Universidad Nacional de Educación a Distancia

Escuela Técnica Superior de Ingeniería Informática

Aplicación de técnicas de aprendizaje profundo a la segmentación de imagen ecocardiográfica

Itziar Marta Castilla Cebas

Director: Jorge Pérez Martín

Co-director: Francisco Javier Díez Vegas

Trabajo de Fin de Máster

Máster Universitario en Ingeniería y Ciencia de Datos

Septiembre 2023

Agradecimientos

Quiero dedicar este TFM a todos los que me han acompañado en este camino.

RESUMEN

La fiebre reumática es una enfermedad infecciosa que afecta con mayor frecuencia a los niños de 5 a 15 años, aunque se puede presentar también en adultos y muy rara vez en niños más pequeños. Suele causar daños en varias partes del cuerpo, especialmente en las válvulas del corazón, dando lugar así a la enfermedad cardíaca reumática (ECR).

Se estima que en la actualidad esta enfermedad afecta a más de 30 millones de personas en el mundo y causa más de 300.000 muertes anuales, muchas de ellas de personas menores de 25 años, además de provocar invalidez permanente en muchos casos. Los problemas que ocasiona son más frecuentes durante el embarazo y el parto.

En países endémicos la prevención primaria es complicada principalmente por la falta de acceso a servicios médicos cualificados por lo que la prevención secundaria, consistente en realizar un cribado mediante ecocardiografía para detectar los pacientes asintomáticos y aplicarles tratamiento profiláctico con penicilina, ha demostrado ser una vía de acción más efectiva.

El principal objetivo de esta línea de TFM es diseñar un sistema inteligente que, mediante técnicas de inteligencia artificial aplicadas al procesamiento de imágenes ecocardiográficas, sirva de ayuda al diagnóstico de la enfermedad cardíaca reumática.

He comparado cuatro arquitecturas diferentes: una red neuronal convolucional (CNN) con tres capas convolucionales, una ResUnet, U-Net y Laddernet. He probado diferentes técnicas para mejorar el rendimiento de los modelos y reducir sus problemas de generalización, incluyendo preprocesar las imágenes con filtro CLAHE, filtro ecualizado, filtro gaussiano, filtro de Sobel y aumento de datos. Así mismo he hecho pruebas de todas las arquitecturas con diferentes tamaños de kérnel 2, 3 y 4.

Finalmente, en el conjunto de pruebas, he obtenido un valor de coeficiente DICE 0,805 para la arquitectura CNN propuesta; 0,890 para la arquitectura Laddernet, 0,895 para ResUnet, 0,894 para una red U-Net preentrenada y 0,911 para U-Net con aumento de datos.

ABSTRACT

Rheumatic fever is an infectious disease that most commonly affects children aged 5 to 15 years, although it can also occur in adults and very rarely in younger children. It usually causes damage to various parts of the body, especially the heart valves, thus giving rise to rheumatic heart disease (RHD).

It is estimated that this disease currently affects more than 30 million people worldwide and causes more than 300,000 deaths annually, many of them in people under 25 years of age, as well as causing permanent disability in many cases. The problems it causes are most frequent during pregnancy and childbirth.

In endemic countries, primary prevention is complicated mainly by the lack of access to qualified medical services, so secondary prevention, consisting of screening by echocardiography to detect asymptomatic patients and apply prophylactic treatment with penicillin, has proven to be a more effective course of action.

The main objective of this line of TFM is to design an intelligent system that, by means of artificial intelligence techniques applied to echocardiographic image processing, will aid in the diagnosis of rheumatic heart disease.

I have compared four different architectures: an convolutional neural network (CNN) with three convolutional layers, a ResUnet, U-Net and Laddernet. I have tested different techniques to improve the performance of the models and reduce their generalization problems, including preprocessing the images with CLAHE filter, equalized filter, Gaussian filter, Sobel filter and data augmentation. Likewise, I have tested all architectures with different kérnel sizes 2, 3 and 4.

Finally, in the test dataset, I obtained a DICE coefficient value 0.805 for the proposed CNN architecture; 0.890 for the Laddernet architecture, 0.895 for ResUnet, 0.894 for a pre-trained U-Net and 0.911 for U-Net with data augmentation.

Abreviaturas

ACNN Anatomically Constrained Neural Networks - Red neuronal con restricciones anatómicas.

ANN Artificial Neuronal Network - Red neuronal artificial

AUC Area under the curve - Área debajo de la curva

BERT Bidireccional Encoder Representation from Transformers - Representación codificadora bidireccional a partir de transformadores

BN Batch normalization - Normalización por lotes

CAD Computed-Aided Diagnosis – Diagnostico asistido por ordenador

CAMUS Cardiac Acquisitions for Multi-structure Ultrasound Segmentation (nombre propio)

CETUS Challenge on Endocardial Three-dimensional Ultrasound Segmentation (nombre propio)

CLAHE Contrast-Limited Adaptive Histogram Equalization - Ecualización adaptativa de histograma limitada por el contraste

CNN Convolutional Neuronal Network - Red neuronal convolucional

DICE Dice Index Coefficient Evaluation – Coeficiente de evaluación del índice Dice

DL Deep Learning – Aprendizaje profundo

ECR Enfermedad cardiaca reumática

ED End of diastole - Fin de la diástole

ES End of sistole - Fin de la sístole

FE Fracción de eyección

GPU Graphics processing unit - Unidad de procesamiento gráfico

GPT Generative Pretrained Transform - Transformación generativa preentrenada

IA Inteligencia Artificial

IoU Intersection over Union - Intersección sobre unión

ML Machine Learning - Aprendizaje automático

MLP Multilayer Perceptron- Perceptron multicapa

PLN Procesamiento de lenguaje natural

RAM Random Access Memory - Memoria RAM

ResNet Residual Network - Red Residual

RCNN Recurrent convolutional neural network - Red neuronal convolucional recurrente

RM Resonancia magnética

SVM Support vector machine - Máquina de vectores de soporte

TAC Tomografía axial computerizada

TFM Trabajo Fin de Máster

TPU Tensor processing unit - Unidad de procesamiento de tensores

UNED Universidad Nacional de Educación a Distancia

U-Net U-shaped net - Red en forma de U

VI Ventrículo izquierdo

VGG Visual Geometry Group Grupo de Geometría Visual (nombre propio)

Índice general

Índice de figuras

Índice de tablas

Capítulo 1 Introducción

La inteligencia Artificial (IA), en concreto en áreas como el Aprendizaje Automático (Machine Learning ML) y el aprendizaje automático (Deep Learning DL), ha experimentado grandes progresos en los últimos años. Estos avances han permitido la aparición de técnicas que permiten automatizar tareas que hasta ahora necesitaban intervención humana. La medicina es una de las áreas en las que la IA está ayudando a aportar soluciones automatizadas, mostrando altas cuotas de robustez y eficacia.

De acuerdo con (Yap et al., 2010), el *diagnóstico por imagen* consiste en la detección de patologías mediante observación por personal médico cualificado de una prueba visual. Existe una gran variedad de patologías que pueden ser estudiadas mediante el diagnóstico por imagen, como el cáncer, enfermedades cardiacas, etc. Los ejemplos de pruebas diagnósticas visuales también son amplios, incluyendo imágenes de ultrasonidos, microscópicas, radiológicas, etc. Además, también puede existir gran variedad en cuanto a la zona corporal u órgano en el que se aplique el diagnóstico. Todo esto hace que el abanico de problemas a los que puede aplicarse el diagnóstico por imagen sea muy amplio.

El uso de la diagnosis asistida por computador (Computer-Aided Diagnosis, CAD) ofrece varias ventajas en el campo de la medicina y el diagnóstico. Algunas de las más destacadas son las siguientes:

- Mejora de la precisión: El CAD utiliza algoritmos de aprendizaje automático y tecnologías avanzadas para analizar imágenes médicas y proporcionar una evaluación más precisa y objetiva. Puede ayudar a los médicos a detectar patrones o características sutiles que podrían pasar desapercibidos en una revisión visual. Esto puede llevar a una detección temprana de enfermedades y una precisión mejorada en el diagnóstico.
- Ayuda en la toma de decisiones clínicas: El CAD puede proporcionar información y recomendaciones adicionales a los médicos durante el proceso de diagnóstico. Esto puede ser especialmente útil en casos difíciles o ambiguos, donde el médico puede beneficiarse de una segunda opinión. Al combinar el juicio clínico del médico con los datos y análisis proporcionados por el CAD, se puede tomar una decisión más informada y precis

- Aumento de la eficiencia: El CAD puede ayudar a agilizar el proceso de diagnóstico al realizar tareas repetitivas y rutinarias de manera rápida y precisa. Esto libera tiempo para que los médicos se centren en aspectos más complejos y desafiantes de la atención médica. Además, el CAD puede analizar grandes volúmenes de datos en poco tiempo, lo que permite un análisis más rápido y una respuesta más oportuna.
- Reducción de errores y omisiones: Los seres humanos son propensos a cometer errores y pueden pasar por alto detalles importantes en la interpretación de imágenes médicas. El CAD actúa como una herramienta complementaria que puede ayudar a minimizar los errores y las omisiones, al proporcionar una revisión adicional y resaltar posibles áreas de preocupación.
- Acceso a conocimientos especializados: El CAD puede integrar una amplia base de conocimientos médicos y experiencia clínica acumulada. Puede ayudar a los médicos a acceder a información y referencias actualizadas sobre enfermedades, condiciones y tratamientos específicos. Esto es especialmente útil para médicos menos experimentados o en áreas remotas donde el acceso a expertos puede ser limitado.

En general, el uso del sistema CAD puede mejorar la precisión, la eficiencia y la calidad del diagnóstico médico. Sin embargo, es importante destacar que el CAD no reemplaza la experiencia y el juicio clínico de los médicos, sino que actúa como una herramienta complementaria.

El aprendizaje automático aplicado al diagnóstico asistido por ordenador (Computer-Aided Diagnosis, CAD) es una aplicación poderosa de la inteligencia artificial en el campo de la medicina y la salud.

Según la técnica y el órgano objeto de análisis, se concretan dominios específicos de tareas automatizables. El aprendizaje profundo y, en concreto, las redes neuronales, sobresalen en la asistencia a dichas tareas.

Los avances en la tecnología de la imagen y las técnicas de aprendizaje automático han hecho avanzar el diagnóstico y el tratamiento de las enfermedades cardiovasculares (Van Steenkiste, 2020).

La ecocardiografía es una prueba segura y de bajo coste para el diagnóstico cardiaco (Attia & Benazza-Benyahia, 2018). Es un examen no invasivo que observa todas las estructuras del corazón, es decir, las válvulas y las cavidades (aurículas y ventrículos).

La enfermedad cardíaca reumática (ECR) es una patología cardiovascular derivada de la fiebre reumática, una enfermedad infecciosa que afecta con mayor frecuencia a los niños de 5 a 15 años, aunque se puede presentar también en adultos y muy rara vez en niños más pequeños. La fiebre reumática suele causar daños en varias partes del cuerpo, especialmente en las válvulas del

corazón, dando lugar así a la ECR. Se estima que en la actualidad esta enfermedad afecta a más de 30 millones de personas en el mundo y causa más de 300.000 muertes anuales, muchas de ellas de personas menores de 25 años, además de provocar invalidez permanente en muchos casos. Los problemas que ocasiona son más frecuentes durante el embarazo y el parto.

