UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Trabajo Fin de Máster del
máster de Ingeniería y Ciencia de Datos

Segmentación interactiva apoyada en segmentación automática aplicada a la detección y caracterización del cáncer de mama

Jose Luis Cuenca Reyes

Dirigido por: Mariano Rincón

Margarita Bachiller Mayoral

Curso: 2021-2022: Septiembre

Índice

Segmentación interactiva apoyada en segmentación automática aplicada a la detección y caracterización del cáncer de mama

Cuenca, J. L., Bachiller, M., Rincón-Zamorano, M.

Septiembre 2022

Resumen

El cáncer de mama es el tipo de cáncer más común entre las mujeres. En 2018 se diagnosticaron 32.825 nuevos casos en España (uno cada 15 minutos) y 266.120 en EEUU. Se estima que el 12,5% de las mujeres de los países de renta alta (una de cada 8) desarrollará un cáncer de mama a lo largo de su vida y más del 4% (aproximadamente una de cada 24 mujeres) morirá por esta causa. La tasa de supervivencia es del 99% cuando la enfermedad se detecta precozmente, pero desciende al 22% cuando se detecta en estado IV. La detección precoz también ahorra recursos económicos: tratar el cáncer de mama en el estado 4 es tres veces más caro que en el estado 0. Por esto, todos los países están interesados en implantar programas de cribado. En la actualidad se utilizan diferentes técnicas de imagen con este fin, cada una con sus ventajas e inconvenientes. Este proyecto tiene por objetivo utilizar técnicas de deep learning para automatizar el análisis de estas imágenes.

En este trabajo se presenta un proceso de preparación de la información para el entrenamiento de modelos de segmentación interactiva de imágenes basado en la predicción previa obtenida mediante segmentación automática y un aporte extra de información sobre las regiones de la imagen donde sí existe lesión y donde no existe, obtenido de las interacciones del usuario con la propia imagen a segmentar.

Los resultados experimentales sobre el conjunto de datos de masas anómalas en senos CDD-DDSM muestran que nuestra propuesta mejora los resultados de la segmentación, además de ofrecer un mecanismo para automatizar la extracción de la información adicional para segmentación interactiva y analizar las implicaciones de utilizar una estrategia u otra de selección de la información adicional así como de la configuración de esta utilizada para entrenar el modelo IIS.

1. Introducción

El cáncer de mama es el cáncer más común, así como la causa más común de muerte por cáncer en mujeres a nivel mundial [44], con más de 2,2 millones de casos en 2020. La detección precoz del cáncer es uno de los factores determinantes en la supervivencia del paciente, contando para ello con diferentes pruebas diagnósticas como son las imágenes, menos invasivas y la biopsia que supone la extracción de tejido, en nuestro caso, de la mama. Entre las técnicas de diagnóstico por imagen existen diferentes alternativas como son la ecografía, mamografía, termografía, resonancia magnética o tomografía por emisión de positrones. De entre todas ellas, la mamografía ha demostrado su efectividad para la detección temprana de cáncer, por ello, es una de las pruebas más empleadas.

Las mamografías son imágenes de la mama captadas mediante rayos X que permiten encontrar signos de la presencia de lesiones. Existen diferentes tipos de lesiones: calcificaciones, nódulos o masas, asimetría en la densidad o distorsión de la arquitectura. Un examen mamográfico típico consiste en la obtención de dos proyecciones de cada mama: Cráneo-Caudal (CC) y Medio Lateral Oblicuo (MLO) con el fin de cubrir la totalidad de la mama. Un ejemplo de ellas se muestra en la figura 1. El radiólogo examinará ambas proyecciones para identificar áreas anormales o lesiones y la densidad del seno.

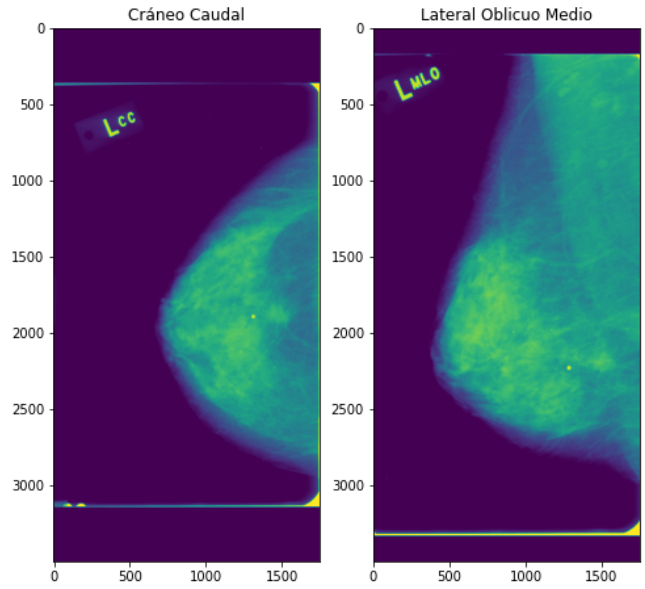

Figure 1: Las dos proyecciones típicas utilizadas para realizar la mamografía: (1) proyección Cráneo-Caudal y (2) Medio Lateral Oblicuo.

El análisis de la densidad en las mamografías se debe a la dependencia de la sensibilidad de la prueba con la densidad mamográfica, esto es, cuanto mayor sea la densidad menor es la sensibilidad de la prueba. La densidad del seno es una medición de la cantidad de fibra de una mama respecto la cantidad de tejido graso. Un seno será denso si tiene alta cantidad de fibra. El tejido mamario denso dificulta que los radiólogos puedan detectar el cáncer debido a que este tipo de tejido se ve blanco en las mamografías al igual que las masas o tumores, mientras que el tejido graso se ve casi negro.

Evidentemente, la detección es más fácil en senos grasos en los que el fondo de la imagen es casi negro que en senos fibrosos en los que el fondo aparecerá blanco. A modo de ejemplo, en la figura 2 se muestran dos imágenes en las que se observa la dificultad que existe en la detección de anomalías en una mama de tejido fibroso frente a una con el tejido más graso.

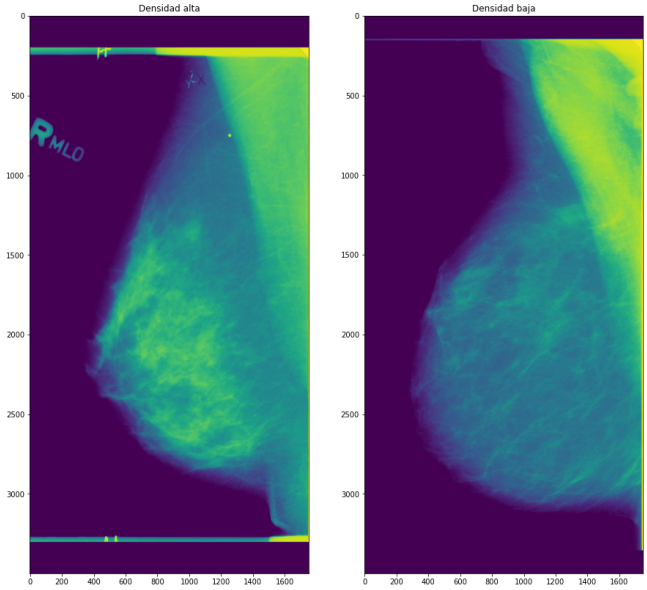

Figure 2: Proyección Medio Lateral Oblicuo de dos senos con diferente densidad de fibras. Mientras que el primero se trata de un seno muy denso donde es difícil apreciar estructuras anómalas, el segundo seno presenta una densidad de fibras menor por lo que es más sencillo detectar anomalías.

Actualmente, es el radiólogo quien interpreta la mamografía y detecta las anomalías. Este proceso presenta dos problemas: La gran cantidad de tiempo dedicado por parte de los especialistas a la tarea de identificar las regiones en cada imagen y los sesgos que el propio especialista puede inducir en la detección [50]. Por ello, resulta muy conveniente disponer de sistemas automáticos o semiautomáticos para la detección de anomalías de manera que se facilite la labor de los radiólogos, bien a través de herramientas automatizadas para la segmentación o mediante propuestas de segmentación en las que únicamente tengan que realizar modificaciones para lograr una segmentación precisa. Minimizar esas modificaciones es clave para conseguir reducir el tiempo que se invierte en anotar las imágenes.

El desarrollo de estos sistemas automáticos es complicado debido a la dificultad en el análisis de las imágenes, en especial, las de apariencia blanquecina por su alta densidad. En este sentido, desde hace años se están desarrollando sistemas de visión por computador que permiten segmentar las lesiones e incluso clasificarlas en malignas o benignas presentando resultados satisfactorios especialmente cuando se aplican técnicas de aprendizaje profundo (en inglés Deep Learning, DL). Sin embargo, éstas entran dentro de las técnicas supervisadas de aprendizaje máquina en las que se requiere un elevado número de muestras, anotadas por los expertos, para poder entrenar a la red. De nuevo, tenemos la necesidad de las anotaciones de las imágenes por el radiólogo.

Por todo ello, dada la limitación de imágenes anotadas disponibles, es lógico pensar que los sistemas basados en DL tendrán limitaciones en el aprendizaje. Nuestra propuesta es el diseño

de un sistema de visión semiautomático, centrado en la segmentación de masas de mamografías MLO y CC, que muestre una solución inicial de segmentación, que llamaremos segmentación automática a partir de la cual, con la menor participación posible del experto médico, realice una segmentación más precisa, que llamaremos segmentación interactiva.

2. Objetivos

Nuestro principal objetivo es la construcción de un sistema de segmentación automático o semiautomático de imágenes aplicado a mamografías de proyección Cráneo-Caudal y Medio Lateral Oblicuo para la detección de masas, que proponga a los radiólogos una segmentación precisa utilizando el menor número de interacciones posibles con la imagen, lo que supone el ahorro de tiempo y esfuerzo además de evitar subjetividades.

Para llevar a cabo este objetivo se han definido los siguientes objetivos parciales:

- Seleccionar la base de datos, analizarla y prepararla para su uso.
- Analizar las estrategias empleadas en otros trabajos para la segmentación de imágenes en general y la segmentación de mamografías en particular.
- Seleccionar una arquitectura para la segmentación automática y evaluar los resultados.
- Proponer estrategias de mejora basadas en la segmentación automática para mejorar la precisión en la detección de las masas.
- Evaluar los resultados de las estrategias propuestas.
- Implementar una prueba de concepto (PoC) para validar la viabilidad técnica de la solución.

3. Trabajo relacionado

El diagnóstico por imagen ha demostrado ser una de las herramientas más eficaces de la medicina moderna capaz de ofrecer información detallada sobre las estructuras internas de los pacientes de manera no invasiva. Con el fin de poder estudiar y analizar diferentes partes del cuerpo humano se han desarrollado múltiples herramientas capaces de generar imágenes diagnósticas en dos, tres o cuatro dimensiones utilizando una variedad de técnicas que incluyen los ultrasonidos (ultrasonografía), rayos X (radiografías, CT, mamografías...) o campos magnéticos (MRI). Este incremento y extensión del uso de las técnicas de diagnóstico por imagen supone también un incremento en el número de imágenes que los radiólogos deben analizar, redundando en los posibles errores que se cometan al analizar las imágenes.

La segmentación de imágenes es la rama de la visión artificial que puede ayudar en esta labor. La segmentación de imágenes consiste en la definición sin solape de regiones homogéneas respecto a una característica física común de los píxeles de la imagen. Esto implica la necesidad de definir

para el problema en que se esté trabajando las regiones que se quieren identificar durante la segmentación. Cada técnica de diagnóstico presenta sus propios problemas, por ejemplo, en el caso de MRI es habitual el problema de la *no-uniformidad de la intensidad* mientras que CT adolece del problema del *promedio de volumen parcial*. El ruido en la imagen es un problema común a todas las técnicas.

Entre las técnicas más estudiadas para la segmentación de imágenes médicas podemos destacar:

- *Tresholding*: En esta aproximación la segmentación se obtiene realizando particiones de las intensidades de la imagen. Aunque es una técnica sencilla de implementar obtiene buenos resultados en imágenes donde diferentes estructuras presentan intensidades diferentes u otra característica cuantificable. Se han propuesto modificaciones a esta aproximación para aplicar umbrales locales frente a un solo conjunto de umbrales globales como en [30] o para añadir información basada en la conectividad [29]. Esta técnica es sensible al ruido y la no-uniformidad de la intensidad de la imagen, además la selección de los umbrales es un factor crítico para obtener una buena segmentación.

- *Edge based*: La detección de bordes es una técnica que intenta identificar los píxeles que separan dos regiones diferenciadas por los valores de los píxeles próximos al borde. De acuerdo con [22], los métodos para la detección de bordes se pueden basar en el histograma de grises de la imagen o en el gradiente.

- *Region based*: En este conjunto de aproximaciones podemos considerar dos tipologías, *regiones crecientes* y *separación y fusión de regiones*. En el caso de las *regiones crecientes*, dado un punto semilla se verifica si los puntos próximos al punto semilla cumplen con un criterio de homogeneidad, si cumplen el criterio se incluyen en la región del punto semilla. Esta aproximación se aplica iterativamente haciendo crecer la región segmentada. Esta aproximación se ha utilizado para la delimitación de tumores y lesiones como en [12] o [48]. La aproximación por *separación y fusión de regiones* se apoya en árboles cuaternarios que segmentan la imagen completa en cuatro cuadrantes para a continuación intentar fusionar estos cuadrantes basado en su uniformidad. El proceso de separación y fusión es repetido recursivamente sobre cada región persistente de la iteración anterior hasta llegar a la unidad de información. A diferencia de la aproximación *region growing* no es necesario definir un punto semilla manualmente [35].

- *Clustering*: Estas aproximaciones se enmarcan en el área del aprendizaje no supervisado. Es necesario definir un criterio de similitud para el algoritmo de agrupamiento, tal que se maximiza la similitud intragrupos y se minimiza la similitud intergrupos. Los algoritmos más habituales han sido K-Means [6] y hard y fuzzy C-Means [9], [3]. Estas aproximaciones son sensibles al ruido y a la no-uniformidad de la intensidad.

- *Modelos deformables*: Son superficies artificiales y cerradas definidas por el usuario capaces de expandirse o contraerse debido a la influencia de la imagen para ajustarse a esta. Los tipos más comunes son los modelos de minimización de energía [37] y métodos basados en conjuntos de niveles [52].

- *Atlas*: Estos métodos se apoyan en un *atlas* existente previamente que contiene información procedente de múltiples pacientes acerca de la anatomía, forma, tamaño y otras características representativas de la región de interés. Esto permite tener probabilidades

a priori para cada píxel de las imágenes a segmentar [4]. También existen aproximaciones con plantillas difusas [61]. El principal problema que tiene esta aproximación es la cantidad de información necesaria para componer el atlas y que se requiere de supervisión experta.

