Qi. A regional-attentive multi-task learning framework for breast ultrasound image segmentation and classification. *IEEE Access*, 11:5377–5392, 2023. doi: 10.1109/ACCESS.2023.3236693. 2.2.3, 4.1, 4.2b, 6

Quan Zhou, Qianwen Wang, Yunchao Bao, Lingjun Kong, Xin Jin, and Weihua Ou. Laednet: A lightweight attention encoder–decoder network for ultrasound medical image segmentation. *Comput. Electr. Eng.*, 99(C), apr 2022b. ISSN

0045-7906. doi: 10.1016/j.compeleceng.2022.107777. URL <https://doi.org/10.1016/j.compeleceng.2022.107777>. 2.2.3

Sheng Yuan, Zhao Qiu, Peipei Li, and Yuqi Hong. Rmau-net: Breast tumor segmentation network based on residual depthwise separable convolution and multiscale channel attention gates. *Applied Sciences*, 13(20), 2023. ISSN 2076-3417. doi: 10.3390/app132011362. URL <https://www.mdpi.com/2076-3417/13/20/11362>. 2.2.3

Shuai Zhang and Yanmin Niu. Lcmunet: A lightweight network combining cnn and mlp for real-time medical image segmentation. *Bioengineering*, 10(6), 2023. ISSN 2306-5354. doi: 10.3390/bioengineering10060712. URL <https://www.mdpi.com/2306-5354/10/6/712>. 2.2.3, 4.2b

Fenglin Cai, Jiaying Wen, Fangzhou He, Yulong Xia, Weijun Xu, Yong Zhang, Li Jiang, and Jie Li. Sc-unext: A lightweight image segmentation model with cellular mechanism for breast ultrasound tumor diagnosis. *Journal of Imaging Informatics in Medicine*, 37(4):1505–1515, 2024. ISSN 2948-2933. doi: 10.1007/s10278-024-01042-9. URL <https://doi.org/10.1007/s10278-024-01042-9>. 2.2.3, 4.2, 4.2a

Anna Pawłowska, Anna Ćwierz Pieńkowska, Agnieszka Domalik, Dominika Jaguś, Piotr Kasprzak, Rafał Matkowski, Łukasz Fura, Andrzej Nowicki, and Norbert Żołek. Curated benchmark dataset for ultrasound based breast lesion analysis. *Scientific Data*, 11(1):148, 2024. ISSN 2052-4463. doi: 10.1038/s41597-024-02984-z. URL <https://doi.org/10.1038/s41597-024-02984-z>. 1

Hanna Piotrzkowska-Wróblewska, Katarzyna Dobruch-Sobczak, Michał Byra, and Andrzej Nowicki. Open access database of raw ultrasonic signals acquired from malignant and benign breast lesions. *Medical Physics*, 44(11):6105–6109, 2017. doi: <https://doi.org/10.1002/mp.12538>. URL <https://aapm.onlinelibrary.wiley.com/doi/abs/10.1002/mp.12538>. 2

Moi Hoon Yap, Gerard Pons, Joan Martí, Sergi Ganau, Melcior Sentís, Reyer Zwiggelaar, Adrian K. Davison, and Robert Martí. Automated breast ultrasound lesions detection using convolutional neural networks. *IEEE Journal of Biomedical and Health Informatics*, 22(4):1218–1226, July 2018. ISSN 2168-2208. doi: 10.1109/jbhi.2017.2731873. 3

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?." explaining the predictions of any classifier. In *Proceedings of the 22nd ACM*

SIGKDD international conference on knowledge discovery and data mining, pages 1135–1144, 2016. 3.1.3.1

Anna Pawłowska, Piotr Karwat, and Norbert Żołek. re:[dataset of breast ultrasound images by w. al-dhabayani, m. gomaa, h. khaled & a. fahmy, data in brief, 2020, 28, 104863]. *Data in Brief*, 48, 2023. 3.1, 3.1.3.2, 4.3

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 3.2.1, 3.2.1.1, 3.2.2.3, B.1

Anish Shah, Eashan Kadam, Hena Shah, Sameer Shinde, and Sandip Shingade. Deep residual networks with exponential linear unit. In *Proceedings of the third international symposium on computer vision and the internet*, pages 59–65, 2016. 3.2.1

Shie Mannor, Dori Peleg, and Reuven Rubinstein. The cross entropy method for classification. pages 561–568, 01 2005. doi: 10.1145/1102351.1102422. 3.2.2.2

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3.2.2.3

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019. 3.2.2.3

Meng Xu, Kuan Huang, Qiuxiao Chen, and Xiaojun Qi. Mssa-net: Multi-scale self-attention network for breast ultrasound image segmentation. In *2021 IEEE 18th international symposium on biomedical imaging (ISBI)*, pages 827–831. IEEE, 2021. 4.1, 4.2

Se Woon Cho, Na Rae Baek, and Kang Ryoung Park. Deep learning-based multi-stage segmentation method using ultrasound images for breast cancer diagnosis. *Journal of King Saud University - Computer and Information Sciences*, 34 (10, Part B):10273–10292, 2022. ISSN 1319-1578. doi: <https://doi.org/10.1016/j.jksuci.2022.10.020>. URL <https://www.sciencedirect.com/science/article/pii/S1319157822003718>. 4.1, 4.2, 4.2a

Satish Bansal. Breast tumor recognition by semantic segmentation of multiclass ultrasound images. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11:938–946, 10 2023. doi: 10.17762/ijritcc.v11i9.8986. 4.1, 4.2

- Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 4.1, 4.2a, 4.2b
- Jeya Maria Jose Valanarasu and Vishal M. Patel. Unext: Mlp-based rapid medical image segmentation network, 2022. URL <https://arxiv.org/abs/2203.04967>. 4.2a
- Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, Dec 2017. ISSN 1939-3539. doi: 10.1109/TPAMI.2016.2644615. 4.2a
- Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 4.2a, 4.2b
- Michał Byra, Piotr Jarosik, Aleksandra Szubert, Michael Galperin, Haydee Ojeda-Fournier, Linda Olson, Mary O’Boyle, Christopher Comstock, and Michael Andre. Breast mass segmentation in ultrasound with selective kernel u-net convolutional neural network. *Biomedical Signal Processing and Control*, 61: 102027, 2020. 4.2b
- Bryar Shareef, Aleksandar Vakanski, Phoebe E. Freer, and Min Xian. Estan: Enhanced small tumor-aware network for breast ultrasound image segmentation. *Healthcare*, 10(11), 2022. ISSN 2227-9032. doi: 10.3390/healthcare10112262. URL <https://www.mdpi.com/2227-9032/10/11/2262>. 4.2b
- Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009. ISSN 0167-8655. doi: <https://doi.org/10.1016/j.patrec.2008.04.005>. URL <https://www.sciencedirect.com/science/article/pii/S0167865508001220>. Video-based Object and Event Analysis. 4.4
- Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 4.4

Apéndice A

Aprendizaje Adversario con Segmentación Semántica Semi-Supervisada

El artículo *Adversarial Learning with Semi-Supervised Semantic Segmentation* (Hung et al. (2018)), AL-S4, propone la utilización de redes generativas adversarias (*GAN*) para combinar funciones de optimización correspondientes al aprendizaje supervisado y al no-supervisado de forma que la red generativa obtenga mapas de segmentación sólidos y sin aberraciones.

A.1. Algoritmo AL-S4

El método AL-S4 utiliza la combinación de una red generadora, $G(\cdot)$, y una red discriminadora, $D(\cdot)$, en un esquema estilo *GAN*.

La red generadora establece una relación entre una imagen de entrada, I , con dimensiones $H \times W \times 1$ a un mapa de probabilidad $H \times W \times C$. Siendo C las diferentes clases a las que se puede asociar un píxel en la máscara asociada, M .

La red discriminadora toma una máscara (real o predicha) con dimensiones $H \times W \times C$ y obtiene un mapa de probabilidad $H \times W \times 1$ que determina si un pixel corresponde a una máscara real o a una predicción.

A diferencia del artículo original, en el trabajo actual se hace uso de la arquitectura EFSNet (Hu and Wang (2020)) como red generadora, mientras que se mantiene la FCN (Long et al. (2015)) como red discriminadora.

Durante el entrenamiento se utilizarán tanto imágenes etiquetadas como imágenes sin etiquetar. Por tanto, se pueden diferenciar dos modos de entrenamiento que intervendrán en un momento dado y con un peso determinado, según la parametrización definida.

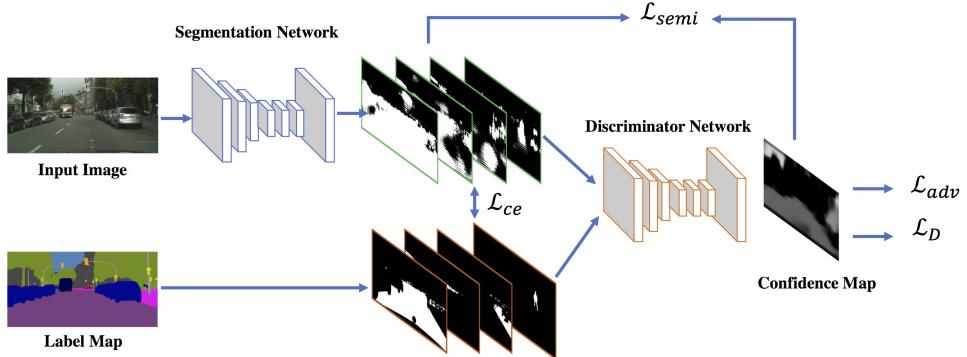


Figura A.1: Esquema de implementación de AL-S4. *Imagen extraída de GitHub* (Hung et al. (2018)).

En primer lugar, cuando se hace uso de imágenes etiquetadas se actualizan los pesos de la red con la propagación de las pérdidas de entropía cruzada, $\mathcal{L}_{cce}(G(I), M)$, (ecuación B.8) y la pérdida adversaria, $\mathcal{L}_{adv}(D(G(I)))$ (ecuación A.2).

En segundo lugar, cuando se hace uso de las imágenes no etiquetadas se utiliza la predicción de la red generadora como *input* de la red discriminadora para obtener un mapa de confianza que es utilizado como para establecer la pérdida semi-supervisada, $\mathcal{L}_{semi}(G(I), D(G(I)))$ (ecuación A.3). En este modo también se hace uso de la pérdida adversaria, $\mathcal{L}_{adv}(D(G(I)))$, como en el caso anterior ya que únicamente depende de la imagen de entrada.