En países endémicos la prevención primaria de la enfermedad cardíaca reumática, entendida esta como evitar los episodios de faringitis estreptocócica haciendo frente a la pobreza, mejorando las condiciones de vida y de las viviendas, y ampliando el acceso a la atención sanitaria, es complicada principalmente por la falta de acceso a servicios médicos cualificados. Por ello, la prevención secundaria, consistente en realizar un cribado mediante ecocardiografía para detectar los pacientes asintomáticos y aplicarles tratamiento profiláctico con penicilina, ha demostrado ser una vía de acción más efectiva.

1.1 Objetivos y tareas

Este trabajo de fin de máster se enmarca dentro del proyecto CAREUM, cuyo objetivo es diseñar un sistema inteligente que, mediante técnicas de inteligencia artificial aplicadas al procesamiento de imágenes ecocardiográficas, sirva de ayuda al diagnóstico de la enfermedad cardíaca reumática. Con ese fin, este trabajo está centrado en la segmentación de imagen ecocardiográfica, de forma que se pueda proponer a los cardiólogos una segmentación precisa, lo que podría suponer un ahorro de tiempo y esfuerzo, además de evitar subjetividades.

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

- Estudiar el dataset CAMUS y buscar otras posibles fuentes de datos a utilizar en el presente trabajo.
- Analizar las estrategias empleadas en otros trabajos para la segmentación de imágenes en general y la segmentación de ecocardiografías en particular.
- Seleccionar/estudiar varias arquitecturas 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 cavidades del corazón.
- Evaluar los resultados de las estrategias propuestas.
- Implementar una prueba de concepto (PoC) para validar la viabilidad técnica de la solución.

Este trabajo se organiza de acuerdo con la siguiente estructura:

- Fundamentos teóricos.
- Estado del arte.
- Materiales y métodos.
- Resultados.
- Discusión y conclusiones.

Capítulo 2 Fundamentos teóricos

2.1 Función cardiaca

De adentro hacia afuera, la estructura del corazón presenta las siguientes capas (Wikipedia, 2023b):

- Endocardio. Tapiza las cavidades internas del corazón, tanto aurículas como ventrículos. Está formado por una capa endotelial, en contacto con la sangre.
- Miocardio. Es la capa más ancha y representa la mayor parte del grosor del corazón. Está formada por tejido muscular encargado de impulsar la sangre mediante su contracción. La anchura del miocardio no es homogénea, es mucho mayor en el ventrículo izquierdo y menor en el ventrículo derecho y las aurículas.
- Pericardio. Se trata de una membrana fibroserosa que envuelve al corazón y lo separa de las estructuras vecinas. Forma una especie de bolsa o saco que cubre totalmente al corazón y se prolonga hasta las raíces de los grandes vasos.

El corazón está dividido en 4 cavidades, como se puede ver en la Figura 2.1:

- 2 cavidades superiores: las aurículas derecha e izquierda, separadas por el tabique interauricular.
- 2 cavidades inferiores: los ventrículos derecho e izquierdo, separados por el tabique interventricular.

Las aurículas se comunican con los ventrículos a través de las válvulas auriculoventriculares. Así pues, distinguimos el corazón derecho, formado por una aurícula y un ventrículo derechos que se comunican a través del orificio tricúspide, y el corazón izquierdo, formado por una aurícula y un ventrículo izquierdo que se comunican a través del orificio mitral. Cada orificio atrio ventricular incluye un aparato valvular formado por un anillo fibroso, válvulas y cordones que conectan las válvulas con los pilares musculares que se insertan en el endocardio. Los orificios aórticos (situado a la entrada de la aorta) y pulmonar (situado a la entrada de la arteria pulmonar) están formado por un anillo fibroso y tres válvulas denominadas sigmoideas (M. Pierre, 2020).

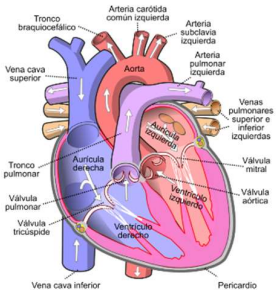

Figura 2.1 El corazón mostrando válvulas, arterias y venas (imagen tomada de (Wikipedia, 2023a)).

La frecuencia cardiaca depende de 2 componentes:

- Un componente mecánico o ciclo cardiaco que puede simplificarse en 3 fases: (M. Pierre, 2020)
    - Una fase de relajación denominada diástole que permite el llenado de sangre de las cavidades cardíacas.
    - Una fase de contracción denominada sístole que se caracteriza por un aumento de la presión intracavitaria.
    - Una fase de eyección de la sangre hacia la red circulatoria.
- Un componente eléctrico directamente responsable de la fase mecánica. El músculo cardiaco está dotado de automatismo: posee fibras musculares especializadas que generan una actividad eléctrica repetitiva de manera espontánea. Estas fibras constituyen lo que se denomina tejido nodal, que tiene un primer grupo celular en la pared de la aurícula derecha, cerca de la desembocadura de la vena cava superior: el nódulo sinoauricular. Éste tiene la propiedad de generar rítmicamente un impulso eléctrico que se propaga a las dos aurículas y provoca su contracción. A continuación, la señal eléctrica se transmitirá al nódulo auriculoventricular y después se retransmitirá a los ventrículos gracias al haz de His y a la red de Purkinje. La contracción de los ventrículos se produce unas fracciones de segundo después de la de las aurículas, debido al tiempo de propagación del impulso nervioso. Por otra parte, hay sístoles auriculares derecha e izquierda casi simultáneos (seguidos de diástoles) y también sístoles simultáneas (seguidos de diástoles) de los ventrículos derecho e izquierdo. La frecuencia cardiaca en reposo es de una media de 60 a 80 latidos por minuto en personas adultas (M. Pierre, 2020).

2.2 Imagen médica

2.2.1 Visión general

El rápido progreso de la ciencia médica y el desarrollo de diversos medicamentos han beneficiado a la humanidad. La ciencia moderna también ha hecho maravillas en el campo quirúrgico. Pero el diagnóstico adecuado y correcto de las enfermedades es la necesidad primordial antes del tratamiento. Cuanto más sofisticados sean los bioinstrumentos, mejor será el diagnóstico. Las imágenes médicas desempeñan un papel importante en el diagnóstico clínico y la terapia del médico, así como en la enseñanza, la investigación, etc.

La imagen médica suele considerarse una forma de representar las estructuras anatómicas del cuerpo con la ayuda de la tomografía computarizada y la resonancia magnética (RM), pero a menudo estas técnicas son más útiles para la función fisiológica que para la anatomía. Con el progreso de la tecnología, la imagen médica ha influido enormemente en el campo de la salud.

Como la calidad de las imágenes médicas influye en el diagnóstico, el procesamiento de imágenes médicas se ha convertido en un tema candente y las aplicaciones clínicas que desean almacenar y recuperar imágenes para fines futuros necesita algún proceso conveniente para almacenar esas imágenes en detalle. He aquí algunos tipos de imágenes médicas:

- Radiografía
- Resonancia magnética
- Medicina nuclear
- Ecografía
- Tomografía

2.2.2 Ultrasonido médico

Los ultrasonidos médicos utilizan ondas sonoras de banda ancha de alta frecuencia en el rango de los megahercios que se reflejan en el tejido en distintos grados para producir imágenes.

Su uso suele asociarse a la obtención de imágenes del feto en mujeres embarazadas. Sin embargo, los usos de los ultrasonidos son mucho más amplios, por ejemplo, la obtención de imágenes de los órganos abdominales, el corazón, las mamas, los músculos, los tendones, las arterias y las venas. Aunque puede proporcionar menos detalles anatómicos que técnicas como el TAC (Tomografía axial computarizada) o la resonancia magnética (RM) , tiene varias ventajas que la hacen de especial interés para el diagnóstico por imagen. En particular, estudia la función de estructuras en movimiento en tiempo real y no emite radiaciones ionizantes (Wikipedia, 2023).

La ecografía difiere de otras modalidades de imagen médica en el hecho de que funciona mediante la transmisión y recepción de ondas sonoras. Las ondas sonoras de alta frecuencia se envían y, dependiendo de la composición de los distintos tejidos, la señal se atenúa y se devuelve a intervalos separados. Un trayecto de ondas sonoras reflejadas en una estructura multicapa puede definirse mediante una impedancia acústica de entrada (onda sonora ultrasónica) y los coeficientes de reflexión y transmisión de las estructuras relativas. Su uso es muy seguro y no parece tener efectos adversos. Además, es relativamente barata y rápida de realizar. Los escáneres de ultrasonidos o ecógrafos pueden trasladarse a los pacientes en estado crítico en las unidades de cuidados intensivos, evitando el peligro causado durante el traslado del paciente al servicio de radiología. La imagen en movimiento obtenida en tiempo real puede utilizarse para guiar procedimientos de drenaje y biopsia. Las capacidades Doppler de los escáneres modernos permiten evaluar el flujo sanguíneo en arterias y venas.

La ecografía Doppler utiliza ondas sonoras para detectar el movimiento de la sangre y los tejidos. Las imágenes muestran la dirección y la velocidad de la sangre a medida que fluye por las arterias o las venas. También muestran el flujo sanguíneo a través del corazón, como puede verse en la Figura 2.2.

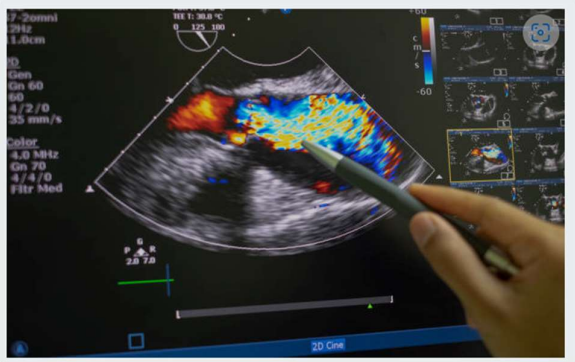

Figura 2.2 Imagen de ecografía Doppler (imagen tomada de (Hernández, 2020))

2.2.3 Ecocardiografía

Cuando los ultrasonidos se utilizan para obtener imágenes del corazón, se habla de ecocardiograma. Se trata de una técnica diagnóstica que permite ver en detalle las estructuras del corazón, incluido el tamaño de las cavidades, la función cardiaca, las válvulas, así como el pericardio. La ecocardiografía utiliza imágenes 2D, 3D y Doppler para crear imágenes del corazón y visualizar el flujo sanguíneo a través de cada una de las cuatro válvulas cardiacas. Esto permite a los médicos diagnosticar y evaluar diversas afecciones del corazón, como enfermedades de las válvulas, insuficiencia cardíaca, cardiopatías congénitas, coágulos de sangre y anomalías estructurales.

La ecocardiografía se utiliza tanto en pacientes que presentan síntomas –por ejemplo, dificultad para respirar o dolor torácico- como en pacientes que se someten a tratamientos contra el cáncer.

Existen diferentes tipos de ecocardiografía, como la ecocardiografía transtorácica (ETT), en la que el transductor se coloca en el pecho del paciente, y la ecocardiografía transesofágica (ETE), en la que el transductor se introduce a través del esófago para obtener imágenes más detalladas del corazón. Además, se puede utilizar la ecocardiografía de estrés para evaluar la respuesta del corazón al ejercicio o a medicamentos que lo estimulan.

Se ha demostrado que la ecografía transtorácica es segura para pacientes de todas las edades, desde fetos hasta ancianos, sin riesgo de efectos secundarios nocivos ni radiación, lo que la diferencia de otras modalidades de diagnóstico por imagen. La ecocardiografía es una de las modalidades de diagnóstico por imagen más utilizadas en el mundo debido a su portabilidad y a su uso en diversas aplicaciones. En situaciones de emergencia, la ecocardiografía es rápida, de fácil acceso y puede realizarse a pie de cama, lo que la convierte en la modalidad preferida de muchos médicos.