- *Redes Neuronales Artificiales*: Las redes neuronales han ganado importancia en las tareas de segmentación de imágenes consiguiendo segmentaciones precisas en diferentes campos de la visión artificial [33].

En el ámbito de las mamografías, recientemente se ha experimentado con técnicas clásicas de segmentación, de machine learning y de deep learning como podemos apreciar en la figura 3 obtenida de [38].

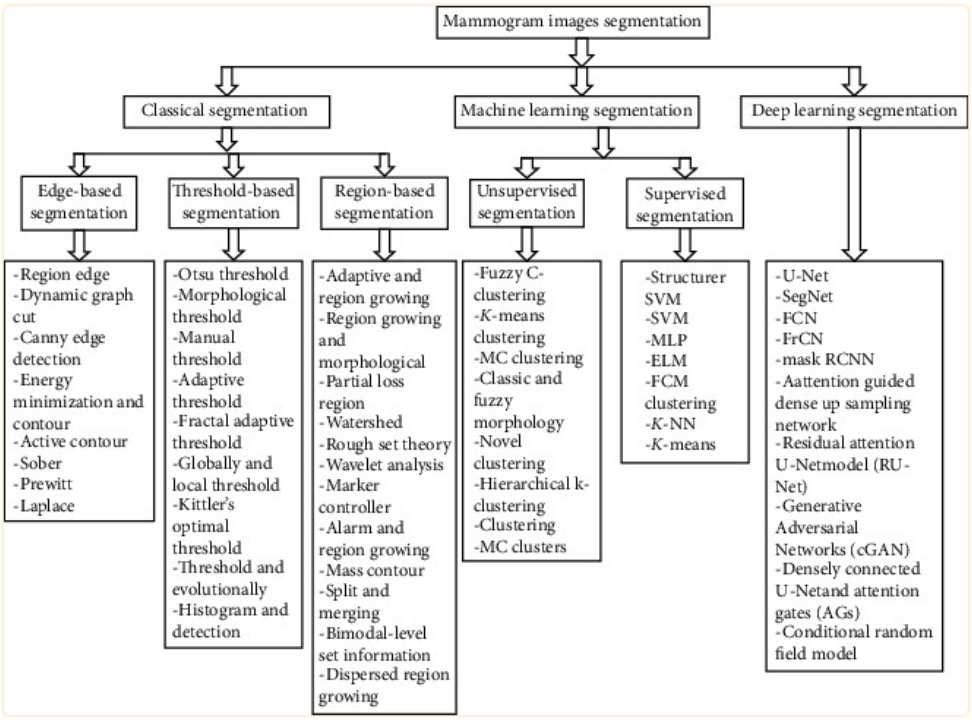

Figure 3: Clasificación de las técnicas de segmentación de imagen mamográfica por [38].

La mayoría de las técnicas clásicas de segmentación de imagen se apoyan en el valor del píxel para segmentar y requieren la selección manual realizada por expertos de determinados valores y características (umbrales, puntos semilla, etc). Fallar en seleccionar estás características iniciales puede suponer el fracaso del clasificador [1]. Otros problemas que dificultan la obtención de resultados con estas técnicas incluyen la sensibilidad al ruido, baja aplicabilidad en mamogramas con bajo nivel de contraste o dificultad para seleccionar niveles globales [38].

En el área de machine learning se ha experimentado con modelos supervisados y no supervisados. Mientras que los métodos no supervisados son sencillos de implementar y no requieren de información previamente etiquetada, tienen como principal problema la selección a priori del número de grupos a identificar [46], [38], [57], para lo cual no se puede establecer un nivel global óptimo debido a la varianza de lesiones que pueden existir de paciente a paciente. En aprendizaje supervisado, el principal problema de las técnicas clásicas de machine learning es que requieren de

la extracción manual o semi-manual de las características para poder definir los modelos, lo que dificulta su desarrollo [1], [40]. Esta extracción se basa en diferentes características apreciables por un **experto** (como textura, forma, niveles de gris). Una mala extracción de características provoca un mal rendimiento de los modelos entrenados [40], [56], [54].

La segmentación mediante técnicas de deep learning ofrece múltiples ventajas [53], [38], [47], [14] donde cabe destacar la capacidad de aprendizaje adaptativo para solucionar problemas complejos, capacidad de auto-organización no requiriendo inicializaciones externas ni intervenciones de expertos a priori, aprendiendo a extraer las características representativas de las imágenes de entrenamiento sin extracción manual y la capacidad de ejecutar en tiempo real por la alta paralelización de las redes neuronales.

Una de las arquitecturas de deep learning con las que más se ha experimentado ha sido Recurrent Neural Network (RNN). Esta arquitectura fue desarrollada originalmente para análisis discreto secuencial y puede ser vista como una generalización de Multi Layer Perceptron (MLP) [32]. Esta arquitectura se ve afectada por los problemas del desvanecimiento y la explosión del gradiente [15], por lo que para mitigar estos problemas se han propuesto modificaciones como Long-short Term Memory (LSTM) [16] que incorpora *unidades especializadas de memoria* o Gated Recurrent Unit (GRU) [5].

Los elementos de las redes neuronales convolucionales se propusieron en los años 80, sin embargo, debido a los requisitos computacionales para entrenar estos modelos, no fue hasta 2012 con AlexNet [25] que se reactivo el interés y la investigación en este área. Las dos principales aportaciones de AlexNet se pueden resumir en que la profundidad de la red mejora su capacidad de aprendizaje y la posibilidad de utilizar GPUs durante el costoso proceso de entrenamiento. Las principales ventajas de las redes neuronales convolucionales (CNN) son que producen redes neuronales más compactas que requieren de menor número de parámetros a entrenar, lo que reduce los riesgos de sobre ajuste y tienen en consideración el contexto local de cada píxel, permitiendo extraer información sobre características de tamaño, forma, disposición, entre otras. Esto mejora las expectativas con respecto a los modelos clásicos de ML que relegan la extracción de características para el entrenamiento de los modelos a un proceso manual o semi-manual con los inconvenientes que tiene asociado y a otras arquitecturas de DL donde no se tiene en consideración la información espacial de la imagen.

Las redes convolucionales completas (FCN) [33] son una mejora de las CNN en donde se prescinde de todos los elementos que no sean convolucionales, típicamente la capa densa de salida de las CNN, de forma que todas las capas de la red están formadas únicamente por elementos convolucionales. Esto permite a la red efectuar una segmentación efectiva a nivel de píxel, dando a cada píxel de la imagen completa su correspondiente etiqueta. Además, en tareas de segmentación de imagen médica, los resultados de FCN han demostrado superar los resultados en pruebas similares utilizando CNN [42] y [59].

En el ámbito de las FCN, la arquitectura UNET [51] se ha impuesto como una de las más habituales y de mayor aplicación en tareas de segmentación de imágenes, existiendo diferentes variaciones a la implementación básica para empujar todavía más las capacidades de la arquitectura. Esta arquitectura es una FCN organizada en dos subredes que implementan un codificador y un decodificador unidos por un puente entre ambas subredes, de forma que la subred de contracción (el codificador) comprime la información de entrada en una representación de espacio

latente, mientras que la subred de expansión (el decodificador) trata de identificar los patrones subyacentes a la información codificada. UNET incorpora *canales de características* (feature channels) entre los niveles relativos de la subred de contracción y la subred de expansión, que permiten propagar información de contexto hacia la subred de expansión.

Entre las variaciones de UNET, podemos destacar Residuals UNET [60] la cual aporta una modificación a la arquitectura para incluir unidades residuales [13] que permiten incrementar la profundidad de la red sin sufrir las penalizaciones del desvanecimiento de gradientes y Attention UNET [43] que aporta la inclusión de puertas de atención, que son bloques especializados capaces de aumentar o disminuir la relevancia de determinadas regiones en la imagen mediante un sistema de pesos blandos. Esta arquitectura y sus variantes han sido aplicadas a diferentes problemas de segmentación automática e interactiva en el campo de la medicina, encontrando ejemplos para detección de la membrana celular [51], detección de páncreas, hígado y bazo en imágenes 3D CT [43], clasificación histopatológica [2], detección del melanoma a través de lesiones cutáneas [20], detección de pólipos intestinales [21], entre otras aplicaciones.

4. Metodología

En este estudio se desarrolla un sistema semiautomático para la segmentación precisa de masas en imágenes de mamografía utilizando técnicas de DL. En el diseño del sistema se han seguido dos fases. En la primera, se diseña un subsistema para la segmentación automática que toma como entradas las mamografías y las máscaras para poder evaluar la segmentación propuesta durante el aprendizaje de la red y, en la segunda, se diseña un subsistema para la segmentación precisa que toma como entradas las mamografías, la segmentación automática anterior e información adicional. Aunque es en la segunda fase en la que se desarrollan la mayoría de los experimentos de este trabajo, fue necesario implementar la primera fase para disponer de una segmentación inicial. En la figura 4 se muestra el esquema del sistema implementado, en la que la información adicional utilizada durante la segmentación interactiva se genera mediante el *Módulo Generador de Información Adicional*. A su vez, la entrada a éste módulo puede proceder del experto médico o bien del *Módulo Generador de Interacciones*.

La *información adicional* se determina a partir de la información relativa a las correcciones que deben realizarse, aportada por el radiólogo durante la interacción con el sistema de segmentación. Debido a las dificultades de la segmentación manual de imágenes de mamografía comentadas previamente y al no disponer de un experto que nos apoye en el proceso de entrenamiento del modelo para la fase II de segmentación interactiva, fue necesario automatizar la tarea de extraer información sobre los aciertos y errores de la segmentación automática para calcular la información adicional con la que entrenar el modelo de segmentación interactiva, refinando así la segmentación automática previa. Esto se consigue en el módulo generador de interacciones simuladas que es explicado más adelante. Para el entrenamiento de los modelos de cada fase se realizó un ajuste fino independiente de los hiperparámetros.

Todos los desarrollos llevados a cabo durante el trabajo se han realizado con Python (versión 3.7.9) debido a las herramientas y facilidades que ofrece para el trabajo con datos y la construcción de modelos. Entre las principales bibliotecas utilizadas tenemos:

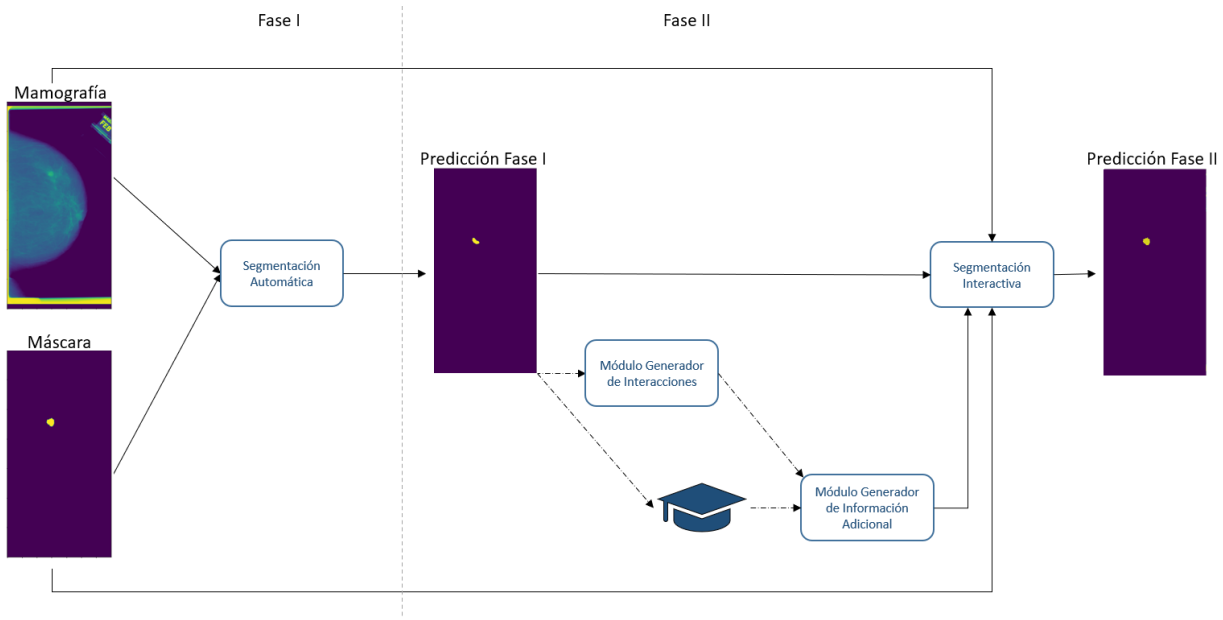

Figure 4: Descripción del proceso implementado para la segmentación de masas en imágenes de mamografía.

- *NumPy*: NumericalPython. Se trata de la biblioteca *por defecto* para computación científica en Python. Ofrece un objeto 'array' multidimensional, varios objetos derivados y el conjunto de rutinas y funciones comúnes necesarios para operaciones rápidas sobre los arrays, incluyendo operaciones matemáticas, logícas, manipulación de la silueta, ordenamiento, selección, I/O, transformaciones discretas de Fourier, operaciones estadísticas básicas y pseudoaleatoreidad entre muchas otras capacidades.

- *Pandas*: Pandas es una biblioteca rápida, potente, flexible y sencilla de utilizar para la ingesta de diferentes fuentes de datos y su manipulación.

- *OpenCV*: Open Source Computer Vision Library. Es una biblioteca escrita en C++ opensource bajo la licencia BSD que incluye multitud de métodos enfocados a tareas de visión por computador.

- *ImageIO*: Esta es una biblioteca Python que ofrece una interfaz de conveniencia para leer y escribir desde un amplio rango de formato de imágenes, incluyendo formatos específicos.

- *Keras*: Es un API de alto nivel para aprendizaje profundo trabajando con TensorFlow como backend. Esta biblioteca ofrece la posibilidad de definir modelos secuenciales de alto nivel empleando el API secuencial, arquitecturas arbitrarias más complejas y no lineales de redes utilizando el API funcional o heredar de las subclases necesarias y reescribir todo lo demás en casos de uso muy complejos.

- *TensorFlow2*: Es un framework para machine learning de Google. Junto a Keras ofrece el backend para poder utilizar recursos hardware (GPU) en el entrenamiento de los modelos.

- *SciKit-Learn*: Es la biblioteca por excelencia para machine learning de Python. Ofrece implementaciones de los principales algoritmos tanto para aprendizaje supervisado como

no supervisado en sus distintas variantes, además de métricas, herramientas para validación y evaluación de los modelos, visualización, manipulación y transformación de los datos o persistencia entre otras características.