El algoritmo de optimización que define esta técnica se puede esquematizar como se observa en la figura A.1 y se detalla en la sección de optimización, A.2

A.2. Optimización

El procedimiento parte de una imagen I y la máscara asociada M , y cuenta con una red generadora G y una red discriminadora D que conforman la *GAN*. Para la configuración del proceso de optimización es necesario definir el valor de, como mínimo, los siguientes hiperparámetros adicionales:

- λ_{adv} , λ_{semi} , $\lambda_{adv-semi}$: Peso de la pérdida adversaria, peso de la pérdida semi-supervisada y peso de la pérdida adversaria con imágenes no etiquetadas, respectivamente.
- s_{adv} , s_{semi} : Paso de entrenamiento de inicio de utilización de la pérdida adversaria y de imágenes no etiquetadas (pérdida semi-supervisada), respectivamente.
- T_{semi} : Umbral de confianza en la pérdida semi-supervisada.

- f_{label} : Fracción de conjunto de datos utilizados con etiqueta del total de datos.
Se ha definido $f_{label} = 7/8$ del conjunto de entrenamiento.

La combinación de estos hiperparámetros variarán considerablemente el resultado obtenido y deben ser ajustados a cada caso de uso en función de las características de los datos y redes neuronales utilizadas. En particular, es de gran importancia el número de muestras, la distribución de clases y el resultado obtenido sin la aplicación de esta técnica.

El método AL-S4 se puede resumir como el flujo que se observa en la imagen de la figura A.1.

En primer lugar, se aplica la red generadora sobre la imagen para obtener el mapa de segmentación, $G(I)$. De este mapa y la máscara se obtiene la pérdida de entropía cruzada: \mathcal{L}_{cce} .

En segundo lugar, se aplica la red discriminadora sobre el mapa de segmentación, $D(G(I))$, y la máscara, $D(M)$ de donde se extraen: la pérdida semi-supervisada, \mathcal{L}_{semi} , la pérdida del discriminador, \mathcal{L}_D , y la pérdida adversaria, \mathcal{L}_{adv} .

En último lugar, se optimizan los modelos generador y discriminador. Al modelo generador se le aplica la pérdida

$$\mathcal{L}_{seg} = \mathcal{L}_{cce} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{semi}\mathcal{L}_{semi} \quad (\text{A.1})$$

recordando que si se utilizan imágenes con máscara, caso supervisado, no se aplica \mathcal{L}_{semi} , y en el caso contrario no se aplica \mathcal{L}_{cce} . Al modelo dicriminador se le aplica \mathcal{L}_D .

Utilizando esta metodología de pérdidas y el aumento de imágenes visto en la sección 3.2.2.1 se han entrenado 20000 iteraciones de lotes de 8 imágenes utilizando un optimizador tipo Stochastic Gradient Descent, SGD, con decaimineto en los pesos de 10^{-4} , momento 0,9, paso de aprendizaje inicial $2,5 \cdot 10^{-4}$ y decaimiento polinómico con potencia 0,9 tal y como se propone en el trabajo de referencia (Hung et al. (2018)).

Las funciones de pérdida y su motivo de utilización se detalla a continuación.

Entropía Cruzada Categórica, \mathcal{L}_{cce}

La entropía cruzada categórica (ecuación B.8) se utiliza, como en la optimización clásica, para ajustar la red generadora a las máscaras reales. Define la diferencia entre la realidad y la predicción de forma que los pesos de la red puedan ser variados en pos de minimizar esta diferencia. Esta pérdida únicamente aplica cuando se utilizan imágenes etiquetadas.

En la prueba realizada se han mantenido los pesos por clase utilizados en proyecto principal mencionados en la subsección 3.2.2.2.

Pérdida Adversaria, \mathcal{L}_{adv}

La pérdida adversaria es una *BCE* en la que se considera que la máscara está totalmente formada por la clase correspondiente. Por lo tanto se puede definir como

$$\mathcal{L}_{adv} = -\log(\hat{x}_n) \quad (\text{A.2})$$

que es un caso particular de la *CCE* en la que únicamente hay dos clases, no hay peso por clase y el valor real siempre es el mismo para todos los píxeles.

En el algoritmo AL-S4, \hat{x}_n es la predicción del discriminador de la predicción del generador, es decir, $\hat{x}_n = D(G(I))$.

Dado que esta pérdida únicamente depende de la imagen inicial se puede utilizar durante todo el entrenamiento independientemente de si se está utilizando el conjunto etiquetado o el no etiquetado. Los autores recomiendan encarecidamente disminuir λ_{adv} al utilizar datos no etiquetados dado que puede llegar a sobrecorregir las predicciones (Hung et al. (2018)).

Con esta pérdida se entrena a la red generadora para *engañar* a la red discriminadora por medio de maximizar la probabilidad de que las predicciones del generador, $G(I)$, pertenezcan al conjunto de máscaras. De esta forma, se fuerza que la red generadora produzca mapas de probabilidad similares a las segmentaciones reales.

Los pesos que se utilizan en esta máscara son $\lambda_{adv} = 0,1$ y $\lambda_{adv-semi} = 0,001$, dependiendo de si se utilizan etiquetas o no, y su impacto se inicia en la primera iteración del entrenamiento, $s_{adv} = 0$, tal y como se propone en el artículo de referencia (Hung et al. (2018)).

Pérdida Semi-Supervisada, \mathcal{L}_{semi}

Esta pérdida es la utilizada cuando se utilizan imágenes no etiquetadas. Se utiliza para generar un mapa de confianza con el que se puedan inferir regiones de confianza con la distribución de las máscaras reales. La consecuencia directa de esta pérdida es la obtención de mapas de probabilidad con píxeles mejor definidos.