En la Figura 2.3 se muestra una imagen correspondiente a una ecocardiografía en las que se diferencian las aurículas y ventrículos.

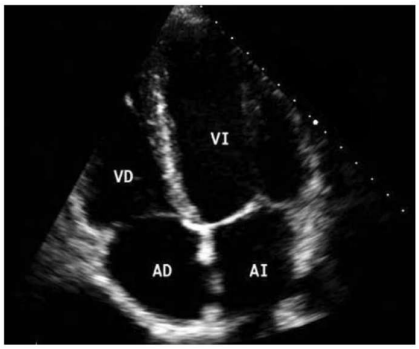

Figura 2.3 Imagen de ecocardiografía en la que se diferencian las aurículas y ventrículos. VD: Ventrículo Derecho, VI: Ventrículo izquierdo, AD: Aurícula derecha, AI: Aurícula izquierda (imagen tomada de (Chasco Ronda, 2010)

Diversas vistas del corazón pueden ser captadas mediante el ecocardiograma (Engelman et al., 2014). Cada vista se define mediante dos atributos:

1. Posición del transductor: Por ejemplo, paraesternal, apical, supraesternal y subcostal.
2. Plano de la imagen: Por ejemplo, eje largo, eje corto, de cuatro o cinco cámaras.

Existen cuatro posiciones principales donde colocar el transductor, como se puede ver en la Figura 2.4: Supraesternal, paraesternal, apical y subcostal.

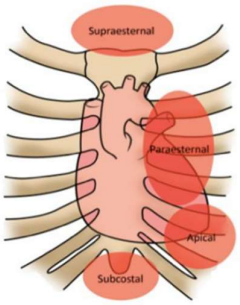

Figura 2.4 Posiciones transductor: Supraesternal, paraesternal, apical y subcostal (imagen tomada de (Serna, 2019)).

Los distintos planos de imagen se obtienen girando la sonda o inclinándola. Se trata de capturar imágenes del corazón desde distintos puntos de vista. Los planos de imagen son, como se muestra en la Figura 2.5: (A) corte longitudinal del ventrículo izquierdo, (B) corte transversal y (C) corte cuatro (Engelman et al., 2014):

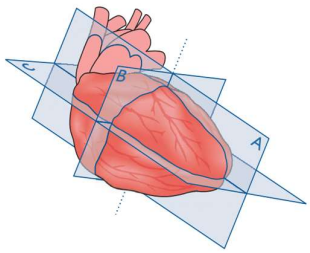

Figura 2.5 Planos ecocardiográficos del corazón: A: Corte longitudinal del ventrículo izquierdo; B: Corte transversal; C: Corte de cuatro cavidades (imagen tomada de (Engelman et al., 2014))

La combinación de los diferentes planos y posiciones del transductor da lugar a diferentes vistas ecocardiográficas (ver Figura 2.6), como pueden ser, eje largo paraesternal, eje corto paraesternal nivel de válvula aórtica y nivel de válvula mitral, apical 4 y 5 (Engelman et al., 2014)

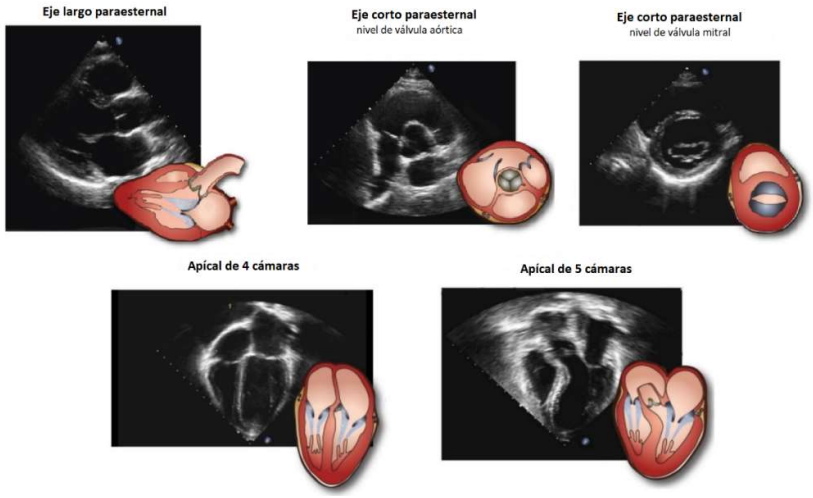

Figura 2.6 Ejemplo cinco vistas ecocardiográficas: Eje largo paraesternal, eje corto paraesternal nivel de válvula aórtica y válvula mitral, apical 4 y 5 cámaras (imágenes tomadas de (Serna, 2019)).

2.3 Visión por computador

Dentro de la visión por computador, existen distintas maneras de abordar la tarea de localización de objetos dentro de una imagen. La Figura 2.7 presenta un ejemplo gráfico de resolución de cada una de esas maneras. En esencia, se diferencian en qué tipo de información permiten predecir (Michelucci, 2019):

- **Clasificación:** consiste en asignar una etiqueta a una imagen o, en otras palabras, en identificar lo que hay en una imagen. Por ejemplo, una imagen de un gato puede tener la etiqueta "gato".
- **Clasificación y localización:** consiste en asignar una etiqueta a una imagen y determinar los bordes del objeto que contiene (y suele dibujar un rectángulo alrededor del objeto).
- **Detección de objetos:** Este término se utiliza cuando hay varias instancias de un objeto en una imagen. En la detección de objetos, se desea determinar todas las instancias de varios objetos (por ejemplo, personas, coches, señales, etc.) y dibujar recuadros delimitadores a su alrededor.
- **Segmentación de instancias:** Se desea etiquetar cada píxel de la imagen con una clase específica para cada instancia separada, para poder encontrar los límites exactos de la instancia del objeto.
- **Segmentación semántica:** se desea etiquetar cada píxel de la imagen con una clase específica. La diferencia con la segmentación por instancias es que en la segmentación

semántica no importa si existen varias instancias de un coche, por ejemplo. Todos los píxeles pertenecientes a los coches serán etiquetados como "coche". En la segmentación por instancias se debería ser capaz de decir cuántas instancias de un coche hay en la imagen y dónde están exactamente.

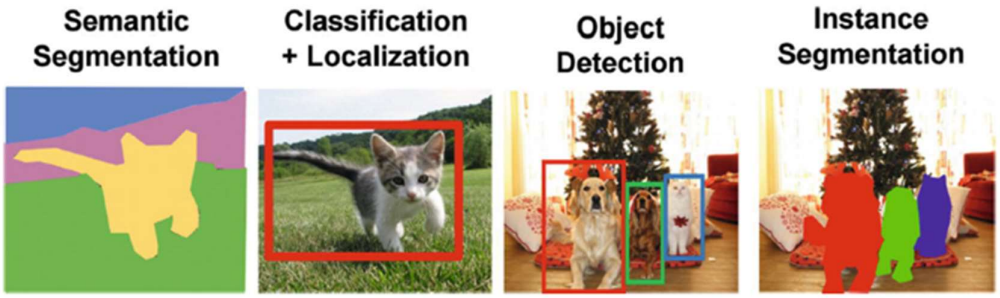

Figura 2.7 Ejemplos visuales de localización de objetos de visión por computador (imagen tomada de (Géron, 2019)).

La segmentación suele ser la tarea más difícil de todas ellas, y en particular la segmentación de instancias. Muchas técnicas avanzadas se unen para resolver esos problemas. Una de las cosas que hay que recordar es que conseguir suficientes datos de entrenamiento no es fácil. Hay que tener en cuenta que, con la segmentación, alguien tiene que clasificar cada píxel de la imagen, lo que significa que los datos de entrenamiento son difíciles de recopilar.

2.4 Aprendizaje automático y aprendizaje profundo

2.4.1 Tipos de aprendizaje automático

El aprendizaje automático es la ciencia (y el arte) de programar ordenadores para que puedan aprender de los datos (Géron, 2019). El aprendizaje automático se trata de un subcampo de la informática en el que las máquinas aprenden a realizar tareas sin tener que ser programadas explícitamente para ellas. Las máquinas detectan un patrón e intentan reproducirlo de algún modo que puede ser directo o indirecto.

Los sistemas de aprendizaje automático pueden clasificarse según los tipos de datos y el tipo de supervisión que reciben durante el entrenamiento. Existen cuatro categorías principales:

- aprendizaje supervisado
- aprendizaje no supervisado
- aprendizaje semisupervisado
- aprendizaje por refuerzo.

2.4.1.1 Aprendizaje supervisado

En el aprendizaje supervisado, los datos de entrenamiento que se entregan al algoritmo incluyen las respuestas deseadas, denominadas etiquetas (Géron, 2019). En la Figura 2.8 se muestran ejemplos de un conjunto de datos de entrenamiento con las etiquetas correspondientes indicando si se trata de un mail spam o no.

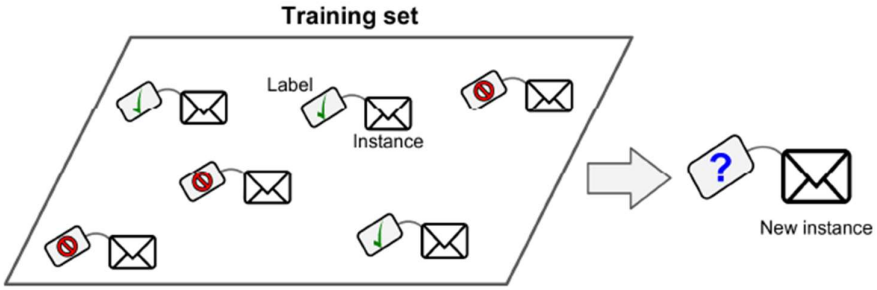

Figura 2.8 Un conjunto de entrenamiento etiquetado para el aprendizaje supervisado; por ejemplo, clasificación de spam (imagen tomada de (Géron, 2019)).

2.4.1.2 Aprendizaje no supervisado

En el aprendizaje no supervisado, los datos de entrenamiento no están etiquetados. Consiste en agrupar las instancias en cierto número de clases, no definido a priori. En la Figura 2.9 se muestra un ejemplo de conjunto de datos de entrenamiento para un algoritmo no supervisado en el que los datos no aparecen etiquetados.

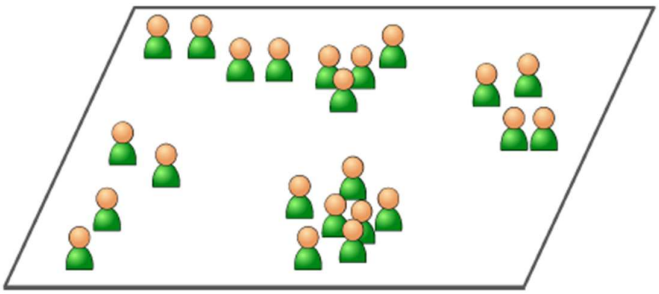

Figura 2.9 Un conjunto de entrenamiento no etiquetado para el aprendizaje no supervisado (imagen tomada de (Géron, 2019)).

2.4.1.3 Aprendizaje semisupervisado

Algunos algoritmos pueden trabajar con conjuntos de datos de entrenamiento parcialmente etiquetados, normalmente cuentan con muchos datos sin etiquetar y pocos etiquetados (Géron, 2019). En la Figura 2.10 se muestra un conjunto de datos para un algoritmo de aprendizaje.