- *MatPlotLib*: Biblioteca con capacidades gráficas y de visualización para Python. Empleada para la generación de los distintos gráficos, visualizaciones y representaciones elaborados durante la práctica.

Además, se contó con otras herramientas software para apoyar en las labores de desarrollo:

- *GitLab*: Plataforma para la gestión del ciclo de vida de proyectos. Es un conjunto de herramientas recopiladas para la gestión y seguimiento de los proyectos, entre la que encontramos repositorios de versiones GIT.
- *Sourcetree*: Interfaz gráfica para GIT.
- *VSCode*: Entorno de desarrollo integrado. Ofrece buena integración con Python teniendo acceso a herramientas especializadas, cuadernos de código, gestión de configuraciones de ejecución, etc...
- *Overleaf*: Herramienta online para la elaboración de documentos en LaTeX.
- *Herramientas de ofimática.*

Para la elaboración del trabajo se contó tanto con una estación de trabajo local como máquinas en la nube de Google Colab. Las especificaciones de la estación local las podemos ver en la tabla 1.

Componente	Modelo
CPU	Ryzen 9 3900X 12-core, 3.8GHz, turboboost @ 4.6GHz
Memoria	56 GB DDR4 3200 CL16
GPU1	Nvdia RTX 3060 12GiB
GPU2	Nvidia GTX 1660 super 6 GiB
PSU	1200W
HDD	2TB @ 7200rpm
S.O.	Windows 10 Pro

Table 1: Especificaciones de la estación de trabajo local.

En el caso de las máquinas en la nube de Google Colab Pro, el único componente que varía entre máquina y máquina es la GPU, pudiendo ser una de las dos expuestas, en el momento de realizar el trabajo se muestran en la tabla 2.

4.1. Dataset

En este estudio se ha utilizado el conjunto de imágenes de mamografía público CDD-DDSM. Algunas imágenes de este conjunto se presentan en la figura 5. Se trata de una versión actualizada y curada de la base de datos DDSM [23] (Digital Database for Screening Mammography)

Componente	Modelo
CPU	Xeon E5-2609 V4, 2 cores*, 1.7 GHz
Memoria	25 GB
GPU1	Nvidia Testas T4 16GB
GPU2	Nvidia Tesla P100 16GB
PSU	-
HDD	130GB
S.O.	Google Colab *NIX

Table 2: Especificaciones de la estación de trabajo de Google Colab Pro.

preparada en el contexto del proyecto. Centrándonos en las imágenes con masas, se compone de 1.514 imágenes y 1.618 máscaras ambas en escala de grises y en formato DICOM [41] (Digital Imaging and COmmunications in Medicine) con tamaños de imagen comprendidos entre 1.786 x 3.920 y 5.431 x 6.931 píxeles y resoluciones entre 42 y 50 micrones. Este formato es el formato estándar empleado para comunicación y gestión de información médica de imagen e información relacionada.

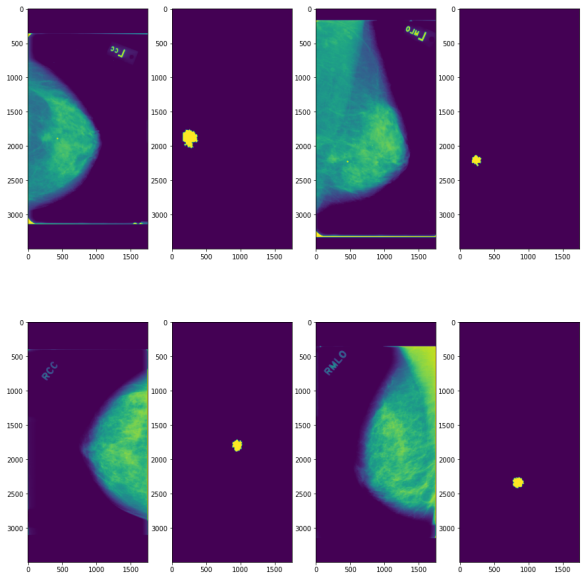

Figure 5: Cuatro imágenes de ejemplo y sus correspondientes máscaras anotadas extraídas del dataset CDD-DDSM. La fila superior se corresponde con las proyecciones cráneo caudal y medio lateral oblicua y la segunda fila lo mismo para otro paciente.

Las máscaras, anotadas por expertos, contienen la verdad base que representa la información *real y verdadera* para cada uno de los píxeles sobre la clase a la que pertenece en cada imagen, esto es, píxel clasificado como masa o píxel clasificado como no masa. Además, pueden existir varias máscaras asociadas a una mamografía, una por cada lesión presente.

4.2. Preparación de los datos

De cara a preparar un conjunto de datos confiable para el entrenamiento de los modelos de segmentación automática se realizó un análisis exploratorio de éstos para comprender mejor sus características. Tras analizar los datos, se comprobó que las imágenes que componen el

dataset original presentan diferentes resoluciones, variando el ancho entre 1.786 y 5431 píxeles y el alto entre 3.920 y 6.931 píxeles, por lo que fue necesario normalizarlas para utilizarlas en el entrenamiento de los modelos. Todas las imágenes fueron ajustadas a las mismas dimensiones de 512×1.024 píxeles. Para preservar la máxima cantidad de información y teniendo en cuenta las características del dataset, las imágenes fueron transformadas a las dimensiones utilizando una combinación de reescalado-centrado y recorte. Durante el proceso, además, se eliminaron las imágenes con más de una máscara (13 imágenes) de predicción (más de una lesión detectada como verdad base) por estar fuera del alcance de los objetivos planteados en este trabajo.

Finalmente, tras la limpieza y normalización, el dataset quedó con un total de 1.308 imágenes, con sus correspondientes máscaras, con las que se compusieron tres subconjuntos: uno para entrenamiento de los modelos con un 75% del total de imágenes, uno para validación del entrenamiento con un 10% del total y el último para prueba y selección de los modelos compuestos por el 15% del total.

Se implementó un *generador* de Tensorflow para el aumento de datos, que aplica transformaciones aleatorias de rotación (-0,1, 0,1), traslación (-0,05, 0,05), aumento (-0,1, 0,1), recortado (-0,1, 0,1) y volteo horizontal a las imágenes disponibles durante el entrenamiento de los modelos para generar transformaciones aleatorias *ilimitadas* a partir del conjunto de datos original.

4.3. Fase I: Segmentación automática

En esta fase se pretende dar una primera segmentación que pueda servir como punto de partida para la segmentación interactiva en la que a través de las interacciones del especialista se consigue refinar la segmentación de la imagen hasta llegar al resultado óptimo.

Inicialmente se consideraron dos arquitecturas CNN para el desarrollo de este trabajo, siendo la primera YOLO [49] y la segunda UNET [51]. Mientras que YOLO es un arquitectura que se ha demostrado muy eficaz para procesar vídeo y detectar y segmentar objetos en *tiempo real*, su entrenamiento es complejo y necesita de un gran volumen de datos para poder entrenarse correctamente. Por otro lado, UNET presenta una arquitectura optimizada para trabajar con datasets pequeños, los más habituales en el campo de la biomedicina, resulta más sencilla de entrenar que YOLO, obtiene buenos resultados y además dispone de diferentes variaciones y extensiones a la arquitectura básica que proponen mejorar todavía más los resultados de la segmentación. Dada la limitación de los datos disponibles, la red neuronal seleccionada en este trabajo fue UNET.

Para seleccionar un modelo para realizar la segmentación automática en la fase I, se analizaron cuatro variaciones de la arquitectura UNET: UNET, Residuals UNET [60], Attention UNET [43] y Residuals Attention UNET [7].

UNET [51] es una arquitectura de red neuronal artificial convolucional completa, FCNN por sus siglas en inglés (Fully Convolutional Neural Network) frente a la red convolucional CNN estándar. La principal diferencia entre ambas es que FCNN solo utiliza capas convolucionales: no existen capas densamente conectadas en el modelo.

En esta arquitectura se implementa una estructura similar a la de un codificador-decodificador. Originalmente construida para la segmentación de imágenes biomédicas, se caracteriza por constar de dos subredes unidas por un puente, como se muestra en la figura 6:

- *Subred de contracción*: Es la primera subred, que aplica una repetición de convoluciones para codificar la imagen en un mapa de características multinivel. Técnicamente, consiste en una repetición de 2 convoluciones 3x3 sin padding seguidas de una activación ReLU y max pooling 2x2 con stride 2. Cada paso duplica el número de canales disponibles.

- *Subred de expansión*: En esta subred se combinan la información del mapa de características aprendidas por la primera subred sobre el espacio de píxeles para obtener la segmentación. Esta subred consiste de una capa de Upsampling seguida de una convolución 2x2 (up-convolution), la concatenación con la capa correspondiente de la etapa de compactación, y 2 convoluciones 3x3 seguidas de una capa de activación ReLU. La última capa de la red es una convolución 1x1 para mapear las predicciones a la máscara de salida.

Una característica importante de UNET son los canales de copia que comunican un nivel de la red contracción con el nivel parejo de la red de expansión. Estos canales permiten a la red propagar la información desde la subred de contracción hasta la de expansión ayudando a mitigar el problema del desvanecimiento del gradiente. Esto, combinado con el aumento de datos recomendado por los autores de la arquitectura, permite entrenar la red con conjuntos pequeños de datos obteniendo un buen rendimiento.

Esta arquitectura se ha utilizado extensamente en el problema de segmentación de imágenes en el campo de la biomedicina ([51], [43], [2], [20] y [21]). Se puede encontrar más información sobre las tres variantes utilizadas (Residuals UNET, Attention UNET y Residuals Attention UNET) en los anexos *A.3 Residuals UNET* a *A.5 Residuals Attention UNET*.

Se entrenó un modelo para cada una de las cuatro arquitecturas de red neuronal propuestas, utilizando el conjunto de datos que se había preparado previamente. Para ello, se realizó un ajuste fino de los hiperparámetros para los cuatro modelos y finalmente, se seleccionó el conjunto de hiperparámetros que mejores resultados habían dado (vistos en la tabla 3) en las diferentes pruebas llevadas a cabo. El optimizador empleado durante el entrenamiento fue ADAM [24].

Opción	**Valor**
Tamaño del lote por época	128
Pasos por época	100
Épocas	150
Tamaño base del filtro	32
Tamaño del lote del generador de imágenes	1
Tasa de aprendizaje	0.0001
Tasa de descarte durante entrenamiento	0.15
Normalización batch	true
Tamaño del lote de validación	100

Table 3: Opciones de configuración para el entrenamiento de los modelos de segmentación automática.

La arquitectura que dio mejores resultados durante la selección fue Attention UNET, seguida

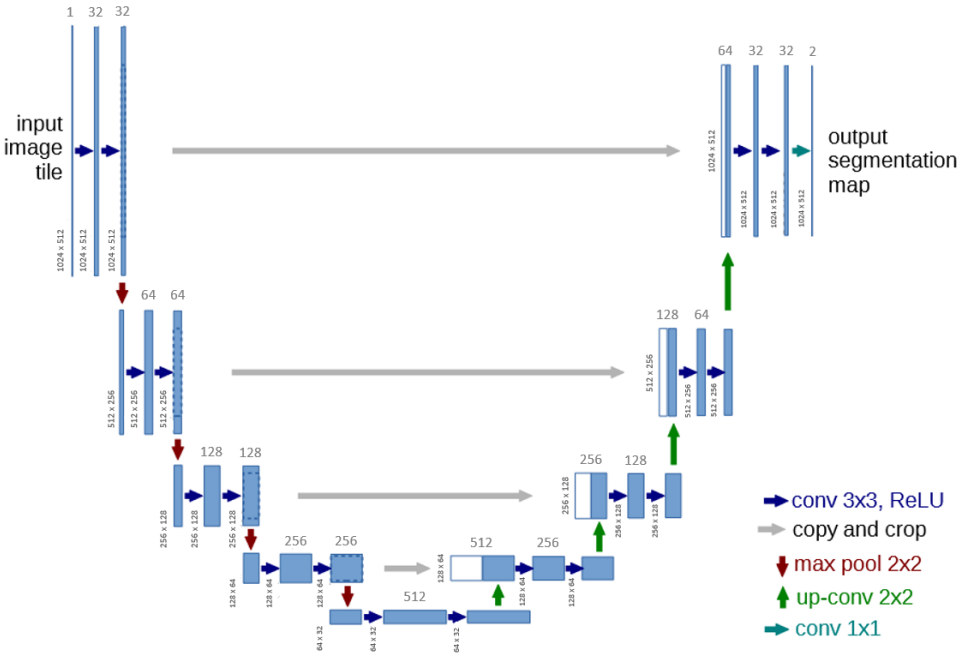

Figure 6: Configuración básica de UNET utilizada para los diferentes modelos.

de Residuals Attention UNET. Debido a que la implementación de Attention UNET es más sencilla, se utilizó este modelo para la realización de la fase I de segmentación automática.

4.4. Fase II: Segmentación interactiva

Para poder entrenar el modelo de segmentación interactiva necesitamos disponer de la imagen original (y su máscara para poder validar la segmentación), el resultado de la segmentación automática y la información de interacción del radiólogo o experto, que será tratada para presentarse como entrada al módulo de segmentación interactiva.

Debido a la carencia de un radiólogo en el equipo y a la voluntad de facilitar y automatizar el proceso de entrenamiento en la medida de lo posible, se definió el Módulo Generador de Interacciones que será el encargado de generar la información adicional necesaria para entrenar la red.

4.4.1. Módulo generador de interacciones simuladas

Este subsistema toma la máscara original de la imagen y la máscara de segmentación automática de la fase I y determina las regiones de acierto y de error de la máscara segmentada con respecto a la máscara original. A partir de estas regiones se genera información adicional, simulando las interacciones del radiólogo con la segmentación automática obtenida de la fase I, que se utilizará para entrenar el modelo de la fase II sin necesidad de intervención humana y siempre teniendo en cuenta que la interacción del radiólogo con el sistema sea la menor posible. En la figura 7 se muestra la imagen y la máscara original junto a la segmentación automática obtenida en la fase I, donde se observa como la segmentación automática requiere de correcciones relativas tanto a

los límites de la región correspondiente a la lesión como a regiones detectadas que no existen en realidad.