Los mapas de confianza se definen utilizando la predicción del discriminador sobre el mapa de segmentación, $D(G(I))$. Cuando estos valores superan cierto umbral, T_{semi} , serán tenidos en cuenta para la pérdida enmascarando los valores *dudosos*, $K(\cdot)$.

La definición de la pérdida se toma entonces como una *BCE* enmascarada entre la máscara auto-aprendida, $\hat{Y}_{n,c}$, definida elemento a elemento como: $\hat{Y}_{n,c^*} = 1$ si $c^* = argmax_c G(I_n)$; y la segmentación obtenida, $G(I_n)$.

$$\mathcal{L}_{semi} = - \sum_{c=1}^C K_n(D(G(I)) > T_{semi}) \cdot \hat{Y}_{n,c} \log(G(I_n)) \quad (\text{A.3})$$

Siguiendo las recomendaciones de los autores (Hung et al. (2018)), el peso de la pérdida en el global de las pérdidas utilizado es $\lambda_{semi} = 0,1$ y se inicia en el paso $s_{semi} = 5000$, es decir, un cuarto del total. Finalmente, se escoge el umbral que define la máscara tal que $T_{semi} = 0,2$.

Pérdida del discriminador, \mathcal{L}_D

La pérdida del discriminador es una *BCE* que debe determinar pixel a pixel si una predicción de la red discriminadora proviene de una máscara, $D(M)$, ó de un mapa de segmentación, $D(G(I))$. Esta es la forma habitual de optimización de redes discriminadoras en esquemas tipo *GAN*.

A.3. Resultados

Se ha aplicado la técnica AL-S4 al conjunto de datos visto (sección 3.1.3) utilizando como red generadora EFSNet y tratando de mantener estáticos todos los procedimientos y metodologías utilizadas de forma que se pueda asegurar que los cambios observados se deban exclusivamente al método AL-S4.

Todos los resultados se observan sobre el mismo conjunto de datos de entrenamiento y validación visto en el capítulo de Resultados, 4.

En primer lugar, se observa que las métricas numéricas, tabla A.1, han disminuido enormemente con lo que se ha perdido una gran precisión de forma global, tanto en entrenamiento como en validación.

En segundo lugar, una visualización de varias muestras, figuras A.2 y A.3, justifica las métricas, pero muestra una virtud de esta metodología que es la seguridad en la predicción. Los mapas de probabilidad están perfectamente delimitados aunque no se correspondan siempre con la realidad.

Se puede comprobar que la red tiene varios fallos debidos, principalmente, a la ponderación de las pérdidas, el paso de inicio de su utilización y la fracción de datos correspondiente a cada uno.

Por un lado, la omisión de categorías. En la prueba presentada las categorías predichas son *normal/benigno*. La clase *malingo* se ha visto sacrificada durante el entrenamiento. En pruebas similares se han obtenido resultados similares en los que se sacrifica la clase *benigno* por lo que se descarta que se deba a una clase sobre o infra-representada.

Por otro lado, la alucinación y omisión tumores con la misma *seguridad* (mapa de probabilidad) con la que se aciertan, figura A.3. Este es un problema peligroso debido al caso de uso. Un fallo de diagnóstico en la detección de tumores en ecografías mamarias puede ser determinante en la salud de una persona. En consecuencia, este es el principal problema que debe ser resuelto.

Tabla A.1: Comparativa de métricas clásico y AL-S4. Se detalla con asterisco (*) la metodología AL-S4.

(a) Clases: normal, benigno, maligno.

Conjunto	\bar{dsc}	$accuracy$	$mIoU$	IoU_{normal}	$IoU_{benigno}$	$IoU_{maligno}$
Entrenamiento	71.42	94.24	59.13	94.69	43.06	39.66
Entrenamiento*	44.71	91.26	38.67	92.56	23.46	0.00
Validación	70.44	94.06	58.14	94.22	43.45	36.74
Validación*	42.83	91.34	37.29	92.55	19.31	0.00

(b) Clases: normal, tumor.

Conjunto	\bar{dst}	$accuracy$	$mIoU$	IoU_{normal}	IoU_{tumor}
Entrenamiento	82.53	94.97	73.15	94.68	51.63
Entrenamiento*	72.75	92.82	62.67	92.56	32.77
Validación	79.72	94.49	69.79	94.22	45.36
Validación*	70.22	92.28	60.51	92.55	28.47

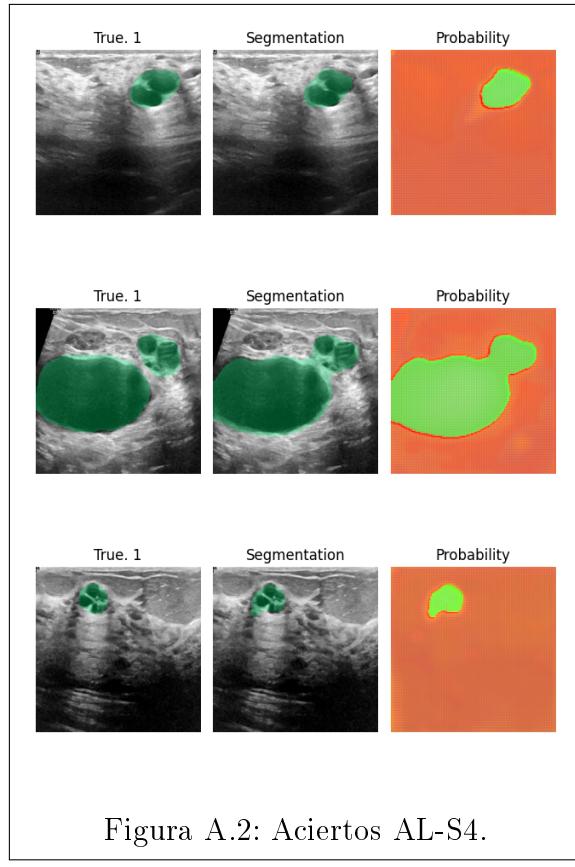


Figura A.2: Aciertos AL-S4.