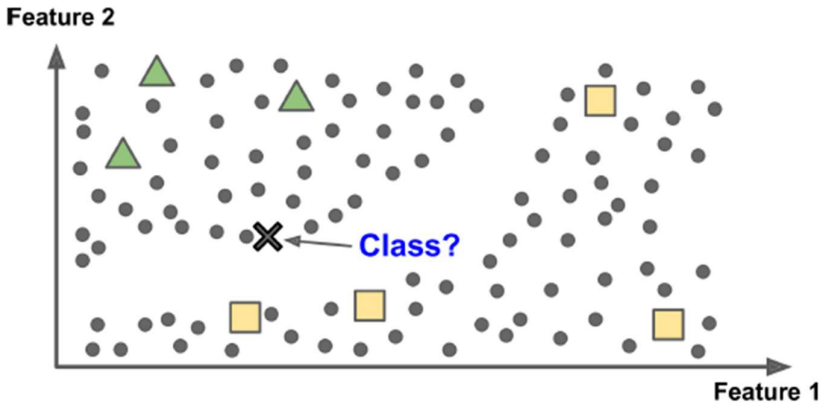

Figura 2.10 Aprendizaje semisupervisado (imagen tomada de (Géron, 2019)).

2.4.1.4 Aprendizaje por refuerzo

El en aprendizaje por refuerzo, el sistema que aprende se denomina *agente*. Puede observar el entorno, seleccionar y realizar acciones, y obtener recompensas a cambio (o penalizaciones, en forma de recompensas negativas) (Géron, 2019).

El agente debe aprender por sí mismo cuál es la mejor estrategia, llamada política, para obtener la mayor recompensa a lo largo del tiempo. Una política define qué acción debe elegir el agente en una situación determinada (Géron, 2019).

En la Figura 2.11 vemos un ejemplo de aplicación de aprendizaje por refuerzo.

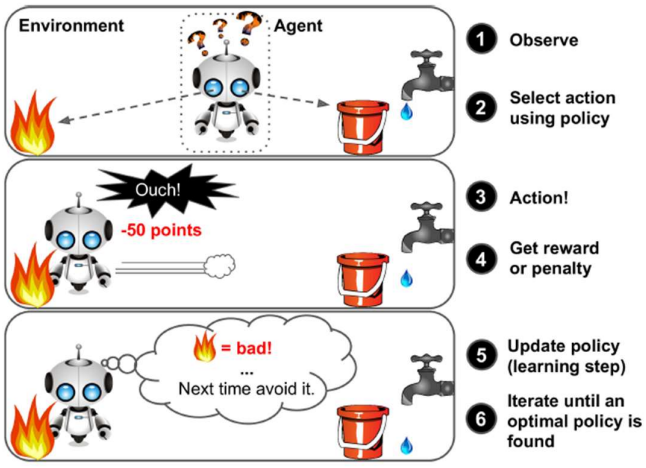

Figura 2.11 Aprendizaje por refuerzo (imagen tomada de (Géron, 2019).

2.4.2 Redes Neuronales

2.4.2.1 Origen

Dado que existen numerosos inventos inspirados en la naturaleza, parece lógico tomar como base la estructura del cerebro para construir sistemas inteligentes.

Nuestro cerebro está compuesto por aproximadamente 10.000 millones de neuronas, cada una conectada a otras 10.000 neuronas. El cuerpo celular de la neurona se denomina soma, donde las entradas (dendritas) y salidas (axones) conectan un soma con otro (ver Figura 2.12).

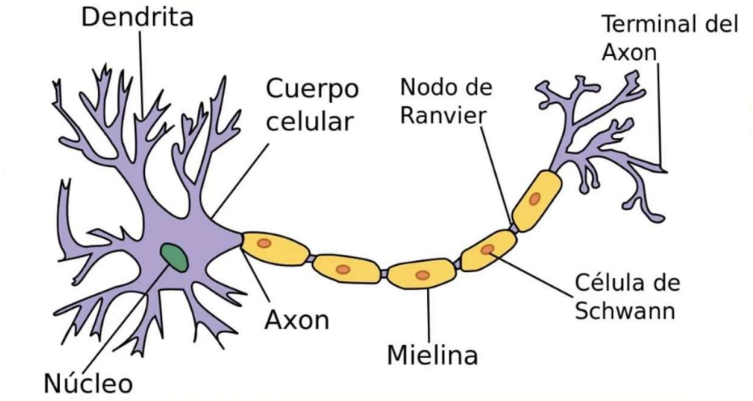

Figura 2.12 Estructura de una neurona biológica. Las neuronas están conectadas a otras neuronas a través de sus dendritas (imagen tomada de (Wikipedia, 2023b)).

Cada neurona recibe entradas electroquímicas de otras neuronas en sus dendritas. Si estas entradas eléctricas son lo suficientemente potentes como para activar la neurona, entonces ésta transmite la señal a lo largo de su axón, pasándola a las dendritas de otras neuronas. Estas neuronas también pueden activarse, continuando así el proceso de transmisión del mensaje.

La clave es que el disparo de una neurona es una operación binaria: la neurona se dispara o no se dispara. No hay diferentes "grados" de disparo. En pocas palabras, una neurona sólo se dispara si la señal total recibida en el soma supera un umbral determinado.

Las Redes Neuronales Artificiales (RNA) se inspiran en lo que sabemos sobre el cerebro y su funcionamiento. El objetivo del aprendizaje profundo no es imitar el funcionamiento de nuestro cerebro, sino tomar las piezas que entendemos y permitirnos establecer paralelismos similares en nuestro propio trabajo.

2.4.2.2 Perceptrón

Warren McCulloch y Walter Pitts propusieron un modelo muy simple de la neurona biológica, que más tarde se conoció como neurona artificial: tiene una o más entradas binarias (on/off) y una salida binaria. La neurona artificial simplemente activa su salida cuando más de un cierto número de sus entradas están activas (Géron, 2019).

El perceptrón se trata de una de las arquitecturas de RNA más sencillas; fue inventado en 1957 por Frank Rosenblatt (Vicente, 2023). Se basa en una neurona artificial ligeramente modificada (véase la Figura 2.13) llamada una *unidad lógica de umbral* (TLU), o a veces una unidad de umbral lineal (LTU): las entradas y salidas se tratan ahora de números (en lugar de valores binarios de

encendido/apagado) y además cada conexiónn de entrada está asociada con un peso. La TLU calcula una suma ponderada de sus entradas (Géron, 2019):

$z = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n = x^T w)$

Posteriormente se aplica una función de paso (*step*) a esa suma y se emite el resultado: h_w(x) = step(z), donde z = x^Tw.

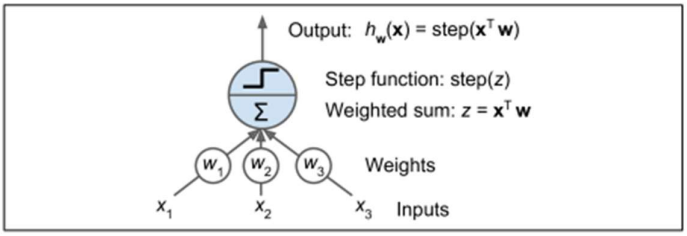

Figura 2.13 Ejemplo TLU (imagen tomada de (Géron, 2019).

Un perceptrón se compone de una sola capa de TLUs, en la que cada unidad está conectada a todas las entradas. Para representar que cada entrada se envía a cada TLU, es común dibujar neuronas de paso especiales llamadas neuronas de entrada: sólo dan salida a cualquier entrada que se les proporcione. Las neuronas de entrada forman la capa de entrada. Además, se suele añadir una característica de sesgo adicional (x_0 = 1), que se suele representar mediante un tipo especial de neurona llamada *neurona de sesgo*, que sólo emite 1 todo el tiempo (Géron, 2019). La adición de la característica de sesgo es útil porque sirve como otro parámetro del modelo que puede ajustarse para que el rendimiento del modelo en los datos de entrenamiento sea lo mejor posible.

En la Figura 2.14, se representa la estructura de un perceptrón con dos entradas y tres salidas. Este perceptrón puede clasificar instancias simultáneamente en tres clases binarias diferentes, lo que lo convierte en un clasificador multisalida (Géron, 2019).

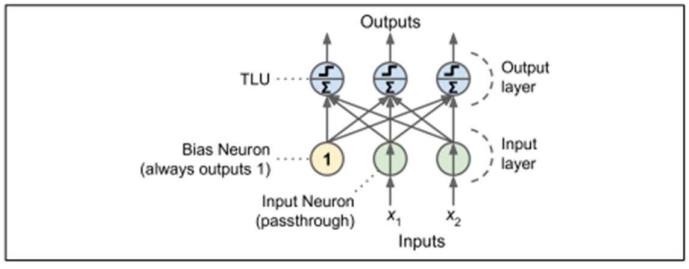

Figura 2.14 Diagrama perceptrón (imagen tomada de (Géron, 2019)).

2.4.2.3 Perceptrón multi-capa y retropropagación

Un perceptrón multi-capa o MLP se compone de una capa de entrada (de paso), una o más capas de TLUs, llamadas capas ocultas, y una capa final de TLUs llamada capa de salida (véase la Figura

2.15). Cada capa, excepto la de salida, incluye una neurona de sesgo y está totalmente conectada a la capa siguiente (Géron, 2019).

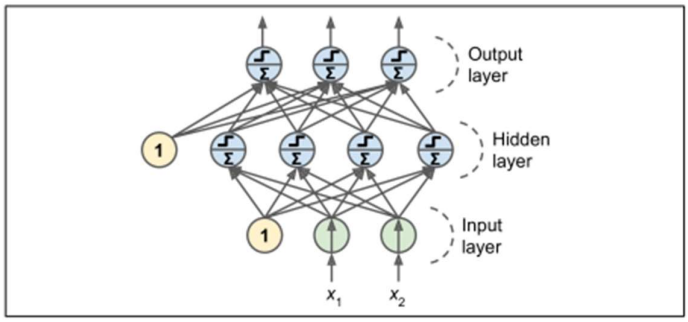

Figura 2.15 Diagrama perceptrón multi-capa (imagen tomada de (Géron, 2019)).

En 1986, David Rumelhart, Geoffrey Hinton y Ronald Williams publicaron un innovador artículo (Rumelhart et al., 1986) en el que presentaban el algoritmo de entrenamiento de retro propagación, que se sigue utilizando hoy en día (Géron, 2019).

El algoritmo de retropropagación utiliza una técnica eficiente para calcular los gradientes automáticamente: en sólo dos pasadas a través de la red, una hacia delante y otra hacia atrás, el algoritmo de retro propagación puede de calcular el gradiente del error de la red con respecto a cada parámetro del modelo. Es decir, es capaz de averiguar cómo debe ajustarse cada peso de conexión y cada término de sesgo para reducir el error. Una vez que tiene estos gradientes, sólo tiene que realizar un paso normal de *descenso del gradiente*, y todo el proceso se repite hasta que la red converja a la solución (Géron, 2019).

Veamos este algoritmo con más detalle (Géron, 2019):

1. Maneja un mini lote (subconjunto del conjunto de datos de entrenamiento) a la vez y repasa el conjunto completo de entrenamiento varias veces. Cada pasada se denomina época o iteración completa (*epoch*).
2. Cada mini lote se envía a la capa de entrada de la red, quien a su vez lo envía a la primera capa oculta. El algoritmo calcula la salida de todas las neuronas en esta capa. El resultado se pasa a la siguiente capa, su salida se computa y se pasa a la siguiente capa, y así sucesivamente hasta que obtenemos la salida de la última capa. Este es el paso hacia delante: es como hacer predicciones, salvo que todos los resultados intermedios se conservan, ya que son necesarios para el paso hacia atrás.
3. A continuación, el algoritmo mide el error de salida de la red; para ello utiliza una función de pérdida que compara la salida deseada y la salida real de la red, y devuelve alguna medida del error.