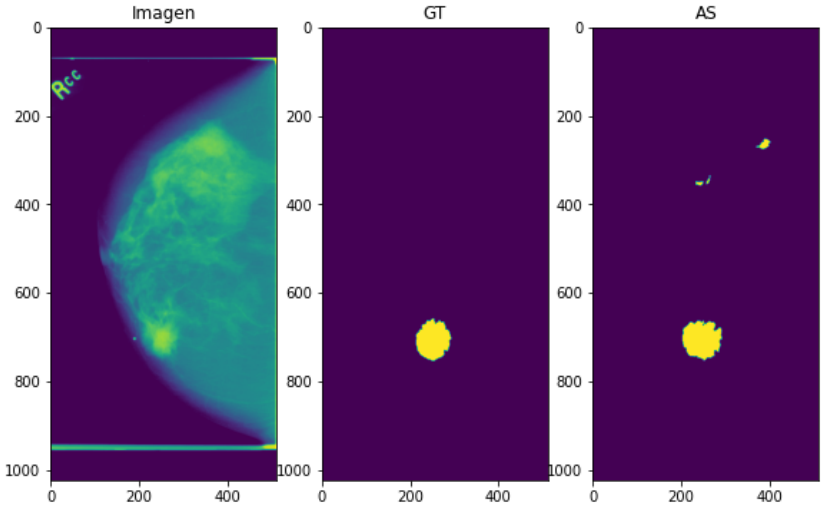

Figure 7: Imagen monocromática a segmentar (Imagen), máscara con la verdad base anotada (GT) y predicción obtenida de la segmentación automática (AS).

Por estas razones, a partir de la máscara con la verdad base y la segmentación automática que hemos realizado, validamos el resultado de dicha segmentación automática, identificando los píxel que han sido correctamente segmentados y los que no, así como el tipo de error: por exceso, se han segmentado píxeles como masa siendo no-masa (falsos positivos) o por defecto, se han segmentado píxeles como no-masa siendo masa (falsos negativos). Así podemos calcular las regiones de acierto (verdaderos positivos), y las regiones de error por exceso (falsos positivos) y por defecto (falsos negativos). La región de acierto se calcula mediante la ecuación 1.

$$\text{Región de acierto: } \begin{cases} 1, & \text{si } a_{xy} \times b_{xy} > 0 \\ 0, & \text{en otros casos} \end{cases} \tag{1}$$

$$\forall x \in [0, \text{ancho}], \forall y \in [0, \text{alto}]$$

Siendo A la máscara con la verdad base y a_{ij} un píxel cualquiera de la máscara A y B la predicción del modelo y b_{ij} un píxel cualquiera de la predicción B. La región de acierto es la intersección de la máscara A y la segmentación automática B, $R_{Ok} = A \cap B$.

De forma similar podemos definir las regiones de error por exceso (ecuación 2) y por defecto (ecuación 3).

$$\text{Región de error exceso : } \begin{cases} 1, & \text{si } b_{xy} - r_{xy} > 0 \\ 0, & \text{en otros casos} \end{cases} \tag{2}$$

$$\forall x \in [0, \text{ancho}], \forall y \in [0, \text{alto}]$$

y

$$\text{Región de error defecto} : \begin{cases} 1, & \text{si } a_{xy} - r_{xy} > 0 \\ 0, & \text{en otros casos} \end{cases} \tag{3}$$

$$\forall x \in [0, \text{ancho}], \forall y \in [0, \text{alto}]$$

Siendo R la máscara con la región de acierto y r_{xy} un píxel cualquiera de dicha máscara. Estas regiones para la imagen de la figura 7 se pueden apreciar en la figura 8.

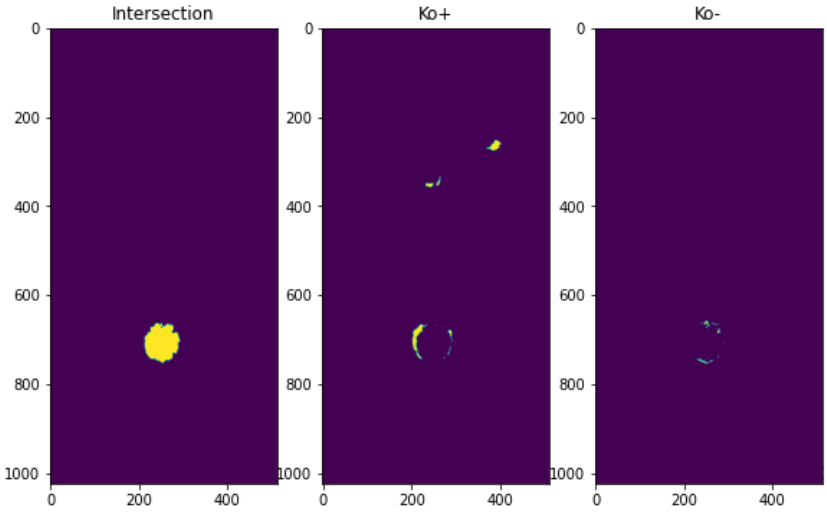

Figure 8: Región de acierto (*Intersection*), región de error por exceso (*Ko+*) y región de error por defecto (*Ko-*).

Teniendo en cuenta el alto coste en tiempo del usuario que supondría incluir manualmente la información a nivel de píxel que defina cada región y considerando la premisa de este trabajo de minimizar las interacciones necesarias del usuario para obtener la información adicional, se va a reducir a un conjunto de puntos representativos de cada una de las zonas detectadas. De esta forma, cada mancha de las regiones de acierto y de error se va a comprimir a un solo punto representativo o semilla. Nuestra estrategia utilizando un punto representativo minimiza la información adicional aportada con respecto a las *tijeras inteligentes* [39] y *cable vivo* [10] o de [34] donde la información adicional es calculada para cada píxel de la imagen.

Para calcular los puntos semilla se han definido cuatro estrategias diferentes:

- *Centro de masas*: Se toma como punto representativo el centro de masas de la mancha con una desviación de ± 10 píxeles.
- *Punto aleatorio sobre la mancha*: Un punto aleatorio de los puntos que forman la mancha.
- *Punto más lejano sobre el perímetro de la mancha*: Del perímetro de la mancha se toma el punto más lejano con respecto a la verdad base.
- *Punto aleatorio sobre la caja que circunscribe la mancha*: Un punto aleatorio de la caja que circunscribe la mancha.

4.4.2. Módulo generador de información adicional

Para poder utilizar bien los puntos seleccionados por el experto médico o bien los puntos representativos de cada mancha calculados por el módulo generador de interacciones simuladas, se calcula un mapa de probabilidad con distribución normal centrada en cada punto semilla. Dependiendo si el punto pertenece a la región de acierto o de errores se utiliza una desviación para el mapa de probabilidad, siendo $\sigma = 25$ en el caso de las regiones de acierto y $\sigma = 10$ para las regiones de error. Debido a que las imágenes presentan en media mayor número de errores que de aciertos, la intención de aumentar sigma en el caso del canal de acierto con respecto al canal de errores es aumentar la cantidad de información suministrada por el único punto que vamos a obtener del canal de aciertos, de forma que se produzca un área más grande que la que se produce en el caso de los canales de errores, donde al generarse áreas de menor tamaño se puede ser más preciso indicando el error.

Los mapas de probabilidad se utilizan para construir nuevas máscaras para la imagen, teniendo tres en total: una máscara con los aciertos o canal de aciertos, una máscara con los errores por exceso o canal de errores por exceso y una máscara con los errores por defecto o canal de errores por defecto. Podemos apreciar un ejemplo de estos canales en la figura 9. Cada estrategia de selección de los puntos semilla produce una máscara diferente.

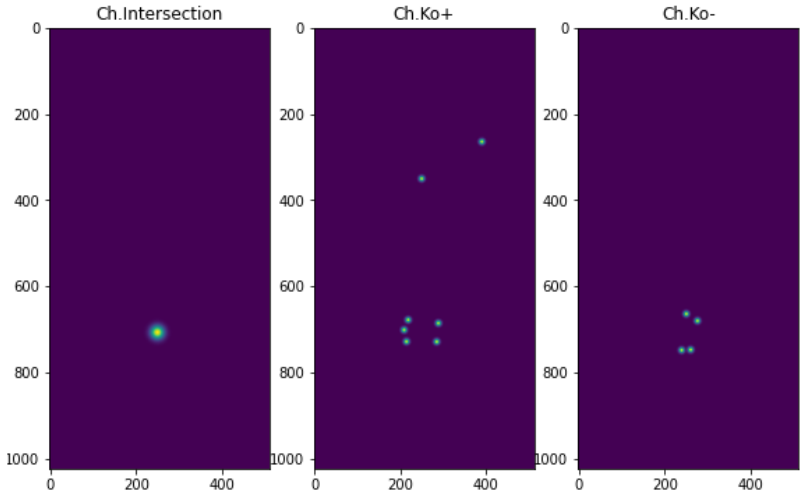

Figure 9: Los tres nuevos canales que se han generado para enriquecer la información a partir de los puntos de interés extraídos de las regiones de acierto y de error. La primera imagen se corresponde al canal de aciertos (Ch.Intersection), la segunda al canal de errores por exceso (Ch.Ko+) y por último el canal de errores por defecto (Ch.Ko-).

Se plantean experimentar con tres configuraciones de la información adicional para poder utilizar estos nuevos canales y entrenar los diferentes modelos:

- *Todos los canales*: Se suministran al modelo los canales de acierto, error por exceso y error por defecto.
- *Canal de aciertos*: Solo se suministra el canal de aciertos.
- *Canales de error*: Solo se suministran los dos canales de error.

4.4.3. Segmentación interactiva

Se van a entrenar varios modelos utilizando diferente información adicional generada a partir de cada estrategia de selección de puntos y de la configuración información de los canales. Su combinación produce doce modelos que serán entrenados y evaluados para medir su calidad.

Como se comentó en el apartado anterior, se van a entrenar modelos utilizando los diferentes conjuntos de máscaras generados por cada estrategia de selección de puntos semilla. Además, se proponen tres configuraciones diferentes de la información a utilizar para el entrenamiento de los modelos: se utilizan todos los canales adicionales, solo el canal de aciertos o solo los canales de error. La combinación de estrategias de selección de puntos semilla y configuraciones de la información produce doce combinaciones diferentes del conjunto de datos que se utilizaran para entrenar cada modelo asociado a cada una de las combinaciones.

La solución propuesta para la fase II de segmentación interactiva se basa en la arquitectura Attention UNET de forma similar a la solución para la fase I de segmentación automática. El modelo se compone de cuatro niveles: En la subred de contracción se realizan cuatro operaciones de *max-pooling* y en la subred de expansión se realizan cuatro operaciones de *upsampling*. El optimizador empleado durante el entrenamiento ha sido ADAM [24].

Para seleccionar la combinación de hiperparámetros para el entrenamiento de los modelos de la fase II, se realizó un ajuste fino de los hiperparámetros a fin de escoger la combinación que ofreciera mejores resultados durante el entrenamiento. La selección resultante se puede ver en la tabla 4. Como principales diferencias de la configuración utilizada en esta fase frente a la configuración utilizada para entrenar el modelo para segmentación automática están el tamaño del lote por época, que se ha reducido de 128 a 32 y el número de épocas que también se han reducido de 150 a 100.

Opción	**Valor**
Tamaño del lote por época	32
Pasos por época	100
Épocas	100
Tamaño base del filtro	32
Tamaño del lote del generador de imágenes	2
Tasa de aprendizaje	0.0001
Tasa de descarte durante entrenamiento	0.15
Normalización batch	true
Tamaño del lote de validación	100

Table 4: Opciones de configuración para el entrenamiento de los modelos de segmentación interactiva.

La razón de reducir el tamaño del lote por época es el notable aumento de tamaño experimentado en la información utilizada para el entrenamiento y la validación, donde las imágenes pasan de un canal a cinco, multiplicando los requisitos de memoria necesaria para poder gestionar el dataset. Además, al disponer de más información por imagen (imagen original, segmentación automática y la información adicional) durante esta fase respecto a la fase I, se han podido acortar la épocas del entrenamiento ya que los modelos han sido capaces de alcanzar la puntuación máxima más rápido que durante la fase I.

5. Resultados

Para el entrenamiento de los modelos de ambas fase se realizó optimización de hiperparámetros, probando separadamente para cada fase el tamaño de las épocas de entrenamiento en el rango $[64, 350]$, pasos por época $[50, 250]$, tamaño del lote $[16, 192]$, tamaño base del filtro $[8, 128]$, tasa de aprendizaje $[10e^{-3}, 10e^{-7}]$, tasa de descarte durante el entrenamiento $[0, 0.5]$ (drop) y tamaño del kernel de los filtros de convolución en el rango $[2, 10]$. Para cada fase se buscó y utilizó la selección de hiperparámetros que mejores resultados produjeron. Se ha podido comprobar como ante determinadas combinaciones de la tasa de aprendizaje y el tamaño del kernel, el aprendizaje del modelo se estanca de forma similar a como ocurría en [45], teniendo que en ambas fases, los mejores resultados se obtenían con tamaños de kernel en $[3, 5]$ y tasas de aprendizaje en $[10e^{-4}, 5 \times 10e^{-6}]$, requiriendo de largas sesiones de entrenamiento para poder *aprender*.

Durante el entrenamiento de los modelos, para ambas fases se utilizaron ¡¡1−*índice de semejanza de Dice*¿¿ [8] como función de pérdida y el ¡¡*índice de Jaccard*¿¿ [19] como métrica.

5.1. Fase I - Segmentación automática

En esta fase se probaron cuatro variaciones de la arquitectura UNET a fin de seleccionar una arquitectura de referencia en la que basar los modelos que se construyesen tanto para segmentación automática como segmentación interactiva. Las puntuaciones de los modelos entrenados se pueden apreciar en la tabla 5. Attention UNET dio los mejores resultados en las diferentes sesiones de entrenamiento que se realizaron, por lo que resultó elegida como arquitectura de referencia para construir el modelo de segmentación automática y el modelo de segmentación interactiva.

	Entrenamiento		**Validación**	
Arquitectura	**Dice**	**Jaccard**	**Dice**	**Jaccard**
UNET	0.3822	0.1152	0.4197	0.1364
Residuals	0.4204	0.1278	0.4342	0.1362
Attention	0.4484	0.1624	0.4337	0.1612
ResAtt	0.3734	0.1580	0.3756	0.1523

Table 5: Evaluación de los modelos de segmentación automática.

Se puede comprobar, comparando las puntuaciones para entrenamiento y validación, que los modelos están entrenados correctamente, mostrando una capacidad de generalización similar a la obtenida durante el entrenamiento. De igual forma, ninguno de los modelos consigue superar la puntuación Dice de 0.45 tras un entrenamiento largo en el que al final el aprendizaje se estanca, no siendo el modelo capaz de aprender más, resaltando la complejidad de la tarea de segmentación para éste tipo de imágenes.