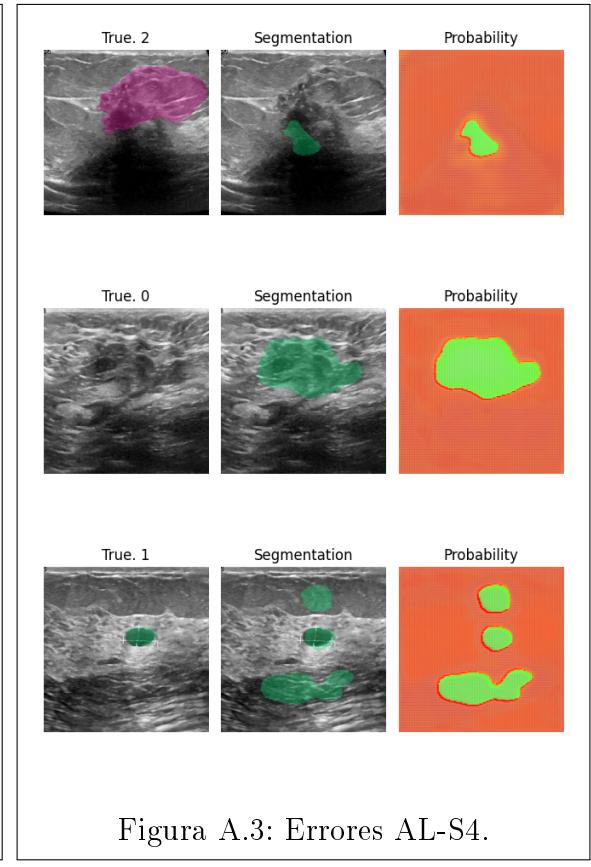


Figura A.3: Errores AL-S4.

La combinación de ambos problemas es debida a la parametrización del algoritmo. La definición de los 7 parámetros adicionales (sección A.2) que rigen el comportamiento expanden el espacio de posibilidades de forma abrupta y sin una guía detallada de las regiones de confianza o del impacto que tiene el número de clases o muestras de un problema dado.

Es necesaria una exploración intensiva del espacio de parámetros para obtener una mejora de la solución supervisada clásica.

A.4. Conclusiones

La utilización de la técnica AL-S4 mejora el mapa de probabilidad de la red generadora promoviendo una mayor seguridad en la red. La desventaja de este método es el alto grado de parametrización que requiere para no incurrir en errores.

En la prueba efectuada se han obtenido peores resultados que en el caso original ya que la pérdida \mathcal{L}_{adv} y \mathcal{L}_{semi} corrompen la optimización de \mathcal{L}_{cce} con lo que se termina con una red neuronal que no es capaz de predecir la clase *maligno* en ningún caso y que alucina/omite tumores.

No hay duda de que esta vía de desarrollo es prometedora por diversos motivos. En primer lugar, se alinea con la investigación en optimización semi-supervisada que es tan utilizada debido a la falta de etiquetas en el campo de la biomedicina. En segundo lugar, mejora la seguridad de la predicción de la red generadora, figura A.2, sin impactar en la latencia cuando se realiza inferencia.

Aún siendo una vía prometedora, es necesario recalcar la dificultad de definición de los hiperparámetros y el impacto de los mismos en el resultado final.

Apéndice B

Formulación matemática

B.1. BatchNorm y sesgo

Si

$$\hat{x}_k = W \cdot x_k + b_k \quad (\text{B.1})$$

es la salida de una capa convolucional sin activación, entonces la normalización por lote, *BatchNorm*, aplica

$$\hat{x}'_k = (\hat{x}_k - \mathbb{E}(\hat{x}_k)) / \sqrt{\text{Var}(\hat{x}_k)} \quad (\text{B.2})$$

donde

$$(\hat{x}_k - \mathbb{E}(\hat{x}_k)) = W \cdot x_k - \mathbb{E}(W \cdot x_k) \quad (\text{B.3})$$

$$\text{Var}(W \cdot x_k + b_k) = \text{Var}(W \cdot x_k) + \varepsilon \quad (\text{B.4})$$

con lo que

$$\hat{x}'_k = (W \cdot x_k - \mathbb{E}(W \cdot x_k)) / \sqrt{\text{Var}(W \cdot x_k) + \varepsilon} \quad (\text{B.5})$$

que no depende del sesgo, b_k , (Ioffe and Szegedy (2015)). El parámetro ε que se deriva de la varianza del sesgo añade un desplazamiento que será corregido con la introducción de una activación *PReLU* (He et al. (2015)), *Parametrized ReLU*, que contiene un parámetro entrenable que define la pendiente en la parte negativa de los números reales, \mathbb{R}^- , y se utilizará para compensar el desvío de *BatchNorm*.