4. Posteriormente, calcula la contribución de cada conexión de salida al error. Esto se hace analíticamente aplicando la regla de la cadena, lo cual hace que este paso sea rápido y preciso.
5. Seguidamente, el algoritmo mide qué parte de estas contribuciones de error proceden de cada conexión de la capa inferior, utilizando de nuevo la regla de la cadena, y así sucesivamente hasta que el algoritmo llega a la capa de entrada. Este paso inverso mide eficientemente el gradiente de error a través de todos los pesos de conexión en la capa de entrada, propagando el gradiente de error hacia atrás a través de la red.
6. Finalmente, el algoritmo realiza un paso de descenso del gradiente para ajustar todos los pesos de conexión de la red, utilizando los gradientes de error que acaba de calcular.

Para que este algoritmo funcione correctamente, los autores introdujeron un cambio clave en la arquitectura del MLP: sustituyeron la función de paso por la función logística, σ(z) =1 / (1 + exp(-z)). Esto era esencial, dado que la función de paso sólo contiene segmentos planos, por lo que no hay gradiente con el que trabajar (el Descenso Gradiente no puede moverse en una superficie plana), mientras que la función logística tiene una derivada no nula bien definida en todas partes, lo que permite al Descenso Gradiente progresar en cada paso. De hecho, el algoritmo de retropropagación funciona bien con muchas otras funciones de activación, no sólo con la función logística (Géron, 2019).

Las funciones de activación más usadas son:

1) Función sigmoidea:

$$f(a) = \frac{1}{1 + exp(-a)}$$

2) Función tangente hiperbólica (o tanh):

$$f(a) = \frac{1 - \exp(-2a)}{1 + exp(-2a)}$$

3) Función de unidad lineal rectificada (o ReLU):

$$f(a) = \begin{cases} \text{a if a} \geq 0 \\ 0 \text{ if a} < 0 \end{cases}$$

En la Figura 2.16 se muestra la representación de las funciones de activación más comunes y sus derivadas:

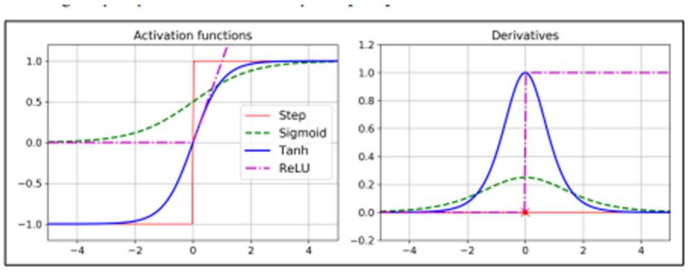

Figura 2.16 Funciones de activación y sus derivadas. (Géron, 2019)

2.4.2.4 Sobreajuste y subajuste

En el aprendizaje automático existe un equilibrio entre optimización y generalización. Según (Chollet, 2021) “la optimización se refiere al proceso de ajustar un modelo para obtener el mejor rendimiento posible en los datos de entrenamiento, mientras que la generalización se refiere a lo bien que se comporta el modelo entrenado en datos que nunca ha visto antes". Al principio del entrenamiento, tanto la optimización como la generalización están correlacionadas. Sin embargo, llega un momento en que el modelo empieza a aprender patrones específicos de los datos de entrenamiento (Fernandez de Toro Espejel, 2023).

Esto se conoce como el *trade-off* sesgo-varianza: el conflicto de intentar minimizar el sesgo y el error de varianza al mismo tiempo (Wikipedia, 2023a).

- Hay error de sesgo cuando el modelo no consigue captar características relevantes de los datos de entrenamiento (mala optimización). Esto se denomina "subajuste".
- Hay error de varianza cuando el modelo no es capaz de generalizar a datos diferentes de los de entrenamiento. El modelo está "sobreajustado".

En la Figura 2.17 se muestra los puntos de datos en naranja y las funciones de predicción en azul. La función de la izquierda sobreajusta los datos, la del centro los subajusta y la de la derecha los ajusta correctamente (Fernandez de Toro Espejel, 2023).

Figura 2.17 Compensación entre sesgo y varianza (imagen extraída de (Amazon Machine Learning, 2015)).

Cuando hay una diferencia significativa en el rendimiento del modelo en el conjunto de entrenamiento y en el conjunto de validación, hay sobreajuste (Fernandez de Toro Espejel, 2023).

2.4.3 Aprendizaje profundo

El aprendizaje profundo es un subcampo del aprendizaje automático, que es, a su vez, un subcampo de la IA. En la Figura 2.18 se muestra un diagrama de Venn en el que aparece reflejada la relación entre inteligencia artificial, aprendizaje automático y aprendizaje profundo.

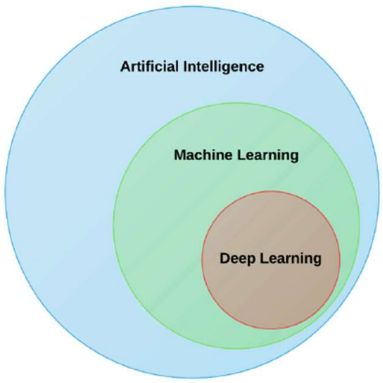

Figura 2.18 Diagrama de Venn que muestra la relación entre el aprendizaje profundo, el aprendizaje automático y la inteligencia artificial (Van Steenkiste, 2020).

El objetivo central de la IA es proporcionar un conjunto de algoritmos y técnicas que puedan utilizarse para resolver problemas que los humanos realizan de forma intuitiva y casi automática, pero que de otro modo son muy complicadas para los ordenadores. Un buen ejemplo de ello puede ser la interpretación del contenido de una imagen, una tarea que un ser humano resulta sencilla, pero que ha demostrado ser muy difícil hasta hace pocos años.

Aunque la IA engloba un amplio y variado conjunto de trabajos relacionados con el razonamiento automático de las máquinas, el subcampo del aprendizaje automático tiende a centrarse en el reconocimiento de patrones y el aprendizaje a partir de datos.

El aprendizaje profundo se podría definir como una clase de técnicas de aprendizaje automático, en las que información se procesa en capas jerárquicas para comprender representaciones y características de los datos en niveles crecientes de complejidad. En la práctica, todos los algoritmos de aprendizaje profundo son redes neuronales, que comparten algunas propiedades básicas comunes. Todos están formados por neuronas interconectadas que se organizan en capas; en lo que presentan diferencias es en la arquitectura y en ocasiones en la forma en la que se entrenan (Vasilev et al., 2019).

Teniendo esto en cuenta, veamos las principales clases de redes neuronales (Vasilev et al., 2019):

- **Perceptrones multicapa (MLP):** Red neuronal con propagación con capas totalmente conectadas y al menos una capa oculta.
- **Redes neuronales convolucionales (CNN):** Una CNN es una red neuronal directa con varios tipos de capas especiales. Por ejemplo, las capas

convolucionales aplican un filtro a la imagen (o sonido) de entrada deslizándolo por toda la señal entrante, para producir un mapa de activación n-dimensional. Existen indicios de que las neuronas de las CNN se organizan de forma similar a las células biológicas en la corteza visual de los seres humanos. Hoy en día, superan a todos los demás algoritmos de ML en un gran número de tareas de visión por ordenador y procesamiento natural del lenguaje.

- **Redes recurrentes:** Este tipo de red tiene un estado interno o memoria el cual se basa en la totalidad o en parte de los datos de entrada ya introducidos en la red. La salida de una red recurrente es una combinación de su estado interno (memoria de entradas) y la última muestra de entrada. Al mismo tiempo, el estado interno cambia para incorporar los nuevos datos de entrada. Debido a estas propiedades, las redes recurrentes son buenas candidatas para tareas que trabajan con datos secuenciales, como texto o series temporales.
- **Autocodificadores:** Son una clase de algoritmos de aprendizaje no supervisado, en los que la forma de la salida es la misma que la de entrada, lo que permite a la red aprender mejores representaciones básicas.

2.4.4 CNN

El objetivo de una CNN es aprender características de orden superior, entendiendo estas características como aquellas que representan patrones y relaciones más complejas y abstractas, en los datos mediante convoluciones. Estas características no son únicamente de una región de la imagen, sino de todo el conjunto de la imagen (ITelligent, 2018). Son muy adecuadas para el reconocimiento de objetos con imágenes y ocupan sistemáticamente los primeros puestos en los concursos internacionales de clasificación de imágenes.

Pueden identificar caras, personas, animales, señales de tráfico, y muchos otros aspectos de los datos visuales. Las CNN se solapan con el análisis de textos mediante el reconocimiento óptico de caracteres, pero también son útiles cuando se analizan palabras como unidades textuales discretas (Goodfellow et al., 2016).

La eficacia de las CNN en el reconocimiento de imágenes es una de las principales razones por las que son reconocidas las capacidades del aprendizaje profundo. Como ilustra la Figura 2.19, las CNN son buenas para construir características posición y (algo) invariantes de rotación a partir de datos de imágenes en bruto.

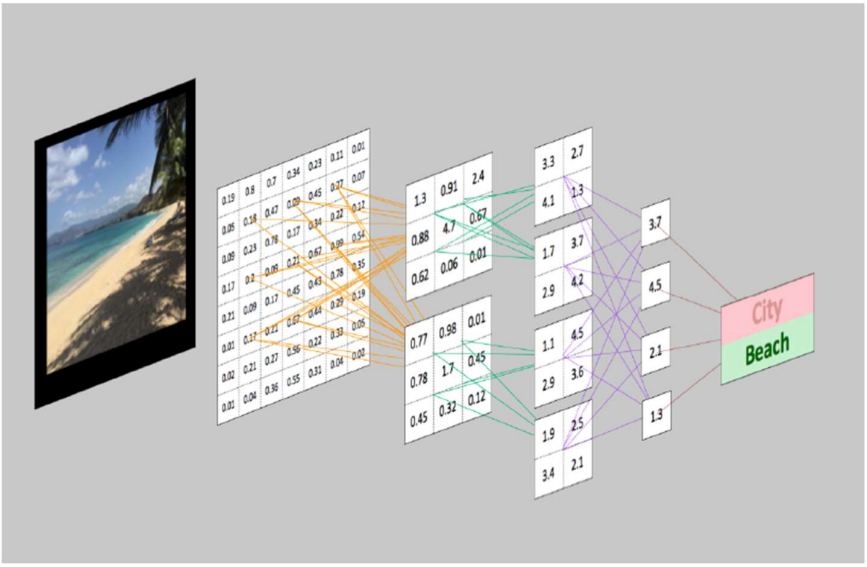

Figura 2.19 CNN y visión por ordenador (imagen tomada de (Patterson & Gibson, 2017).

Una red neuronal convolucional (CNN) es una estructura/arquitectura de aprendizaje profundo que puede reconocer y clasificar características en imágenes para visión por ordenador. Se trata de una red neuronal multicapa diseñada para analizar entradas visuales y realizar tareas como la clasificación de imágenes, la segmentación y la detección de objetos. Las CNN también pueden utilizarse para aplicaciones de aprendizaje profundo en la atención sanitaria, como el análisis de imágenes médicas.

Una CNN consta de dos partes principales (Khoshdeli et al., 2017):

- Las capas convolucionales, que extraen las distintas características de la imagen para su análisis.
- Una capa totalmente conectada que toma la salida de la última capa de convolución y obtiene la mejor descripción/clasificación de la imagen.

La arquitectura de las CNN se inspira en la organización y funcionalidad del córtex visual de los animales y está diseñada para imitar el patrón de conectividad de las neuronas del cerebro humano.