5.2. Fase II - Segmentación interactiva

Tal y como se comentó en la sección anterior, se dispone de un total de 12 modelos cuya variación se debe la información adicional utilizada para el entrenamiento de cada uno. Debido al problema del balanceo de clases habitual en los trabajos de segmentación de imágenes, se ha

optado por utilizar para el entrenamiento de los modelos la función de perdida '1 - Índice de Dice' [31] [24] mientras que como métricas se ha utilizado el índice de Jaccard [19].

Para evaluar estos modelos se utilizó un subconjunto de prueba compuesto por 200 imágenes reservadas previamente al entrenamiento de los modelos de la fase II. Para cada imagen segmentada se calcularon cinco métricas: Índice de semejanza Dice, distancia de Hausdorff modificada al percentil 95, diferencia media de áreas, sensibilidad y puntuación F1. Se puede encontrar más información sobre estas métricas en el anexo *C. Evaluación*.

En la tabla 6 se muestran los resultados medios obtenidos. La columna *Selección* indica la estrategia de selección de puntos semilla utilizada, mientras que *Configuración* indica la configuración de información adicional utilizada. Todas las puntuaciones se han escalado de tal forma que cuanto más próximas estén a 0, mejor calidad del resultado.

Selección	Configuración	Dice	Hausdorff	AAD	Sensitivity	F1	Puntuación
Centro de masas	Pos&Neg	0.2130	42.86	0.2821	0.1945	0.2090	0.1868
	Pos	0.4250	287.72	0.5375	0.3802	0.2569	0.3679
	Neg	0.2064	47.09	0.2927	0.1906	0.1984	0.1855
Aleatorio sobre la mancha	Pos&Neg	0.2804	45.73	0.3214	0.2715	0.2618	0.2346
	Pos	0.4261	288.09	0.5278	0.3865	0.2584	0.3678
	Neg	0.3246	60.08	0.4129	0.2983	0.3142	0.2800
Sobre la caja	Pos&Neg	0.2752	58.59	0.3036	0.2715	0.2565	0.2311
	Pos	0.4344	295.16	0.4909	0.4228	0.2644	0.3717
	Neg	0.2944	77.85	0.3904	0.2725	0.2725	0.2590
Borde más lejano	Pos&Neg	0.3789	206.69	0.4298	0.3494	0.2510	0.3163
	Pos	0.4199	284.27	0.5178	0.3805	0.2505	0.3611
	Neg	0.4331	104.29	0.5964	0.3876	0.3590	0.3726

Table 6: Evaluación de los modelos de segmentación interactiva, donde se muestra la estrategia de *Selección* de puntos semilla, la *Configuración* de los canales adicionales suministrados para el entrenamiento y las puntuaciones media de cada combinación de *Selección* y *Configuración* para las métricas: Índice de semejanza *Dice*, Distancia de *Hausdorff*, Diferencia Media de Áreas (*AAD*), Sensibilidad (*Sensitivity*) y puntuación *F1*. La última columna contiene la media de las puntuaciones del modelo.

Durante el entrenamiento y evaluación de los modelos se han apreciado varias consideraciones. Los modelos entrenados utilizando como información adicional solo los canales de error han conseguido, por lo general, mejores puntuaciones en todas las métricas que los modelos entrenados solo con el canal de aciertos. De igual forma entrenar utilizando el canal de aciertos y los canales de errores consigue una leve mejora con respecto a los modelos entrenados con solo los canales de errores. De todas las variaciones consideradas, centro de masas utilizando el canal de aciertos y los canales de error o solo los canales de error, han conseguido las mejores puntuaciones en todas las métricas, situándose el rendimiento de ambos modelos a la par.

5.3. Prueba de concepto

Finalmente, se evaluaron los cuatro modelos entrenados utilizando todos los canales de información adicional (canal de aciertos y canales de errores) disponibles para cada una de las cuatro estrategias de selección de puntos, implementando en una prueba de concepto las dos

fases definidas para el proceso, en el que el usuario interactúa con la imagen y ésta se corrige a través de las diferentes iteraciones. Para esta prueba se utilizaron el modelo para segmentación automática entrenado durante la fase I y los modelos que habían sido entrenados utilizando como información adicional tanto el canal de aciertos como los de error durante la fase II, de forma, que en la PoC se pudiesen implementar diferentes tipos de interacción con el ratón para representar cada tipo de información que puede utilizar el modelo:

- *Información de aciertos*: Donde está bien segmentada la imagen.
- *Información de errores por exceso*: con información sobre regiones que contienen falsos positivos.
- *Información de errores por defecto*: con información sobre regiones que contienen falsos negativos.

Para este experimento se seleccionaron 10 imágenes del conjunto de prueba reservado y cada imagen fue segmentada iterativamente tres veces, partiendo la primera iteración de la segmentación automática realizada para la imagen y las sucesivas del estado de la segmentación de la iteración anterior y la información adicional proporcionada por el usuario (un solo click por iteración). Al final de cada iteración, para cada imagen, se tomaron lecturas de las métricas utilizadas para la evaluación de los modelos y se anotaron para poder calcular su media y desviación.

Los resultados se muestran en la tabla 7. La columna puntuación es el resultado de la media de las cinco métricas utilizadas previamente en la evaluación mostrada en la tabla 6. Durante esta prueba se pudieron igualar o superar las puntuaciones obtenidas por los modelos durante la evaluación. El modelo basado en la estrategia de selección de centros de masas es el que más rápido ha superado la puntuación de la evaluación, seguido de la estrategia del punto más lejano sobre el borde. Las estrategia de un punto aleatorio sobre la mancha y un punto sobre la caja de circunscripción también han superado la puntuación de evaluación pero han requerido más interacciones del usuario para conseguirlo.

Modelo	**Interacciones**	**Puntuación**
Centro de masas	1	0.1984 ± 0.07
	2	0.1263 ± 0.04
	3	0.0992 ± 0.01
Aleatorio mancha	1	0.77 ± 1.14
	2	0.296 ± 0.17
	3	0.2272 ± 0.1
Sobre la caja	1	0.5281 ± 0.99
	2	0.2305 ± 0.2
	3	0.178 ± 0.11
Borde más lejano	1	0.2393 ± 0.36
	2	0.1561 ± 0.04
	3	0.1367 ± 0.02

Table 7: Resultados de la PoC sorbe las 10 imágenes seleccionadas para la prueba tras 1, 2 y 3 interacciones del usuario con la imagen y sus correspondientes correcciones iterativas.

6. Discusión y trabajos futuros

Todavía existen múltiples opciones a explorar. La primera es la solidez del entrenamiento resultante de utilizar una combinación de las diferentes estrategias de selección propuestas en este trabajo. La intención detrás de esta propuesta es conseguir una mayor robustez del modelo entrenado a diferentes tipos de clicks de los usuarios finales.

Entrenamiento por etapas. El conjunto de datos ha resultado desafiante, primero debido a la incapacidad de diagnosticar las imágenes disponibles al no disponer de un especialista en diagnóstico por imagen y segundo a la inestabilidad del proceso de entrenamiento junto con al estancamiento del aprendizaje que se produce eventualmente. Una posible solución podría ser entrenar el modelo en diferentes iteraciones, empezando con tasas de aprendizaje más altas y disminuyéndola progresivamente en sucesivas iteraciones a fin de afinar los resultados. Otra aproximación es entrenar entregando los puntos de interés con los que se enriquece la segmentación interactiva de uno en uno.

Procesamiento de las proyecciones Cráneo-Caudal y Medio Lateral Oblicua simultáneamente durante el proceso de segmentación, de forma que se disponga de más información sobre las posibles lesiones existentes y el modelo sea capaz de segmentarlo más precisamente.

De igual forma, hay espacio para mejorar las métricas utilizadas para el entrenamiento, ya que se han demostrado vulnerables a las condiciones habituales de este tipo de imágenes: ruido, figuras a segmentar altamente irregular o falta de definición en determinadas zonas muy densas.

7. Conclusiones

En este trabajo se ha realizado el análisis de las bases de datos públicas disponibles para seleccionar la más adecuada a nuestros objetivos. La base de datos escogida fue CDD-DDSM, la cual se preparó y procesó de cara a utilizarla en el trabajo.

Se estudió y analizó la evolución de las diferentes propuestas y aproximaciones para la segmentación de imágenes asistida por computador, teniendo en consideración desde las técnicas clásicas hasta el estado del arte para este tipo de labores, donde se emplean herramientas más sofisticadas como los modelos de machine learning o de deep learning.

A continuación, se procedió a analizar y evaluar las diferentes arquitecturas de red neuronal profunda convolucional completa disponibles y sus variantes para seleccionar una con la que se entrenó un modelo para segmentar las imágenes. Las arquitecturas consideradas son YOLO y UNET, de las cuales se escogió UNET por presentar diferentes ventajas sobre YOLO, además de múltiples variantes sobre la versión básica de la arquitectura, que se podían utilizar en los experimentos. Se entrenaron modelos para cada variante de la arquitectura y se escogió la que obtuvo las mejores puntuaciones Dice y Jaccard para el dataset disponible. Sin embargo, a pesar de la cantidad de entrenamiento o del ajuste de parámetros con que se probara a entrenar los distintos modelos, no se consiguió superar la puntuación Dice de 0.45 con ninguno de ellos.

De cara a la segmentación interactiva, se propone una herramienta para generar información adicional (el módulo generador de información adicional) a partir de las interacciones del experto con una imagen segmentada previamente, que se puede utilizar para entrenar el modelo de segmentación interactiva y para preparar las interacciones del experto con la imagen segmentada antes de una nueva iteración de segmentación en un *entorno real* de explotación.

No solo se ha propuesto el módulo generador de información adicional, si no que también se propone una aproximación para automatizar la extracción de información de la segmentación automática (módulo generador de interacciones simuladas) original sin necesidad de un experto para realizar las correcciones necesarias. En esta propuesta, se consideran cuatro estrategias para la selección de la información, que se han estudiado y comparado a través de los distintos modelos construidos y las validaciones llevadas a cabo.

Disponiendo de estos dos módulos y el modelo para realizar la segmentación automática, se pueden entrenar modelos para realizar la segmentación interactiva y evaluarlos. En esta etapa, se han propuesto diferentes tipos de configuración de la información adicional (utilizando toda la información disponible, solo el canal de aciertos o solo los canales de error) para las que se han construido modelos en combinación con las estrategias de selección de puntos propuestas en el módulo generador de interacciones simuladas, lo que nos proporciona una base de doce combinaciones de estrategia y configuración con las que entrenar modelos para evaluar cada propuesta.

Durante la evaluación se ha observado que los modelos entrenados utilizando las configuraciones "toda la información" y los entrenados utilizando solo los "canales de error" tienen un rendimiento similar y superior a aquellos entrenados solo con el canal de aciertos como información adicional. De igual forma, la estrategia de selección "centro de masas" ha presentado los mejores puntuaciones consistentemente en las diferentes pruebas realizadas. Para esta estrategia se han conseguido puntuaciones Dice próximas al 0.9, lo cual es una mejora notable frente al 0.45 obtenido con los modelos entrenados para segmentación automática.

Finalmente se ha implementado una prueba de concepto del proceso en la que se han evaluado utilizando las métricas de evaluación descritas en el apartado *5.2. Fase II - Segmentación interactiva*. Para esta prueba se han evaluado las mismas 10 imágenes en los cuatro modelos entrenados para la opción de configuración de dos canales y evaluando los resultados de la segmentación interactiva de cada imagen después de un click (una interacción) del usuario en tres iteraciones consecutivas con la imagen.

Las puntuaciones medias de los modelos son similares a las conseguidas durante el entrenamiento y la evaluación salvo por el modelo de selección de punto aleatorio sobre la mancha. Con las tres iteraciones se han conseguido puntuaciones Dice en el modelo entrenado para la estrategia "centro de masas" de ~ 0.93, lo cual valida la aplicabilidad técnica del proceso completo propuesto en este trabajo a un entorno real con las debida depuración y mejora del código de cara a su productivización.

Bibliografía

[1] Dina Abdelhafiz et al. "Convolutional neural network for automated mass segmentation in mammography". In: *BMC Bioinformatics* 21.1 (Dec. 2020), p. 192. ISSN: 1471-2105. DOI: 10.1186/s12859-020-3521-y. URL: https://doi.org/10.1186/s12859-020-3521-y.

[2] Md. Zahangir Alom et al. "Breast Cancer Classification from Histopathological Images with Inception Recurrent Residual Convolutional Neural Network". In: (Nov. 2018).

[3] J. C. Bezdek, L. O. Hall, and L. P. Clarke. "Review of MR image segmentation techniques using pattern recognition". In: *Med Phys* 20.4 (1993), pp. 1033–1048.

[4] A.C. Evans C.A. Cocosco A.P. Zijdenbos. "A fully automatic and robust brain MRI tissue classification method". In: *Med. Image Anal.* 7 (2003), pp. 513–527. URL: https://math.berkeley.edu/~sethian/2006/Publications/Book/2006/book_1999.html.

[5] Kyunghyun Cho et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014).

[6] G.B. Coleman and H.C. Andrews. "Image segmentation by clustering". In: *Proceedings of the IEEE* 67.5 (1979), pp. 773–785. DOI: 10.1109/PROC.1979.11327.

[7] DICOM Commitee. "DICOM Homepage". In: ().

[8] Lee R. Dice. "Measures of the Amount of Ecologic Association Between Species". In: *Ecology* 26.3 (1945), pp. 297–302. DOI: https://doi.org/10.2307/1932409. eprint: https://esajournals.onlinelibrary.wiley.com/doi/pdf/10.2307/1932409. URL: https://esajournals.onlinelibrary.wiley.com/doi/abs/10.2307/1932409.

[9] Joseph C. Dunn. "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters". In: 1973.

[10] Alexandre Xavier Falcão et al. "User-Steered Image Segmentation Paradigms: Live Wire and Live Lane". In: *Graph. Model. Image Process.* 60 (1998), pp. 233–260.

[11] Kunihiko Fukushima. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological Cybernetics* 36.4 (Apr. 1980), pp. 193–202. ISSN: 1432-0770. DOI: 10.1007/BF00344251. URL: https://doi.org/10.1007/BF00344251.

[12] P. Gibbs et al. "Tumour volume determination from MR images by morphological segmentation". In: *Phys Med Biol* 41.11 (Nov. 1996), pp. 2437–2446.

[13] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: 7 (Dec. 2015).

[14] Mohammad Hesam Hesamian et al. "Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges". In: *Journal of Digital Imaging* 32.4 (Aug. 2019), pp. 582–596. ISSN: 1618-727X. DOI: 10.1007/s10278-019-00227-x. URL: https://doi.org/10.1007/s10278-019-00227-x.