B.2. Número de parámetros en convoluciones desagregadas

Considerando dos vías de convolución de un mapa de características:

1. Capa convolucional usual: tamaño de kernel k_s y stride s .
2. Capas desagregadas: tres capas convolucionales consecutivas tal que la primera aplica el stride, s , con $k_0 = (s \times s)$ para variar la dimensión, la segunda realiza la extracción de características utilizando $k_1 = k_s$ y, finalmente, la tercera aumenta la dimensión de los canales utilizando $k_2 = (1 \times 1)$

El número de parámetros de una capa convolucional usual viene dado por

$$k_s^2 \cdot C_{in} \cdot C_{out} \quad (\text{B.6})$$

mientras que el número de parámetros de las capas desagregadas son

$$s^2 \cdot C_{in}^2 + k_s^2 \cdot C_{in}^2 + C_{in} \cdot C_{out} = C_{in} (C_{in} (s^2 + k_s^2) + C_{out}) \quad (\text{B.7})$$

con C_{in} el número de canales de entrada y C_{out} el número de canales de salida.

Suponiendo $s = 2$ para una reducción del tamaño original en un factor 2, un $k_s = (3 \times 3)$ y $C_{out} = sC_{in} = 2C_{in}$ se tiene sin desagregar: $params_0 = 18 \cdot C_{in}^2$ y desagregado: $params_1 = 15 \cdot C_{in}^2$.

Se han supuesto estos parámetros dado que son los utilizados habitualmente en EFSNet (Hu and Wang (2020)) pero se puede comprobar que es extensible a todo caso en que $(s^2 + k_s^2 + f) < k_s^2 \cdot f$ siendo $f = \frac{C_{out}}{C_{in}}$ el factor de aumento en los canales.

Debido a la utilización de *BatchNorm* (sección B.1) no se ha tenido en cuenta el sesgo.

B.3. Entropía cruzada

La entropía cruzada utilizada se define en la ecuación B.8 y se entiende como la suma ponderada por los pesos de cada clase del logaritmo de la normalización exponencial, *softmax*, de la salida de la red neuronal multiplicada por los valores reales.

$$\mathcal{L}_{CCE} = - \sum_{c=0}^C \omega_c \log \left(\frac{\exp(\hat{x}_{n,c})}{\sum_{c=0}^C \exp(\hat{x}_{n,c})} \right) y_{n,c} \quad (\text{B.8})$$

donde el subíndice n es correspondiente a la muestra del lote, el subíndice c es referido a la clase, ω_c es el peso definido para la clase c , $\hat{x}_{n,c}$ es la salida de la red neuronal y $y_{n,c}$ es el valor real.

B.4. Accuracy

La métrica *accuracy* o exactitud define el ratio entre los aciertos y el total.

$$\text{accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (\text{B.9})$$

donde TP son los verdaderos positivos, TN los verdaderos negativos, FP los falsos positivos y FN los falsos negativos.

B.5. Coeficiente de similaridad de Dice

El coeficiente de similaridad de Dice, dsc , o de Sørensen-Dice se trata de un estadístico para determinar el grado de similaridad entre dos muestras A y B . Se define como el ratio entre dos veces la intersección sobre el total.

$$dsc = \frac{2 |A \cap B|}{|A| + |B|} \quad (\text{B.10})$$

donde $|A \cap B|$ es la intersección entre ambas muestras.

B.6. Índice de Jaccard

El índice de Jaccard, comúnmente denominado coeficiente de Intersección sobre Union (*IoU*), se define como el ratio entre la intersección entre la unión de dos conjuntos A y B .

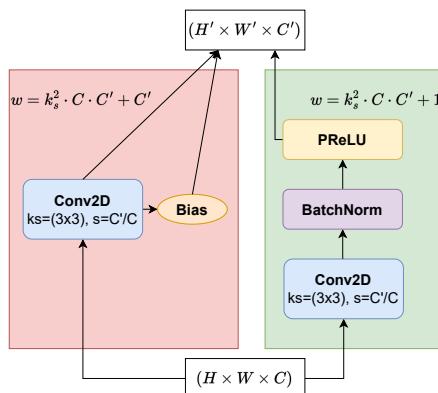
$$ji = \frac{|A \cap B|}{|A \cup B|} \quad (\text{B.11})$$

donde $|A \cap B|$ es la intersección y $|A \cup B|$ la unión entre ambas muestras.

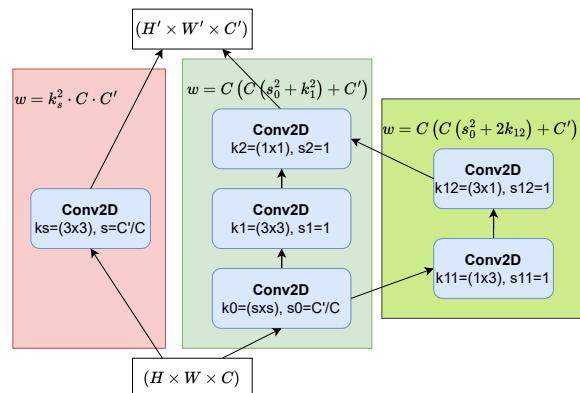
Apéndice C

Imágenes adicionales

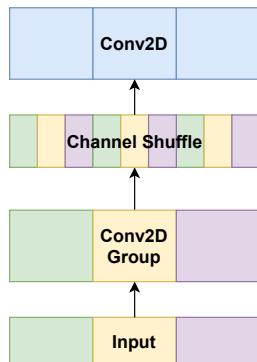
C.1. Principales metodologías para convolución eficiente



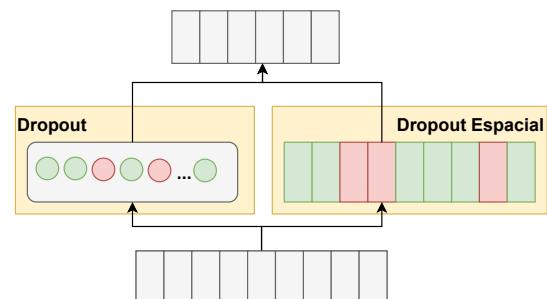
(a) Normalización por lote. A la izquierda la convolución usual con sesgo. A la derecha la convolución eficiente con *BatchNorm* y *PReLU*.