Las neuronas de una CNN se dividen en una estructura tridimensional, en la que cada grupo de neuronas analiza una pequeña región o característica de la imagen. En otras palabras, cada grupo de neuronas se especializa en identificar una parte de la imagen. Las CNN utilizan las predicciones de las capas para producir una salida final que presenta un vector de puntuaciones de probabilidad para representar la probabilidad de que una característica específica pertenezca a una determinada clase.

Una CNN se compone de varios tipos de capas como se muestra en la Figura 2.20**.**

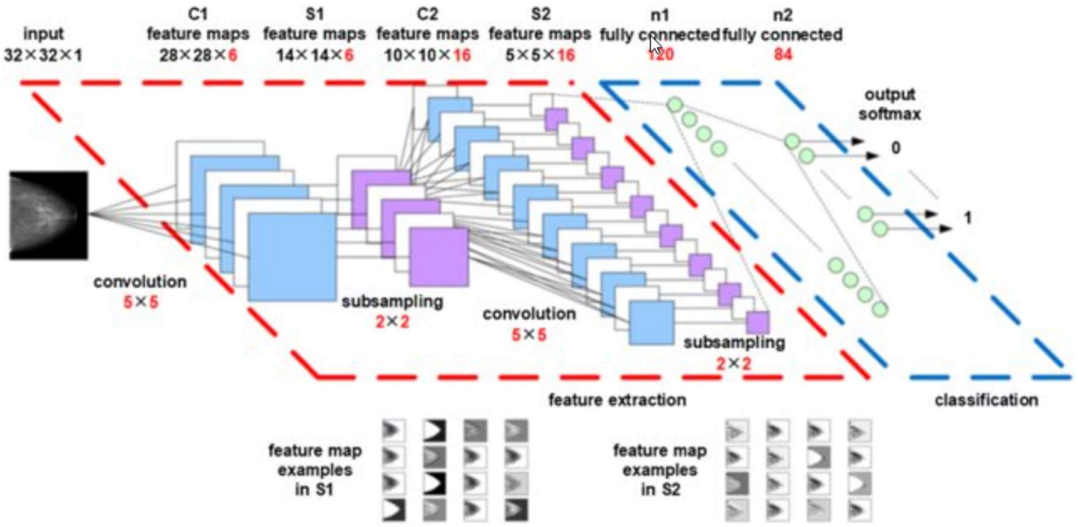

Figura 2.20 Ejemplo estructura CNN (imagen (Lan et al., 2020)).

- Capa convolucional: crea un mapa de características para predecir las probabilidades de clase de cada característica aplicando un filtro que escanea toda la imagen, unos pocos píxeles cada vez, como se puede ver en la Figura 2.21**.**

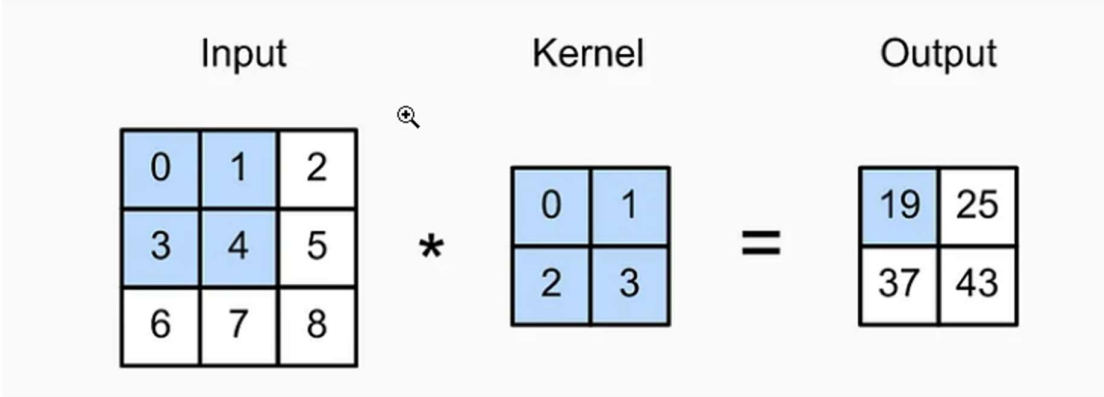

Figura 2.21 La operación de convolución (imagen tomada de (A. Zhang et al., 2019)

- Capa de agrupamiento (pooling): reduce la cantidad de información que la capa convolucional generó para cada característica y mantiene la información más esencial (el proceso de las capas convolucional y de agrupamiento suele repetirse varias veces). En la Figura 2.22 podemos ver el resultado de aplicar la función max con un tamaño 2x2.

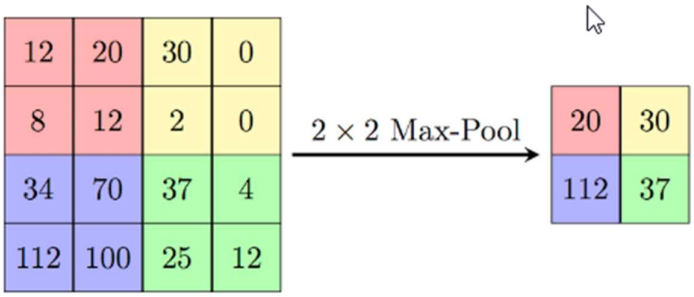

Figura 2.22 Capa de agrupamiento aplicando función max. (imagen tomada de (Jauregui, 2020)

- Capa de entrada totalmente conectada: "aplana" las salidas generadas por las capas anteriores para convertirlas en un único vector que pueda utilizarse como entrada para la capa siguiente.
- Capa totalmente conectada: aplica pesos sobre la entrada generada por el análisis de características para predecir una etiqueta precisa.
- Capa de salida totalmente conectada: genera los valores finales para determinar una clase para la imagen.

2.4.5 Arquitecturas CNN

2.4.5.1 U-Net

Esta arquitectura de red neuronal fue inicialmente publicada en 2015 por Olaf Ronneberger, Philipp Fischer y Thomas Brox (Ronneberger et al., 2015a).

En este trabajo se basaron en una arquitectura “totalmente convolucional" (Long et al., 2015). Modificaron y ampliaron esta arquitectura para que funcionara con muy pocas imágenes de entrenamiento y produjese segmentaciones más precisas. La idea principal de (Long et al., 2015) consiste en complementar una red convolucional habitual con capas sucesivas, en las que los operadores de *pooling* se sustituyen por operadores de *upsampling*. El nombre de U-Net viene de la forma de U que adopta la red en su representación gráfica, como muestra la Figura 2.23. Por lo tanto, estas capas aumentan la resolución de la salida. Para localizar, las características de alta resolución se utiliza una ruta de contracción compuesta de capas convolucionales y de reducción de muestreo (usualmente capas de max-pooling). Después de pasar por la ruta de contracción, la red U-Net utiliza una "ruta de expansión" o "ruta de sobremuestreo" para aumentar nuevamente la resolución espacial de la representación intermedia. Esto se logra mediante capas de convolución transpuesta. La combinación de estas dos rutas (contracción y expansión) es fundamental en U-Net y permite que las características de alta resolución obtenidas en la ruta de contracción se combinen con la información de la ruta de expansión para obtener una salida que conserve tanto los detalles finos como las características de alto nivel de la imagen de entrada.

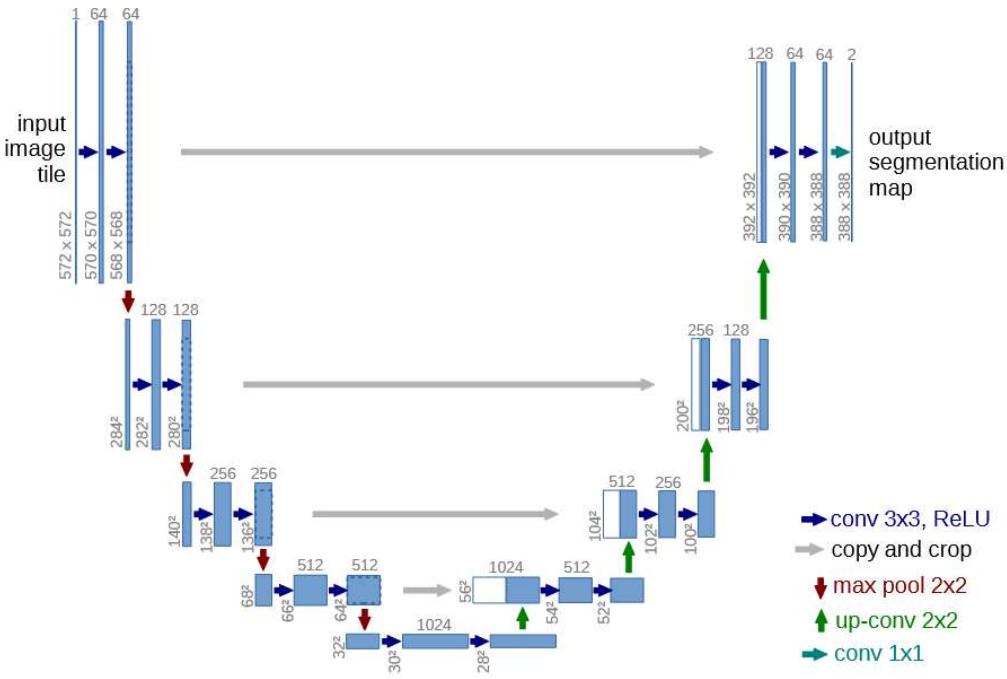

Figura 2.23 Arquitectura U-net. Cada recuadro azul corresponde a un mapa de características multicanal. El número de canales se indica en la parte superior del recuadro. El tamaño x-y se indica en el borde inferior izquierdo del recuadro. Los recuadros blancos representan mapas de características copiadas. Las flechas indican las distintas operaciones (imagen tomada de (Ronneberger et al., 2015a)).

Una modificación importante de la arquitectura U-Net es que en la parte de muestreo ascendente o ruta de expansión utiliza también un gran número de canales de características, que permiten a la red propagar la información de contexto a capas de mayor resolución. Como consecuencia, la ruta expansiva es más o menos simétrica a la ruta de contracción, y da lugar a una arquitectura en forma de U. La red no tiene capas totalmente conectadas y sólo utiliza la parte válida de cada convolución, es decir, el mapa de segmentación sólo contiene los píxeles cuyo contexto completo está disponible en la imagen de entrada. Esta estrategia permite segmentar sin fisuras imágenes de tamaño arbitrario mediante una estrategia de superposición de mosaicos. Para predecir los píxeles de la región fronteriza de la imagen, el contexto que falta se extrapola reflejando la imagen de entrada. Esta estrategia de mosaico es importante para aplicar la red a imágenes de gran tamaño, ya que de otro modo la resolución se vería limitada por la memoria de la GPU (Ronneberger et al., 2015a).

2.4.5.2 LadderNet

U-Net y sus variantes en la literatura tienen toda una estructura de codificador-decodificador. Sin embargo, el número de caminos para el flujo de información en U-Net es limitado. Por eso Juntang Zhuang (Zhuang, 2019) propuso LadderNet, una red neuronal convolucional multirramal para la segmentación semántica, como se muestra en la Figura 2.24, que tiene más rutas de flujo de información. Las características en diferentes escalas espaciales se nombran con letras de la A

a la E, y las columnas se nombran con números del 1 al 4. Las columnas 1 y 3 se denominan ramas codificadoras, y a las columnas 2 y 4 ramas decodificadoras (Zhuang, 2019).

Utilizamos la convolución con un intervalo de 2 para pasar de características de campo receptivo pequeño a características de campo receptivo grande (por ejemplo, de A a B), y utilizamos la convolución transpuesta con un intervalo de 2 para pasar de características de campo receptivo grande a características de campo receptivo grande (por ejemplo, de B a A). El número de canales se duplica de un nivel al siguiente (por ejemplo, de A a B)(Zhuang, 2019).