[15] Sepp Hochreiter. "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6 (Apr. 1998), pp. 107–116. DOI: 10.1142/S0218488598000094.

[16] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-term Memory". In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[17] D H Hubel and T N Wiesel. "Receptive fields and functional architecture of monkey striate cortex". en. In: *J Physiol* 195.1 (Mar. 1968), pp. 215–243.

[18] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.

[19] Paul Jaccard. "THE DISTRIBUTION OF THE FLORA IN THE ALPINE ZONE.1". In: *New Phytologist* 11.2 (1912), pp. 37–50. DOI: https://doi.org/10.1111/j.1469-8137.1912.tb05611.x. eprint: https://nph.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-8137.1912.tb05611.x. URL: https://nph.onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8137.1912.tb05611.x.

[20] Mina Jafari et al. "DRU-net: An Efficient Deep Convolutional Neural Network for Medical Image Segmentation". In: (Apr. 2020).

[21] Debesh Jha et al. "DoubleU-Net: A Deep Convolutional Neural Network for Medical Image Segmentation". In: July 2020, pp. 558–564. DOI: 10.1109/CBMS49503.2020.00111.

[22] Wen-Xiong Kang, Qing-Qiang Yang, and Run-Peng Liang. "The Comparative Research on Image Segmentation Algorithms". In: *2009 First International Workshop on Education Technology and Computer Science*. Vol. 2. 2009, pp. 703–707. DOI: 10.1109/ETCS.2009.417.

[23] Rana Khaled et al. "Categorized contrast enhanced mammography dataset for diagnostic and artificial intelligence research". In: *Scientific Data* 9.1 (Mar. 2022), p. 122. ISSN: 2052-4463. DOI: 10.1038/s41597-022-01238-0. URL: https://doi.org/10.1038/s41597-022-01238-0.

[24] Diederik Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations* (Dec. 2014).

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[26] Y. LeCun et al. "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4 (Dec. 1989), pp. 541–551. ISSN: 0899-7667. DOI: 10.1162/neco.1989.1.4.541. eprint: https://direct.mit.edu/neco/article-pdf/1/4/541/811941/neco.1989.1.4.541.pdf. URL: https://doi.org/10.1162/neco.1989.1.4.541.

[27] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[28] Yann LeCun. "Generalization and network design strategies". In: 1989.

[29] Chulhee Lee et al. "Unsupervised connectivity-based thresholding segmentation of midsagittal brain MR images". English. In: *Computers in Biology and Medicine* 28.3 (May 1998), pp. 309–338. ISSN: 0010-4825. DOI: 10.1016/S0010-4825(98)00013-4.

[30] H.D. Li et al. "Markov random field for tumor detection in digital mammography". In: *IEEE Transactions on Medical Imaging* 14.3 (1995), pp. 565–576. DOI: 10.1109/42.414622.

[31] Hongwei Li et al. “Fully Convolutional Network Ensembles for White Matter Hyperintensities Segmentation in MR Images”. In: *NeuroImage* 183 (Feb. 2018). DOI: 10.1016/j.neuroimage.2018.07.005.

[32] Geert Litjens et al. “A survey on deep learning in medical image analysis”. In: *Medical Image Analysis* 42 (2017), pp. 60–88. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2017.07.005. URL: https://www.sciencedirect.com/science/article/pii/S1361841517301135.

[33] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965.

[34] Zhengdong Lu and Miguel A. Carreira-Perpinan. “Constrained spectral clustering through affinity propagation”. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587451.

[35] I. N. Manousakas et al. “Split-and-merge segmentation of magnetic resonance medical images: performance evaluation and extension to three dimensions”. In: *Comput Biomed Res* 31.6 (Dec. 1998), pp. 393–412.

[36] O. Matan et al. “Handwritten character recognition using neural network architectures”. In: 1990.

[37] Tim McInerney and Demetri Terzopoulos. “Deformable models in medical image analysis: a survey”. In: *Medical Image Analysis* 1.2 (1996), pp. 91–108. ISSN: 1361-8415. DOI: https://doi.org/10.1016/S1361-8415(96)80007-7. URL: https://www.sciencedirect.com/science/article/pii/S1361841596800077.

[38] E. Michael et al. “Breast Cancer Segmentation Methods: Current Status and Future Potentials”. In: *Biomed Res Int* 2021 (2021), p. 9962109.

[39] Eric Mortensen. “Interactive Segmentation with Intelligent Scissors”. In: *Graphical Models and Image Processing* 60 (Sept. 1998), pp. 349–384. DOI: 10.1006/gmip.1998.0480.

[40] Abdullah-Al Nahid and Yinan Kong. “Involvement of Machine Learning for Breast Cancer Image Classification: A Survey”. In: *Computational and Mathematical Methods in Medicine* 2017 (Dec. 2017), p. 3781951. ISSN: 1748-670X. DOI: 10.1155/2017/3781951. URL: https://doi.org/10.1155/2017/3781951.

[41] Zhen-Liang Ni et al. “RAUNet: Residual Attention U-Net for Semantic Segmentation of Cataract Surgical Instruments”. In: Sept. 2019.

[42] D. Nie et al. “FULLY CONVOLUTIONAL NETWORKS FOR MULTI-MODALITY ISOINTENSE INFANT BRAIN IMAGE SEGMENTATION”. In: *Proc IEEE Int Symp Biomed Imaging* 2016 (2016), pp. 1342–1345.

[43] Ozan Oktay et al. “Attention U-Net: Learning Where to Look for the Pancreas”. In: (Apr. 2018).

[44] World Health Organization. “Cáncer de mama”. In: (2021). https://www.who.int/es/news-room/fact-sheets/detail/breast-cancer.

[45] Mariano Rincón-Zamorano Pablo Duque Asens and Jose Manuel Cuadra. “Data Preprocessing for Automatic WMH Segmentation with FCNNs”. In: (2019).

[46] Dinesh D Patil and Sonal G Deore. "Medical image segmentation: a review". In: *International Journal of Computer Science and Mobile Computing* 2.1 (2013), pp. 22–27.

[47] Dzung Pham, Chenyang Xu, and Jerry Prince. "A Survey of Current Methods in Medical Image Segmentation". In: *Annual review of biomedical engineering* 2 (Feb. 2000), pp. 315–37. DOI: 10.1146/annurev.bioeng.2.1.315.

[48] S. Pohlman et al. "Quantitative classification of breast tumors in digitized mammograms". In: *Med Phys* 23.8 (Aug. 1996), pp. 1337–1345.

[49] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: June 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

[50] M Rincón et al. "Improved Automatic Segmentation of White Matter Hyperintensities in "MRI" Based on Multilevel Lesion Features". en. In: *Neuroinformatics* 15.3 (July 2017), pp. 231–245.

[51] 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*. Ed. by Nassir Navab et al. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4.

[52] J.A. Sethian. "Deformable models in medical image analysis: a survey". In: *Cambridge University Press* (1999). URL: https://math.berkeley.edu/~sethian/2006/Publications/Book/2006/book_1999.html.

[53] N. Sharma and L. M. Aggarwal. "Automated medical image segmentation techniques". In: *J Med Phys* 35.1 (Jan. 2010), pp. 3–14.

[54] E. Song et al. "Breast mass segmentation in mammography using plane fitting and dynamic programming". In: *Acad Radiol* 16.7 (July 2009), pp. 826–835.

[55] Jost Tobias Springenberg et al. "Striving for simplicity: The all convolutional net". In: *arXiv preprint arXiv:1412.6806* (2014).

[56] J. Tang et al. "Computer-aided detection and diagnosis of breast cancer with mammography: recent advances". In: *IEEE Trans Inf Technol Biomed* 13.2 (Mar. 2009), pp. 236–251.

[57] Daniel J Withey and Zoltan J Koles. "Medical image segmentation: Methods and software". In: *2007 Joint Meeting of the 6th International Symposium on Noninvasive Functional Source Imaging of the Brain and Heart and the International Conference on Functional Biomedical Imaging*. IEEE. 2007, pp. 140–143.

[58] Kouichi Yamaguchi et al. "A neural network for speaker-independent isolated word recognition". In: *ICSLP*. 1990.

[59] W. Zhang et al. "Deep convolutional neural networks for multi-modality isointense infant brain image segmentation". In: *Neuroimage* 108 (Mar. 2015), pp. 214–224.

[60] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. "Road Extraction by Deep Residual U-Net". In: *IEEE Geoscience and Remote Sensing Letters* 15.5 (2018), pp. 749–753. DOI: 10.1109/LGRS.2018.2802944.

[61] Juan Zhou and Jagath C. Rajapakse. “Segmentation of subcortical brain structures using fuzzy templates”. In: *NeuroImage* 28.4 (2005). Special Section: Social Cognitive Neuroscience, pp. 915–924. ISSN: 1053-8119. DOI: https://doi.org/10.1016/j.neuroimage.2005.06.037. URL: https://www.sciencedirect.com/science/article/pii/S1053811905004660.

Apéndices

A. CNN

A.1 Historia

Las redes neuronales convolucionales (CNN) son redes neuronales artificiales que se basan en la operación de convolución en lugar de la multiplicación de matrices. Son utilizadas para tratar datos con topología de malla. Estas redes están inspiradas en los procesos biológicos del córtex visual del cerebro.

Entre los año 1958 y 59 el equipo de David H. Hubel y Torsten Wiesel realizaron una serie de experimentos en gatos y más tarde en monos [17], revelando información crucial sobre el funcionamiento del córtex visual. Este descubrimiento les concedió el premio Nobel en Fisiología o Medicina en 1981. Entre los descubrimientos realizados notaron que muchas neuronas del córtex visual disponían de un pequeño *campo receptor local*, son neuronas que solo reaccionan ante el estimulo en una región limitada del campo de visión. Los campos receptores de múltiples neuronas pueden sobreponerse, componiendo entre todas las neuronas la representación del campo visual. Los autores llegaron a definir el comportamiento de diferentes tipos de neuronas especializadas, como por ejemplo aquellas que solo reaccionaban ante líneas horizontales, mientras que otras solo reaccionaban ante determinadas otras direcciones. De igual forma existen otras neuronas que reaccionaban a patrones más complejos combinación de las neuronas de los niveles inferiores.

El estudio de este nuevo conocimiento sobre el córtex visual llevó al equipo de Kunihiko Fukushima en 1980 a proponer el Neocognitron [11] que introducía los dos tipos básicos de capa utilizados para construir las CNN: *convolución* y *downsampling*. El Neocognitron se considera la primera CNN.

En 1990 el equipo de Kouichi Yamaguchi [58] propusieron un nuevo tipo de capa para las CNNs, *pooling layer*, que aplica un filtro de tamaño fijo que extrae el valor máximo de la imagen en esa región.

En 1998 Yann LeCun y su equipo, tras una década de investigación en la que propusieron la versión original de LeNet [26], probaron que reducir el número de parámetros libres a entrenar aumenta la capacidad de generalización del modelo [28] y la utilización de Backpropagation para el entrenamiento de los modelos [36], en 1998 propusieron LeNet-5 [27], una CNN capaz de reconocer dígitos escritos a mano a una resolución de imagen de 32×32 píxel.

Durante el Imagenet Large Scale Visual Recognition Challenge de 2012 se produjo la victoria de AlexNet [25]. En este trabajo como principales novedades se presentan una arquitectura de red neuronal artificial convolucional capaz de reconocer entre las 1.000 clases representadas en el dataset con un error del 15.3% en el *top-5*, utilizando imágenes hasta 8 veces más grandes que las originales utilizadas para LeNet y la capacidad de entrenar el modelo utilizando GPUs de propósito general, lo cual reduce considerablemente los tiempos de entrenamiento.

A.2 Red Neuronal Convolucional

El bloque de construcción más importante de una CNN es la capa de convolución. Inspiradas en los procesos neurológicos involucrados en la visión natural descritos por Hubel y Wiesel, en estas capas las neuronas se conectan solo a los píxel de la imagen original que están en su *campo de recepción*. Las siguientes capas de neuronas se conectan solo a las neuronas de su *campo de recepción* del nivel anterior.

La operación de convolución se puede aplicar a imágenes de una o más dimensiones, pudiéndose expresar en el caso de dos dimensiones como:

$$y_{i,j} = \sum_{h=-\infty}^{\infty} \sum_{w=-\infty}^{\infty} K_{h,w} I_{i-h,j-w}$$

Siendo $y_{i,j}$ el píxel de la posición i, j en la imagen convolucionada, K el filtro de convolución e I la imagen original.

Las capas de convolución aplican diferentes filtros sobre la imagen, de forma que cada filtro se va desplazando como una *ventana* sobre la imagen, extrayendo información de ésta y reduciendo la dimensionalidad como se aprecia en la figura 10.

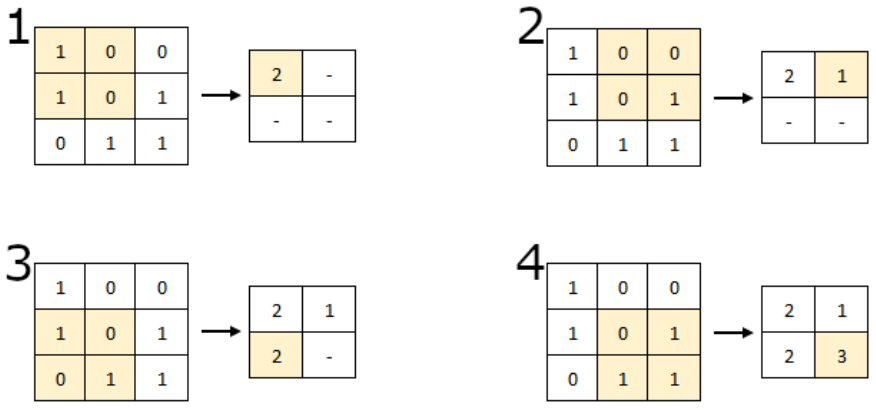

Figure 10: Aplicación de la convolución a una imagen de entrada utilizando el filtro $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$.

Existen dos parámetros que permiten controlar el comportamiento de la convolución sobre la imagen: *padding* y *stride*. El primer parámetro, *padding*, se utiliza para controlar el tamaño

de la imagen convolucionada, pudiéndose forzar que se respeten las dimensiones de la imagen original o que produzca una imagen convolucionada de tamaño *válido* menor que la original. El parámetro *stride* permite controlar el tamaño del desplazamiento de la ventana que aplica el filtro de convolución a la imagen original. Este parámetro suele estar inicializado a 1.