(b) Convolución Desagregada y Factorizada. A la izquierda la convolución usual. En el centro la convolución desagregada. A la izquierda la factorización del bloque central de la convolución factorizada. *No se muestra el sesgo.*

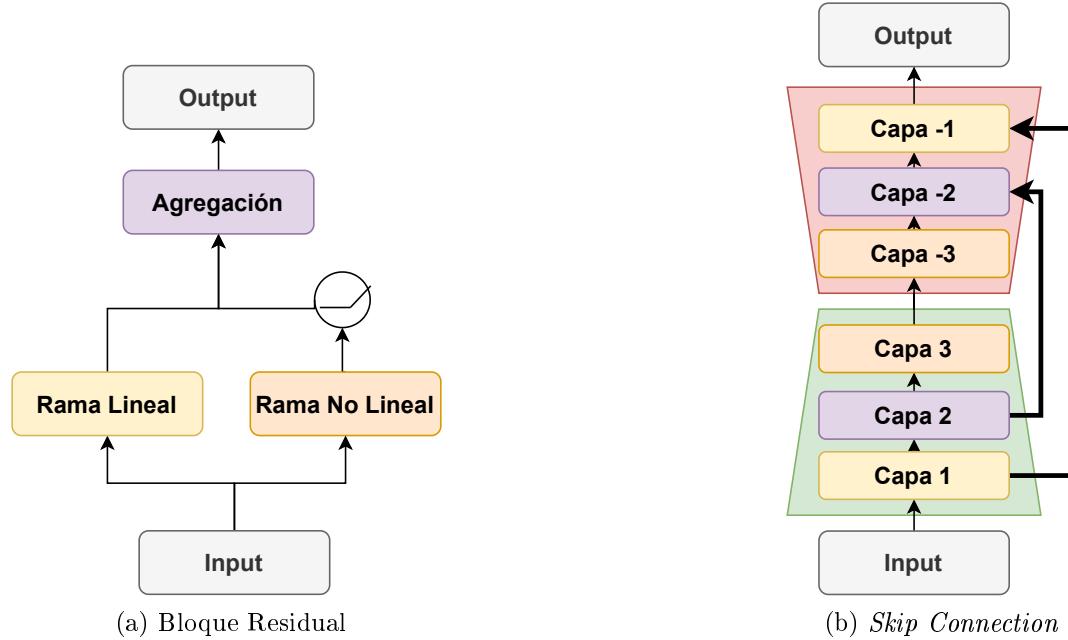


(c) Convolución por grupo con barajado de canales. Se muestran en colores (tres) las agrupaciones de los canales.



(d) Dropout Espacial. A la izquierda el Dropout clásico (*eliminación de la propagación de varias neuronas en una capa*). A la derecha el Dropout Espacial en el que se elimina la propagación de canales aleatorios (rojo) de una imagen.

Figura C.1: Reducción de parámetros en operaciones de convolución.