LadderNet puede considerarse como una cadena de U-Nets. Las columnas 1 y 2 pueden considerarse una U-Net, y las columnas 3 y 4, otra U-Net. Entre dos U-Nets, hay conexiones de salto en los niveles A-D. A diferencia de las redes en U, en las que las características de las ramas codificadoras se concatenan con las características de las ramas decodificadoras, en Laddernet se suman las características de dos ramas. En la Figura 2.24 se muestra una LadderNet compuesta por 2 U-Nets, pero se pueden añadir más para formar estructuras de red complicadas (Zhuang, 2019).

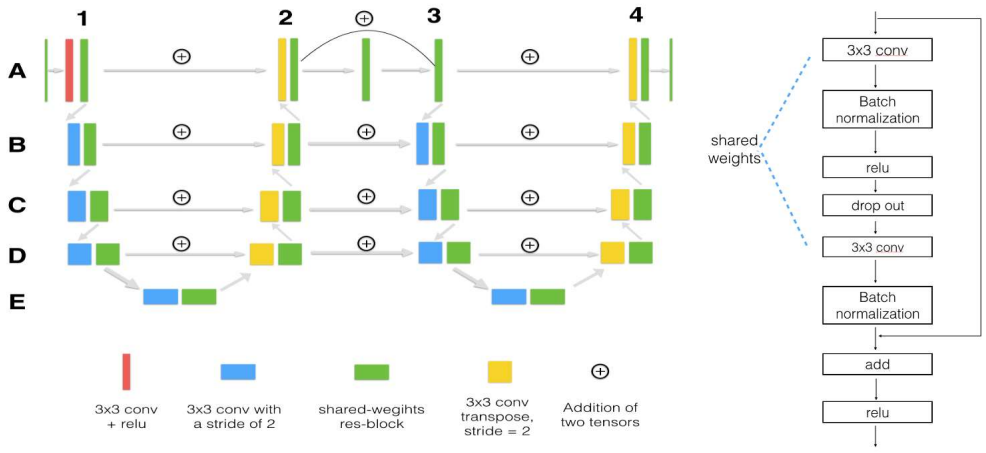

Figura 2.24 Arquitectura LadderNet propuesta por Juntang Zhuang (Zhuang, 2019)

La adición de ramas codificador-decodificador aumenta el número de parámetros y la dificultad del entrenamiento. Para resolver este problema, se propusieron los bloques residuales de pesos compartidos, que se muestran en la Figura 2.24. A diferencia del bloque convolucional residual estándar, propuesto por (He et al., 2015), las dos capas convolucionales del mismo bloque comparten los mismos pesos. Como en la red neuronal convolucional recurrente (RCNN) (Alom et al., 2017), las dos capas convolucionales del mismo bloque pueden verse como una capa recurrente, con la diferencia de que las dos capas de normalización de lotes son diferentes. Se añade una capa de abandono entre dos capas convolucionales para evitar el sobreajuste. El bloque residual de pesos compartidos combina la fuerza de la conexión de salto, la convolución

recurrente y la regularización de abandono, y tiene muchos menos parámetros que un bloque residual estándar (Zhuang, 2019).

2.4.5.3 ResUnet

En 2018 Zhang et al propusieron una red ResUnet profunda (Zhang et al., 2018).

La segmentación semántica tiene el problema de que, para obtener un resultado más fino, es muy importante utilizar detalles de bajo nivel conservando la información semántica de alto nivel. Sin embargo, entrenar una red neuronal profunda de este tipo es muy difícil, especialmente cuando sólo se dispone de muestras de entrenamiento limitadas. Una posible solución para este problema es emplear una red preentrenada y luego ajustarla con el conjunto de datos objetivo. Otra solución es emplear una amplia ampliación de datos. Además del aumento de datos, la arquitectura de U-Net también contribuye a aliviar el problema del entrenamiento. La intuición detrás de esto es que la copia de características de bajo nivel a los correspondientes niveles altos en realidad crea un camino para la propagación de la información, permitiendo que las señales se propaguen entre los niveles bajos y altos de una manera mucho más fácil, lo que no sólo facilita la propagación hacia atrás durante el entrenamiento, sino que también compensa los detalles más finos de bajo nivel a las características semánticas de alto nivel. (Z. Zhang et al., 2018)

La profundización mejoraría el rendimiento de una red neuronal multicapa, pero podría dificultar el entrenamiento, produciendo un problema de degradación. Para superar estos problemas, He et al. (He et al., 2015) propusieron la red neuronal residual, que consiste en una serie de unidades residuales apiladas.

Hay múltiples combinaciones de normalización por lotes (BN), activación ReLU y capas convolucionales en una unidad residual. He et al. (He et al., 2016) presentaron una discusión detallada sobre los impactos de las diferentes combinaciones en y sugirieron un diseño de pre-activación completa. (Kalaivani & Seetharaman, 2022)

Zhang et al propusieron una red ResUnet profunda, que es una red neuronal de segmentación semántica que combina los puntos fuertes de la U-Net y de la red neuronal residual. Esta combinación aporta dos ventajas: 1) la unidad residual facilita el entrenamiento de la red; 2) las conexiones de salto dentro de una unidad residual y entre los niveles bajos y altos de la red facilitan la propagación de la información sin degradación, haciendo posible el diseño de una red neuronal con muchos menos parámetros que, sin embargo, podría alcanzar un rendimiento comparable o incluso mejor en la segmentación semántica. Estos autores utilizaron una arquitectura de 7 niveles de profundidad (Z. Zhang et al., 2018) (cf. Figura 2.25).

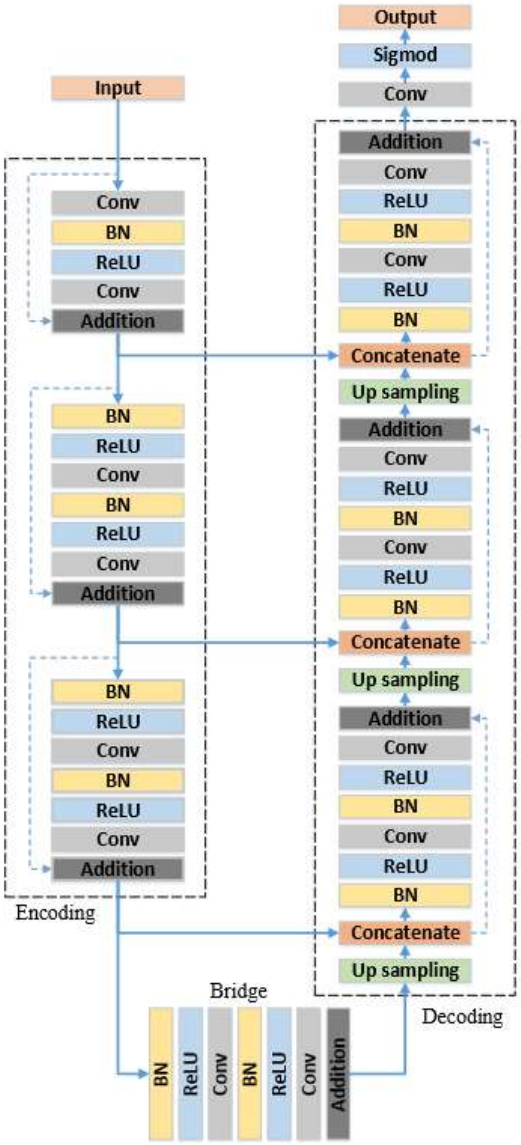

Figura 2.25 Arquitectura ResUnet profunda (Zhang et al., 2018).

Esta red ResUnet consta de tres partes: codificación, puente y decodificación. La primera parte codifica la imagen de entrada en representaciones compactas. La última parte recupera las representaciones para realizar una categorización por píxeles, es decir, una segmentación semántica. La parte central sirve de puente entre la codificación y la descodificación (Z. Zhang et al., 2018).

Las tres partes se construyen con unidades residuales que constan de dos bloques de convolución de 3 × 3 y un mapeo de identidad. Cada bloque de convolución incluye una capa BN, una capa de activación ReLU y una capa convolucional. El mapeo de identidad conecta la entrada y la salida de la unidad. La ruta de codificación tiene tres unidades residuales. En cada unidad, en lugar de utilizar la operación de *pooling* para reducir el tamaño del mapa de características, se aplica una

zancada (*stride*) de 2 al primer bloque de convolución para reducir el mapa de características a la mitad. En consecuencia, la ruta de descodificación también se compone de tres unidades residuales. Antes de cada unidad, se realiza un muestreo ascendente de los mapas de características del nivel inferior y una concatenación con los mapas de características de la ruta de codificación correspondiente. Después del último nivel de la ruta de descodificación, se utiliza una convolución 1 × 1 y una capa de activación sigmodal para proyectar los mapas de características multicanal en la segmentación deseada. En total tiene 15 capas convolucionales, frente a las 23 capas de U-Net. Cabe señalar que el recorte que realiza U-Net es innecesario en esta red (Z. Zhang et al., 2018).

2.4.6 Modelos preentrenados

Los modelos preentrenados en aprendizaje automático son modelos que se han entrenado en un gran conjunto de datos utilizando una arquitectura y un objetivo específicos, y que luego se han guardado para que otros puedan utilizarlos en diversas tareas sin tener que empezar el proceso de entrenamiento desde cero. Estos modelos están diseñados para aprender características y representaciones generales a partir de los datos en los que se han entrenado, que luego pueden afinarse o adaptarse para tareas o dominios específicos.

Existen varios modelos preentrenados populares en distintos dominios del aprendizaje automático:

Procesamiento del Lenguaje Natural (PLN):

- BERT (Bidirectional Encoder Representations from Transformers): es un modelo basado en transformadores preentrenado con una cantidad masiva de datos de texto. Comprende las relaciones contextuales del lenguaje y puede ajustarse para tareas como el análisis de sentimientos, el reconocimiento de entidades con nombre, etc.
- GPT (Generative Pretrained Transformer): es una serie de modelos basados en transformadores diseñados para tareas de generación de texto. El GPT-3, por ejemplo, es conocido por su capacidad para generar textos coherentes y contextualmente relevantes.

Visión por ordenador:

- VGG (Visual Geometry Group): Arquitectura de red neuronal convolucional profunda preentrenada con un gran conjunto de datos de imágenes. Suele utilizarse para tareas de clasificación de imágenes.
- ResNet (Red Residual): Una arquitectura profunda que introduce bloques residuales, lo que facilita el entrenamiento de redes muy profundas. También se utiliza para la clasificación de imágenes y tareas relacionadas.

Reconocimiento del habla:

- DeepSpeech: Un modelo preentrenado para la conversión de voz a texto. Se entrena en grandes conjuntos de datos de habla transcrita para convertir con precisión el lenguaje hablado en texto.

Aprendizaje por transferencia y ajuste:

El aprendizaje por transferencia consiste en tomar un modelo previamente entrenado y adaptarlo a una tarea diferente pero relacionada. Esto resulta especialmente útil cuando se dispone de pocos datos para la nueva tarea.

El ajuste fino consiste en entrenar un modelo previamente entrenado en un conjunto de datos más pequeño y específico para la tarea en cuestión, lo que ayuda al modelo a ajustar sus características aprendidas para adaptarse mejor a la nueva tarea.

El uso de modelos preentrenados puede ahorrar mucho tiempo y recursos informáticos, ya que la fase inicial de aprendizaje captura patrones y características generales de los datos. Este conocimiento puede aprovecharse para una amplia gama de tareas sin necesidad de empezar el proceso de formación desde cero.