El filtro es una matriz de tamaño (k_1, k_2) que se utiliza aplicando la multiplicación binaria de matrices entre la región de la imagen sobre la que se encuentra la *ventana* y el filtro de convolución y sumando el valor de los productos. Dependiendo del filtro que se aplique se extrae determinada información como se muestra en la figura 11, donde se aplican tres filtros a la misma imagen para extraer diferente información de esta.

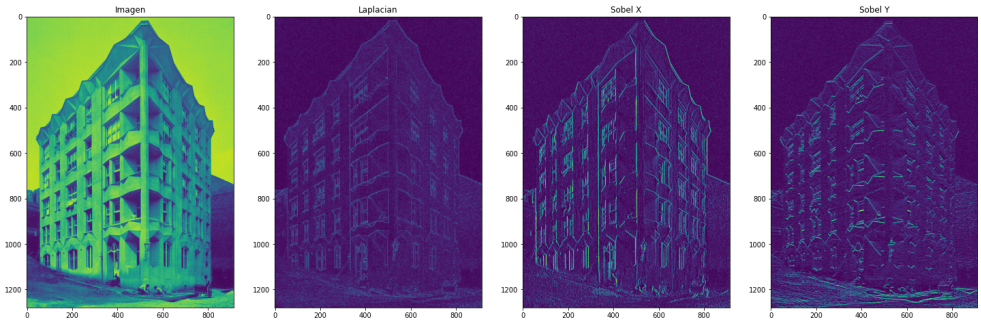

Figure 11: Imagen original y tres filtros aplicados a ella: Laplace, Sobel X y Sobel Y.

Otros elementos comunes utilizados junto a la capa de convolución son la capa de *pooling* o la de normalización. La capa de *pooling* reemplaza una región por un estadístico representativo de esta, por ejemplo el valor máximo o la media aritmética como en la figura 12, reduciendo la complejidad computacional de la imagen. Como alternativa a esta capa es posible utilizar una capa de convolución con un *stride* mayor al del resto de capas [55].

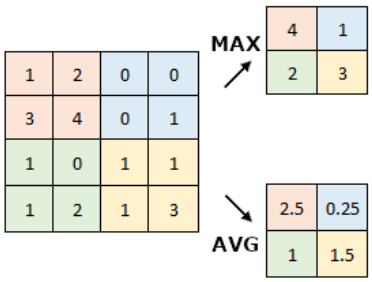

Figure 12: Reducción de dimensionalidad mediante *max pooling* de 2×2 sobre una imagen de 4×4 píxel, aplicando el estadístico *valor máximo*.

La capa de normalización *batch* aplica una normalización a la salida de determinadas capas ocultas de la red de forma que el valor de activación se mantenga próximo a 0 y su desviación estándar próxima a 1. Esta mejora acelera el entrenamiento de los modelos, permitiendo tasas de aprendizaje más altas [18].

A.3 Residuals UNET

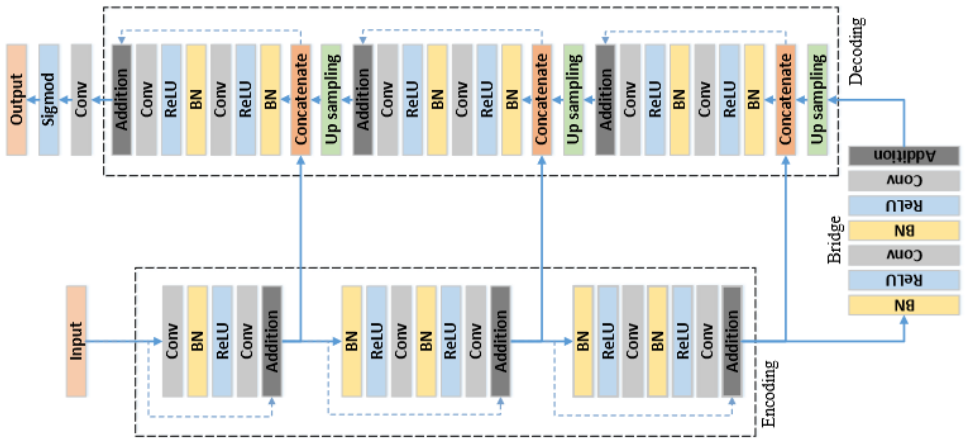

Figure 13: Residuals UNET.

En este trabajo, los autores [60] proponen una modificación basada en los trabajos de He et al [13], sobre redes neuronales residuales con la intención de mejorar todavía más los resultados conseguidos por la arquitectura UNET. La modificación, apreciable en la figura 13, consiste en una serie de unidades residuales apiladas. Cada unidad residual se puede entender como una forma general de la ecuación 4.

$$\begin{aligned} y_l &= h(x_l) + F(x_l, W_l), \\ x_{l+1} &= f(y_l) \end{aligned} \tag{4}$$

Estas redes (Residuals-*) implementan accesos directos entre capas no contiguas de la red que permiten saltarse determinadas capas intermedias. Al poder saltar estas capas, el entrenamiento de la red se aligera, reduciendo también el impacto de la profundidad de la red en el problema del desvanecimiento de gradientes al haber menos capas sobre las que propagarse. Esta técnica se puede aplicar tanto a redes neuronales artificiales (NN) como a redes convolucionales (CNNs).

A.4 Attention UNET

En esta modificación a UNET los autores proponen la utilización de un sistema de puertas en rejilla que permite que los coeficientes de atención sean más específicos a regiones locales. La arquitectura de red neuronal resultante se puede ver en la figura 14. En la figura 15 se muestra el detalle de las puertas de atención que se implementan en la entrada de los niveles de la subred de expansión.

Hay que notar que este modelo utiliza una solución aditiva de atención frente a otras opciones multiplicativas de atención, que de hecho son computacionalmente más sencillas pero no ofrecen resultados tan precisos como la propuesta. La atención aditiva se puede formular como 5:

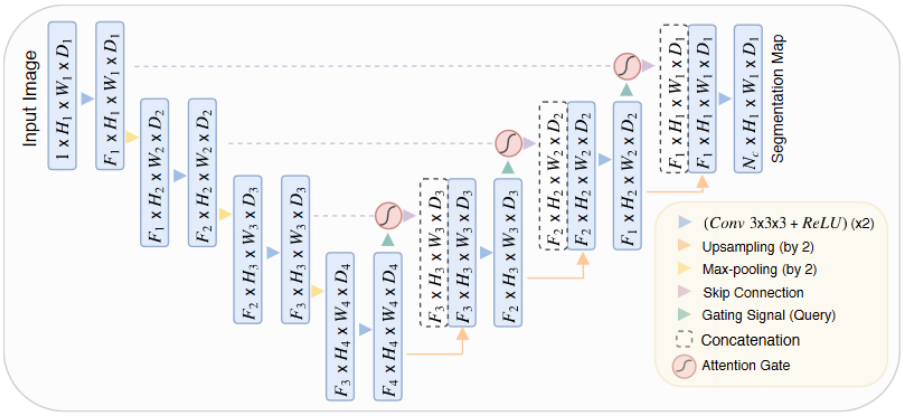

Figure 14: Arquitectura de la red Attention UNET.

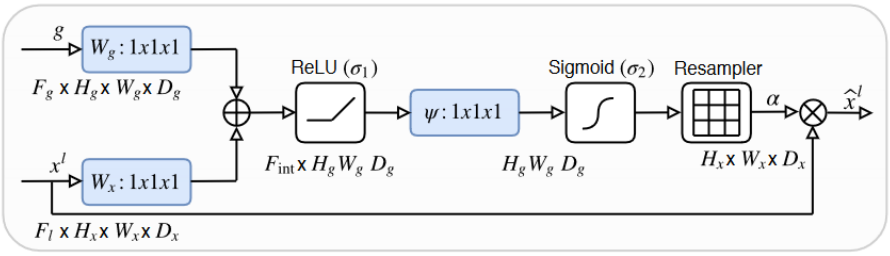

Figure 15: Detalle de una puerta de las puertas de atención utilizadas por Attention UNET.

$$
\begin{aligned}
q_{att}^{l} &= \psi^{T}(\sigma_1(W_x^{T} x_i^{l} + W_g^{T} g_i^{l} + b_g)) + b_\psi \\
\alpha_i^{l} &= \sigma_2(q_{att}^{l}(x_i^{l}, g_i; \Theta_{att}))
\end{aligned}
\tag{5}
$$

Donde $\sigma_2(x_i, c) = \frac{1}{1+\exp(-x_{i,c})}$ se corresponde a la función de activación sigmoide y Θ_{att} es el conjunto de parámetros formado por las transformaciones lineales $W_x \in R^{F_l \times F_{int}}$, $W_g \in R^{F_g \times F_{int}}$, $\psi \in R^{F_{int} \times l}$ y los sesgos $b_\psi \in R$ y $b_g \in R^{F_{int}}$.

A.5 Residuals Attention UNET

Tercera y última variación de UNET que vamos a analizar. En este caso, los autores combinan las unidades residuales y las puertas de atención para producir una nueva arquitectura capaz de trabajar con máscaras en alta resolución. El esquema general de la arquitectura se muestra en la figura 16.

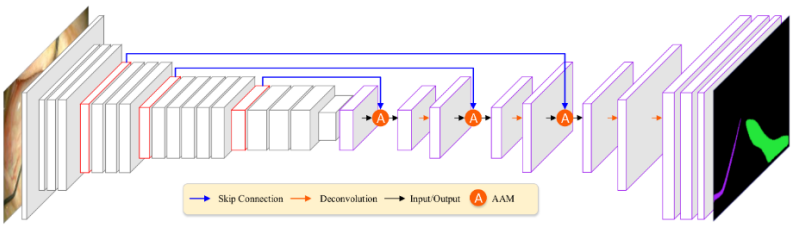

Figure 16: Detalle de la arquitectura de red neuronal propuesta en Residuals Attention UNET.

B. DATA

Durante el proceso se han generado y utilizado diferentes conjuntos específicos para la fase en la que nos encontrásemos, aunque todos ellos tienen como conjunto original *CDD-DDSM*. En esta sección se describen las diferentes versiones de los datos.

B.1 INPUT DATA

El conjunto de datos inicial, denominado CDD-DDSM, está formado a partir de la agregación de otros conjuntos de datos más pequeños. Se divide en dos subconjuntos:

- *Entrenamiento*: 1.166 imágenes y 1.253 máscaras.
- *Prueba*: 348 imágenes y 365 máscaras.

Las imágenes originales están en formato DICOM (Digital Imaging and COmmunications in Medicine). Este, es el formato estándar empleado para comunicación y gestión de información médica de imagen e información relacionada. Este formato permite la integración de hardware de diferentes fabricantes utilizando un formato común para la transmisión y almacenamiento de la informción.

Este formato fue desarrollado por la American College of Radiology (ACR) y la National Electrical Manufactures Association (NEMA). La primera versión del formato se liberó en 1985, seguida rápidamente por una segunda versión en 1988. DICOM agrupa la información en una dataset compuesto por un número variable de atributos aleatorios, entre los que podemos encontrar el identificador del paciente, información de historia clínica, anotaciones sobre la imagen, etc. y un atributo especial que contiene la información de la imagen. Para garantizar la consistencia y calidad de la visualización de la información en diferentes dispositivos, la imagen se codifica en escala de grises utilizando una tabla de búsqueda definida por el comité DICOM.

B.2 AS DATA

De cara a preparar un conjunto de datos confiable para el entrenamiento de los modelos de segmentación automática se realizó un análisis exploratorio de estos para comprender mejor sus características.

Tras analizar los datos se comprobó que las imágenes que componían el dataset presentan diferentes resoluciones, variando el ancho entre 1.786 y 5431 píxel y el alto entre 3.920 y 6.931 píxel, por lo que fue necesario normalizarlas para poder trabajarlas. Todas las imágenes fueron ajustadas a las mismas dimensiones de 1.750×3.500 píxel. Para preservar la máxima cantidad de información y teniendo en cuenta las características del dataset, las imágenes fueron transformadas a las dimensiones utilizando una combinación de reescalado y centrado y recorte (center & crop) de las imágenes a las dimensiones deseadas. Durante el proceso además se eliminaron las imágenes con más de una máscara de predicción (más de una lesión detectada como verdad base) por estar fuera del alcance.

Esto nos permite construir el dataset de segmentación automática, el cual se divide en tres subconjuntos de imágenes y sus correspondientes máscaras:

- *Entrenamiento*: 1.000 imágenes. Utilizado durante el entrenamiento del modelo para ajustar sus pesos.
- *Validación*: 108 imágenes. Utilizado durante la validación del entrenamiento para corregir la red en consecuencia y garantizar que aprenda.
- *Prueba*: 200 imágenes. Utilizado durante la prueba y selección de modelo para comprobar la calidad del ajuste y sus capacidad de generalización.

B.3 IIS DATA

Cuando se dispuso de un modelo de segmentación automática se procedió a predecir cada una de las 1.000 imágenes del dataset de segmentación automática, produciendo para cada imagen una predicción. Esta predicción, junto con la verdad base original, nos permite detectar las regiones de acierto y de error en la predicción, pudiéndolas utilizar posteriormente para extraer información enriquecida con la que entrenar el modelo de segmentación interactiva.

La región de acierto entre la máscara original (verdad base) y la predicción de AS es la intersección entre ambas imágenes. Las regiones de error son aquellas zonas en las que la predicción de AS ha fallado, bien por no detectar lesión (falsos negativos) o por detectarla sin que existiera (falsos positivos). En la ***[REVISAR]*** se trata con mayor profundidad este apartado.

B.4 PIPELINES

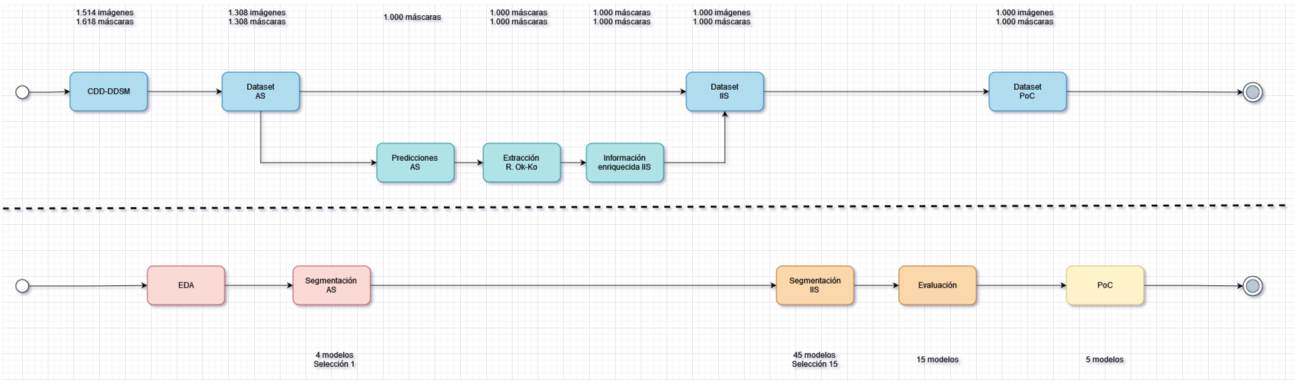

Figure 17: Pipeline de datos y de procesos implementados durante el trabajo. En la primera fila se puede apreciar el proceso de preparación y manipulación de datos desde la entrada del conjunto de datos inicial hasta la finalización del proceso. Paralelamente, en la segunda fila, se aprecian los procesos ejecutados *intercalados con el paso* del proceso de datos correspondiente. En la primera fila se muestra también el volumen de imágenes y máscaras manejado en cada momento. En la segunda fila se muestra el volumen de modelos pendientes de selección tras cada paso.