Figura C.2: Bloque residual y *skip-connections*

C.2. Transformaciones de aumento de datos

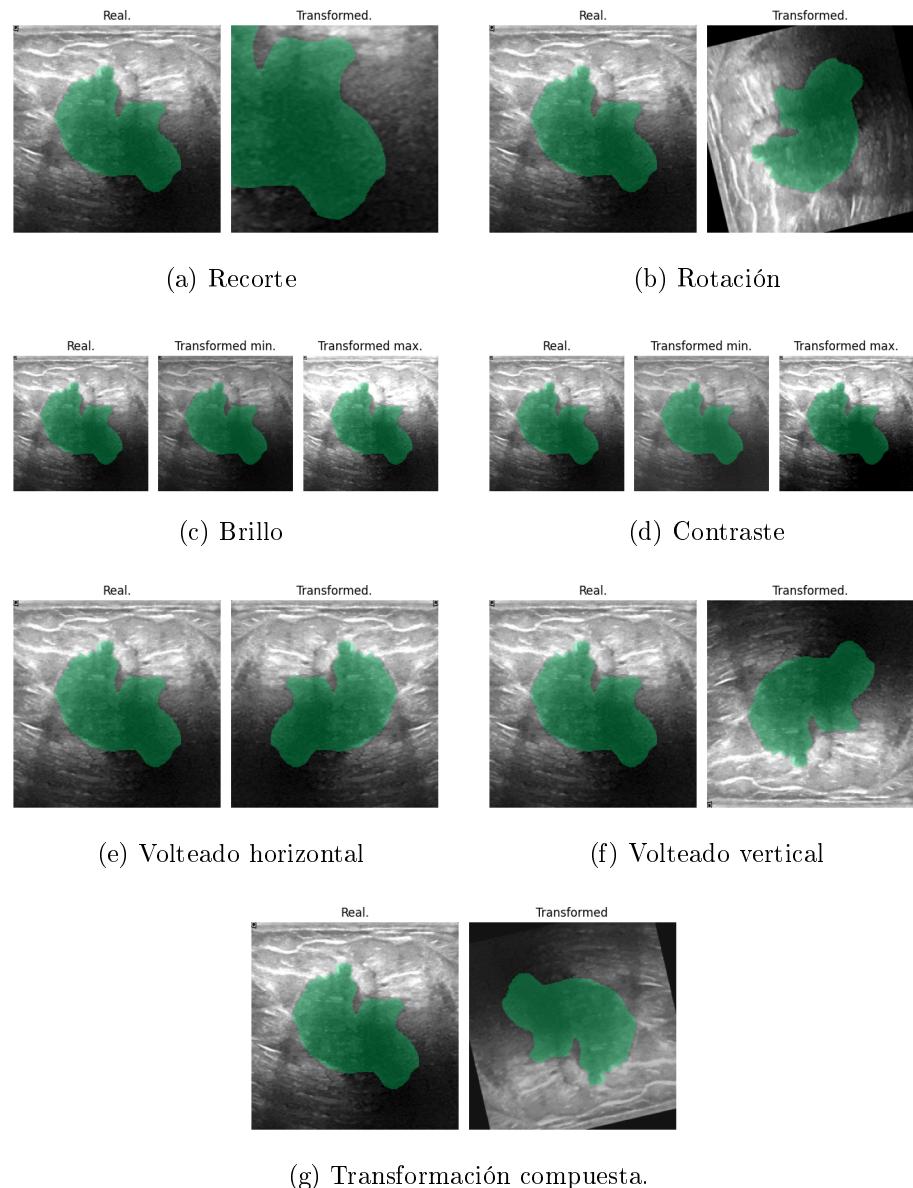


Figura C.3: Transformaciones de aumento de datos.

C.3. Matrices de confusión

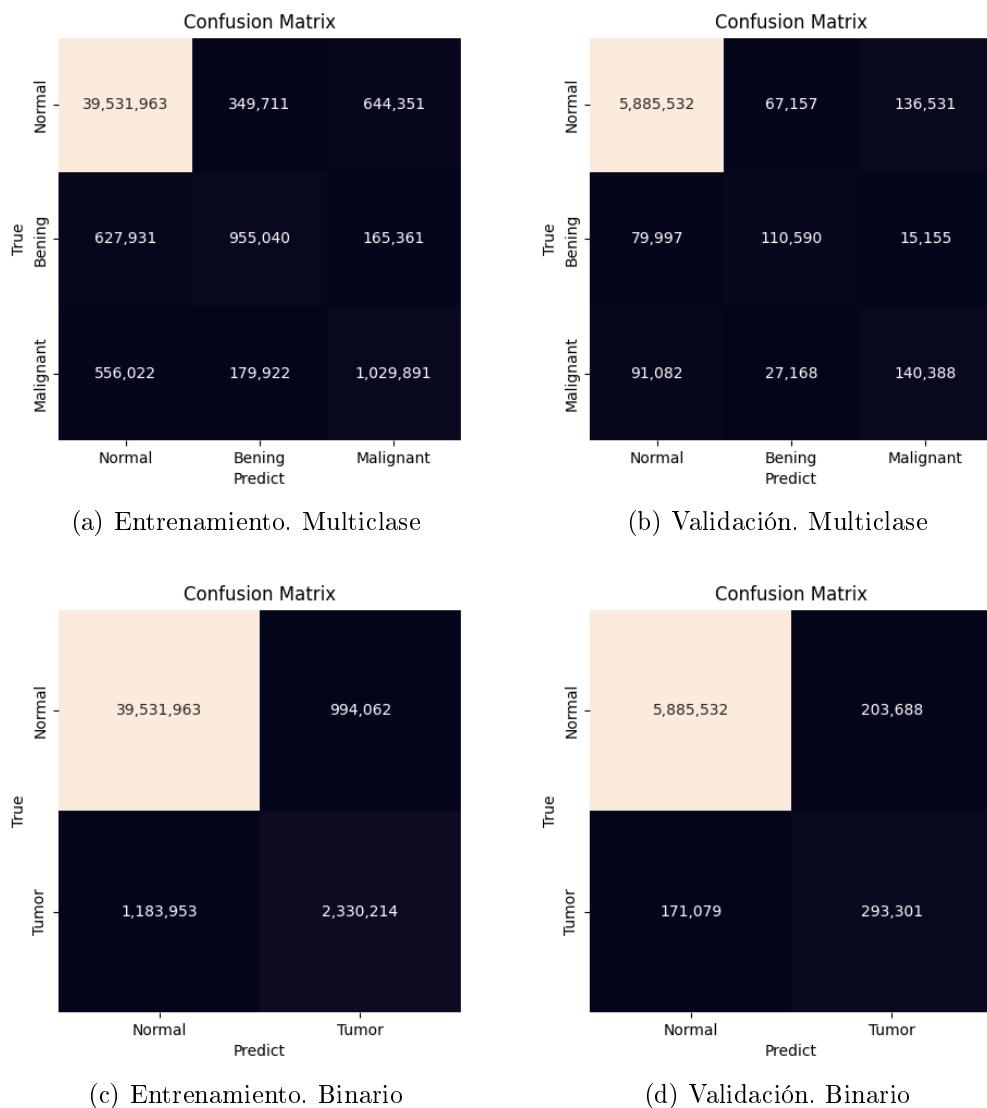


Figura C.4: Matrices de confusión.