El proceso completo de preparación de datos y entrenamiento/selección de modelos completo que se ha seguido durante la práctica se puede apreciar en la figura 17.

En el carril superior, carril de datos, se muestra la preparación de los datos de principio a fin, mientras que el carril inferior, carril de procesos, se muestran los procesos realizados con los datos disponibles. Aunque no hay conexiones directas entre ambos carriles, se encuentran sincronizados, de forma que entre los pasos de *CDD-DDSM* y *Dataset AS* del carril de datos, se ejecuto el paso *EDA* del carril de procesos.

C. Métricas

Las métricas utilizadas para la evaluación de los modelos han sido:

- *DICE*,
- *IoU*,
- *Distancia Hausdorff*,
- *Average Area Difference*,
- *Sensitivity*,
- *F1 Score*,

C.1 El problema de la precisión a nivel de píxel

La métrica de precisión se puede expresar como en la ecuación 6.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{6}$$

Siendo TP el total de *verdaderos positivos*, TN el total de *verdaderos negativos*, FP los *falsos positivos* y FN los *falsos negativos*.

Supongamos dos imágenes de 10×10 píxel cada una como en el caso de la figura 18. En esta figura podemos considerar: TP=1, TN=96, FP=0, FN=3. Si aplicamos la métrica de la precisión a ambas imágenes, vemos que la imagen segmentada tiene una precisión del 97%, lo cual refleja incorrectamente la realidad de la bondad del ajuste entre las dos imágenes.

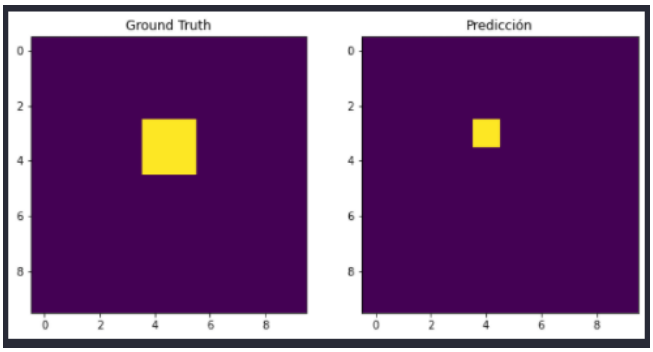

Figure 18: Dos imágenes de 10x10 píxel, la primera con un área de 2x2 píxel encendidos. La segunda con un área de 1x1 píxel encendido.

Viendo los resultados parecería que hemos obtenido un buen modelo capaz de segmentar correctamente, sin embargo, como se aprecia en la imagen, ni se acerca. Este problema se conoce como ***balanceo de clases*** (*class imbalance* en inglés) y se produce cuando una clase es mucho

más frecuente que las otras clases. Por ejemplo en el caso de la segmentación biomédica, la zona anotada o verdad base suele ser un área mucho más reducida que el resto de la imagen, como en la imagen de ejemplo.

Para solucionar estos problemas se plantea el uso de las métricas expuestas a continuación.

C.2 Dice Simmilarity Index

El coeficiente Sørensen–Dice se define como:

$$\text{DICE} = \frac{2 \times |X \cap Y|}{|X| + |Y|} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{7}$$

Siendo TP el total de *verdaderos positivos*, FP los *falsos positivos* y FN los *falsos negativos*. Gráficamente se puede entender como:

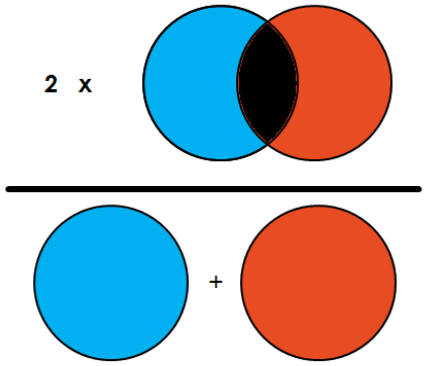

Figure 19: Representación visual del índice de similaridad Dice.

Sobre el ejemplo de la imagen 19, vemos que la puntuación obtenida es 0.4, lo que es más eficaz representando el éxito de la segmentación.

C.3 Intersections over Unions

El índice de Jaccard, propuesto simultáneamente tres veces diferentes, mide la relación entre *las especies comunes en dos regiones de los Alpes frente al total de especies de cada región*. De esta forma se puede expresar como:

$$IoU = \frac{\text{Intersection}}{\text{Union}} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{8}$$

Siendo TP el total de *verdaderos positivo*, FP los *falsos positivos* y FN los *falsos negativos*.

Sobre el ejemplo de la imagen 20, vemos que la puntuación obtenida es 0.25, lo que es más eficaz representando el éxito de la segmentación.

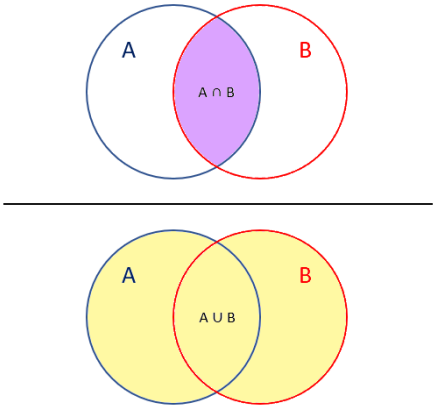

Figure 20: Representación visual del índice de Jaccard.

C.4 Distancia de Hausdorff

Es una métrica empleada en matemáticas para medir la distancia a la que se encuentran dos subconjuntos de un espacio métrico. Se expresa como:

$$d_H(X,Y) = \max\left\{\sup_{x\in X} d(x,Y), \sup_{y\in Y} d(X,y)\right\} \tag{9}$$

Donde sup es el *supremum* del conjunto, $d(a,B) = \inf_{y\in Y} d(a,B)$ es la distancia mínima de un punto a tal que $a \in A$ a todos los puntos del conjunto B.

Para implementar este método hay que tener en cuenta el coste computacional de calcular la distancia de cada punto de un conjunto a los todos los puntos de otro conjunto, siendo el producto cartesiano de ambos conjuntos y aumentando notoriamente la memoria requerida para calcular según se incrementa el tamaño de los conjuntos con que se trabaja.

Por esta razón, un paso previo para calcular esta métrica para cada verdad base y su predicción consiste en *erosionar* los conjuntos de cada imagen. Esta operación, básicamente desgasta las figuras de las imágenes hasta quedarse solo con el borde. De esta forma, si por ejemplo tenemos una figura cuadra de 10×10 píxel, tras erosionar pasaría de tener 100 puntos a comparar (cada píxel que compone el cuadrado) a *solo* 40 puntos (el perímetro). Cuando consideramos ambas imágenes, encontramos que el problema pasa de comparar 100 puntos con 100 puntos (10.000 combinaciones) a 40 con otros 40 (1.600). El resultado de *erosionar* una imagen se puede observar en 21

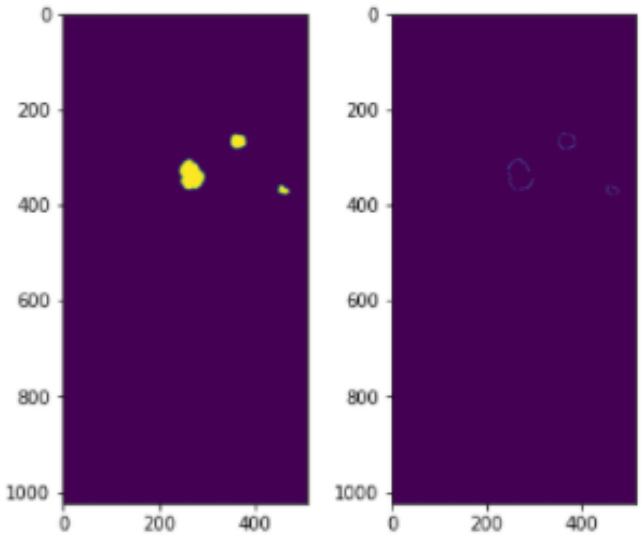

Figure 21: Proceso de erosión sobre una imagen. Se ha eliminado cada píxel del cuerpo de la mancha, dejando solo aquellos píxel pertenecientes al borde de cada mancha.

C.5 Average Area Difference

Está métrica es la diferencia de áreas de las dos imágenes. Para calcular las áreas de las imágenes, se han contado el total de píxel presentes en cada una y se ha obtenido la diferencia.

$$AAD = \frac{|Y_t - Y_p|}{Y_t}$$

Donde Y_t es la verdad base anota e Y_p la segmentación realizada por el modelo. Al tomar el valor absoluto, esta métrica penaliza por igual tanto a los modelos que sobre segmentan como a los que no segmentan suficiente.

C.6 Sensibilidad

La sensibilidad o exhaustividad es la tasa de verdaderos positivos (intersección) captados por el modelo del total de positivos. Se puede entender como una métrica de la cantidad de información captada correctamente.

Matemáticamente se puede expresar como:

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

C.7 Puntuación F1

La puntuación F1 trata de ofrecer una métrica combinada de la sensibilidad y la precisión, expresada como:

$$F1 = 2 \times \frac{precission * recall}{precission + recall} \tag{11}$$

Teniendo en cuenta que la precisión es una medida de la calidad del modelo que indica el porcentaje de verdaderos positivos sobre todos los positivos detectados. Matemáticamente se puede expresar como:

$$Precission = \frac{TP}{TP + FP} \tag{12}$$

D. CÓDIGO

El proyecto y los experimentos elaborados han sido desarrollados utilizando cuadernoS Jupyter de forma que se pudiera tener junto al código las opciones, consideraciones y razones. A continuación se describe cada uno de los cuadernos suplidos junto al trabajo.

Adicionalmente a los cuadernos, las funciones más comunes y reutilizadas en los diferentes cuadernos fueron llevadas a sus propios scripts de la biblioteca del proyecto de forma que se pudieran importar y reutilizar en posteriores desarrollos.

0 - Preview UNET

Cuaderno de toma de contacto del problema de segmentación utilizando redes convolucionales completas, concretamente la arquitectura UNET. Esta era una de las dos arquitecturas propuestas para el trabajo, siendo YOLO la alternativa. Después de indagar un poco en la literatura sobre segmentación, se decidió utilizar UNET por dos razones: Experiencia previa con YOLO y conocimiento de la dificultad de entrenar esta arquitectura y UNET se presentaba como una opción habitual y sólida para segmentación de imágenes en el campo de la biomedicina (eficaz entrenando con datasets no muy grandes y diferentes variaciones sobre la versión original en las que trabajar).

En este cuaderno se presenta una implementación básica de UNET utilizando Tensorflow2 para el dataset ISBI-2012.

1 - EDA

Recibido el conjunto de datos para el trabajo se procedió a analizar su composición.

En este cuaderno se desarrolla el análisis exploratorio de los datos, así como la exportación del dataset a formato PNG y una resolución constante de 1750×3500 píxel, tanto para la imagen como para su máscara.

4 - Selección AS

Para la selección de un modelo para segmentación automática se implementaron tres variaciones de la versión básica de UNET: Residuals UNET, Attention UNET y Residuals Attention UNET, que junto a la versión básica de UNET se procedieron a evaluar.

En este cuaderno se definen las tres variaciones de UNET que se van a considerar para a continuación entrenar cada uno de los cuatro modelos y seleccionar uno.

5 - Predicción AS

Para la selección interactiva de las imágenes vamos a utilizar la información de una segmentación automática previa además de información simulada de interacciones del usuario con la imagen para mejorar la segmentación.

En este cuaderno se predicen cada una de las 1.000 imágenes originales del conjunto de entrenamiento utilizado para le modelo de segmentación automática. Para cada una de estas imágenes se obtiene una segmentación que se procede a persistir en formato PNG para su

posterior tratamiento.

15 - Extracción regiones

Para simular las interacciones del usuario con la segmentación automática del usuario, punto de partida de nuestra segmentación interactiva, debemos extraer las regiones donde la segmentación automática ha acertado y ha fallado.

En este cuaderno se determinan las regiones de acierto y de error de la segmentación automática y persisten en formato PNG para su posterior reutilización.

16 - Linea Base IIS

Línea base para la segmentación interactiva.

En este cuaderno se desarrolla la línea base para la segmentación interactiva en la que se utilizan tal cual, sin procesar los puntos de interés ni calcular las máscaras enriquecidas con los mapas de probabilidad generados a partir de los POIs, se entrena un modelo utilizando la arquitectura Attention UNET para segmentación interactiva.

17 - Selección Puntos Semilla

Para aplicar las cuatro estrategias de selección de puntos explicadas en el apéndice C.5, se generan nuevas máscaras a partir de las regiones de acierto y de error de la segmentación original. Estas máscaras contienen los mapas de probabilidad construidos a partir de los puntos de interés extraídos mediante las estrategias definidas.

En este cuaderno se implementan las cuatro estrategias (cinco implementaciones en total, dos para centro de masas) de selección de puntos de interés, se generan las máscaras con los mapas de probabilidad y se persisten para su posterior reutilización.

21 - Entrenamiento IIS

En este cuaderno se entrenan los diferentes modelos para segmentación interactiva, resultantes de la combinación de configuración de la información adicional y estrategia para extracción de los puntos semilla. Se pueden entrenar los doce modelos en una sola ejecución del cuaderno, pero la recomendación es entrenar menos modelos en varias rondas.

25 - Evaluación

En este cuaderno se evalúan los modelos utilizando las métricas propuestas en el apéndice C.8.

100 - PoC

En este cuaderno se implementa una pequeña prueba de concepto que, cargando los modelos para segmentación automática e interactiva previamente entrenados permite al usuario interactuar con diferentes imágenes no utilizadas previamente para el entrenamiento del modelo interactivo, ofreciendo al final del proceso la puntuación de la nueva segmentación tras considerar la información aportada por el usuario.