2.4.6.1 VGG

VGG es una red neuronal convolucional profunda propuesta por Karen Simonyan y Andrew (Simonyan & Zisserman, 2014). VGG es el acrónimo del nombre de su grupo, Visual Geometry Group, de la Universidad de Oxford. Este modelo obtuvo el segundo puesto en la competición ILSVRC-2014, donde se logró un 92,7% de rendimiento en la clasificación. El modelo VGG investiga la profundidad de capas con un tamaño de filtro convolucional muy pequeño (3 × 3) para tratar imágenes a gran escala. Los autores publicaron una serie de modelos VGG con diferentes longitudes de capa, de 11 a 19 (Le, 2021).

En resumen (Le, 2021):

- Todas las configuraciones de VGG tienen estructuras de bloques.
- Cada bloque de VGG consiste en una secuencia de capas convolucionales a las que sigue una capa de max-pooling. En todas las capas convolucionales se aplica el mismo tamaño de núcleo (3 × 3). Además, los autores utilizaron un tamaño de relleno de 1 para mantener el tamaño de la salida después de cada capa convolucional. También se aplica un max-pooling de tamaño 2 × 2 con strides de 2 para reducir a la mitad la resolución después de cada bloque.

- Cada modelo VGG tiene dos capas ocultas totalmente conectadas y una capa de salida totalmente conectada.

La estructura de VGG16 se describe en la figura Figura 2.26:

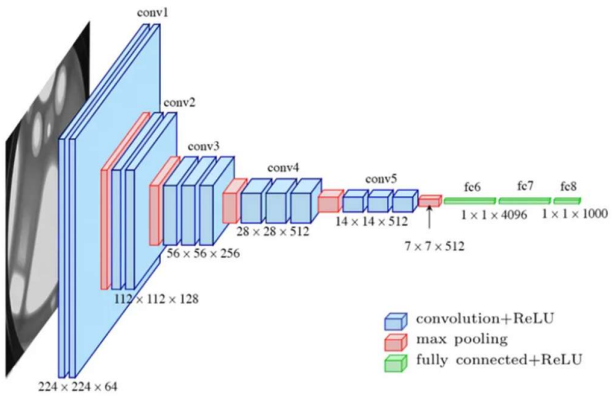

Figura 2.26 Arquitectura de VGG (Sugata & Yang, 2017).

2.4.7 Aumento de datos

La precisión de los modelos de aprendizaje profundo depende en gran medida de la calidad, la cantidad y el significado contextual de los datos de entrenamiento (Shah, 2022). Sin embargo, la escasez de datos es uno de los retos más comunes en la construcción de modelos de aprendizaje profundo. En los casos de uso de producción, recopilar esos datos puede ser costoso y llevar mucho tiempo.

El aumento de datos es un proceso de incremento artificial de la cantidad de datos mediante la generación de nuevos puntos de datos a partir de datos existentes. Este aumento de datos puede llevarse a cabo con un doble propósito:

- Mejora el rendimiento del modelo: Al aumentar el tamaño del conjunto de datos de entrenamiento, el modelo puede obtener un mejor rendimiento en datos nuevos o no vistos previamente. Esto es especialmente útil cuando el conjunto de datos original es pequeño o limitado, lo que puede llevar a problemas de sobreajuste si se utilizan modelos más complejos.
- Mejora la robustez del modelo: Al introducir diferentes variaciones y transformaciones en los datos de entrenamiento, el modelo se vuelve más resistente y adaptable a diversas condiciones y cambios en los datos de prueba. Estos son particularmente beneficioso cuando se espera que el modelo funcione en entornos reales donde los datos pueden tener cierta variabilidad y ruido.

Cabe preguntarse cuál es la diferencia entre datos aumentados y datos sintéticos.

- Datos sintéticos: Cuando los datos se generan artificialmente sin utilizar imágenes del mundo real. Los datos sintéticos suelen generarse mediante redes generativas antagónicas (*generative adversarial networks*, GANs).
- Datos aumentados: Derivados de imágenes originales con algún tipo de transformaciones geométricas menores (como volteo, traslación, rotación o adición de ruido) con el fin de aumentar la diversidad del conjunto de entrenamiento.

Estas son algunas de las razones por las que las técnicas de aumento de datos han ido ganando popularidad en los últimos años.

- Mejora el rendimiento de los modelos de ML: Los métodos de aumento de datos se utilizan ampliamente en prácticamente todas las aplicaciones de aprendizaje profundo, como la detección de objetos, la clasificación de imágenes, el reconocimiento de imágenes, la comprensión del lenguaje natural, la segmentación semántica y mucho más. Los datos aumentados están mejorando el rendimiento y los resultados de los modelos de aprendizaje profundo al generar instancias nuevas y diversas para los conjuntos de datos de entrenamiento.
- Reduce los costes operativos relacionados con la recopilación de datos: La recopilación y el etiquetado de datos pueden ser procesos largos y costosos para los modelos de aprendizaje profundo. Las empresas pueden reducir los gastos operativos transformando los conjuntos de datos mediante técnicas de aumento de datos.

2.5 Técnicas de filtrado de imágenes

El filtrado es una técnica que permite resaltar algunas de las características de una imagen. Existen varias técnicas de filtrado de imágenes utilizadas en el procesamiento de imágenes para mejorar la calidad, reducir el ruido o resaltar características específicas.

El proceso de filtrado consiste en la aplicación a cada uno de los píxeles de la imagen de una matriz de filtrado de tamaño NxN (generalmente de 3x3, aunque puede ser mayor) compuesta por números enteros y que genera un nuevo valor mediante una función del valor original y los de los pixeles circundantes. El resultado final se divide entre un escalar, generalmente la suma de los coeficientes de ponderación. Los filtros se pueden expresar mediante la siguiente ecuación (Universidad de Murcia, 2005):

$$ND'_{i,j} = \frac{ND_{i-1,j-1} + ND_{i,j-1} + ND_{i+1,j-1} + ND_{i-1,j} + ND_{i,j} + ND_{i+1,j} + ND_{i-1,j+1} + ND_{i-1,j+1} + ND_{i-1,j+1}}{9} \quad (2.1)$$

donde *i* y *j* representan la fila y la columna de cada pixel, ND_{ij} su Nivel Digital y ND'_{ij} el Nivel Digital obtenido tras hacer el filtrado.

Los filtros más utilizados son los de paso bajo, de paso alto, los filtros direccionales y los de detección de bordes.

2.5.1 Filtros de paso bajo

Un filtro de paso bajo se trata de una técnica de filtrado utilizada en el procesamiento de imágenes para atenuar las altas frecuencias de la imagen y preservar las características de baja frecuencia. Su principal objetivo es suavizar la imagen al eliminar el ruido de alta frecuencia y resaltar las características de bajo contraste.

Este tipo de filtro se basa en la idea de que las altas frecuencias en una imagen representan cambios rápidos en los valores de los píxeles, como los bordes y detalles finos, mientras que las bajas frecuencias corresponden a áreas suavemente transicionales o regiones uniformes.

El resultado de aplicar este tipo de filtros es una imagen suavizada, donde los detalles finos y los bordes se ven menos pronunciados. Esto puede ser útil en casos donde se desea reducir el ruido de alta frecuencia, como el ruido impulsivo o el ruido generado por sensores.

Algunos ejemplos de este tipo de filtros son:

Filtro de media: Este filtro reemplaza el valor de cada píxel por la media de los valores de los píxeles vecinos. La máscara utilizada en este filtro suele ser de tamaño cuadrado o rectangular. Este filtro suaviza la imagen y elimina el ruido de alta frecuencia.

Filtro de mediana: En este filtro, el valor de cada píxel se reemplaza por el valor mediano de los píxeles vecinos. Se emplea para eliminar el ruido impulsivo o "sal y pimienta". Es especialmente efectivo para preservar los bordes y detalles finos en comparación con el filtro de media.

Filtro de media ponderada: En este filtro, se asignan pesos diferentes a los píxeles vecinos en función de su distancia al píxel central. Los píxeles más cercanos tienen más peso que los píxeles más alejados. Este filtro suaviza la imagen y reduce el ruido de alta frecuencia.

Filtro gaussiano: Este filtro aplica una convolución con una máscara gaussiana a la imagen. La máscara tiene una forma de campana, con los coeficientes más altos en el centro y disminuyendo hacia los bordes. El filtro gaussiano suaviza la imagen y reduce el ruido de alta frecuencia.

Filtro de suavizado bilateral: Este filtro suaviza la imagen manteniendo los bordes. Combina la información de intensidad de los píxeles vecinos ponderada por la similitud de intensidad y distancia espacial. Es útil para reducir el ruido mientras se preservan los detalles y los bordes.

2.5.2 Filtros de paso alto

Su objetivo es resaltar las zonas de mayor variabilidad eliminando lo que sería la componente media, precisamente la que detectan los filtros de paso bajo, consiguiendo de esta manera aumentar la nitidez de la imagen. Un filtro de paso alto es la base de la mayoría de los métodos de enfoque. Una imagen es más nítida cuando aumenta el contraste entre zonas contiguas con poca variación de brillo u oscuridad. Un filtro de paso alto tiende a retener la información de alta frecuencia dentro de una imagen mientras reduce la información de baja frecuencia. El núcleo del filtro de paso alto está diseñado para aumentar el brillo del píxel central en relación con los píxeles vecinos. La matriz del kérnel suele contener un único valor positivo en su centro, que está completamente rodeado de valores negativos.

2.5.3 Filtros direccionales

Los filtros direccionales se tratan de una técnica de filtrado utilizado en el procesamiento de imágenes para detectar estructuras que siguen una determinada dirección en el espacio, resaltando el contraste entre los píxeles situados a ambos lados de la estructura.

2.5.4 Filtros para la detección de bordes

El objetivo de los filtros de detección de bordes en procesamiento de imágenes y visión por computadora es identificar los límites o contornos de objetos presentes en una imagen. Los bordes representan transiciones significativas en la intensidad de los píxeles y son puntos donde la imagen experimenta un cambio brusco.

Los filtros de detección de bordes se aplican a una imagen para resaltar estas transiciones de intensidad y permitir que los algoritmos de visión por computadora y análisis de imágenes identifiquen y segmenten objetos en función de sus contornos.

Se ha aplicado para extraer los contornos de ecocardiografías (Hussein et al., 2011).

Uno de los más utilizados es el detector de bordes de Sobel, que realza la variación entre filas y columnas. EL filtro Sobel puede ser utilizado para extraer contornos. El filtro Sobel se compone de dos núcleos o kérnels: uno para la detección de bordes verticales y otro para la detección de bordes horizontales. Estos núcleos son matrices pequeñas que se aplican a la imagen original mediante una operación de convolución.

2.5.5 Filtros de aumento de contraste

2.5.5.1 CLAHE

EL filtro de Ecualización Adaptativa del Histograma con Contraste Limitado (CLAHE) se utiliza frecuentemente para aumentar el contraste en las imágenes de ultrasonidos (Singh et al., 2020).

Consideremos una imagen cuyos valores de píxel se limitan únicamente a un rango específico de valores. Una imagen más brillante tiene un histograma como el de la izquierda en la Figura 2.27.

CLAHE "estira el histograma" haciendo que se parezca más al de la derecha. Esto generalmente aumenta el contraste de la imagen. CLAHE aplica una cuadrícula a la imagen y luego realiza la ecualización del histograma en cada celda. La amplificación del contraste está limitada por un factor llamado "límite de recorte", para reducir el problema de la amplificación del ruido(Fernandez de Toro Espejel, 2023)

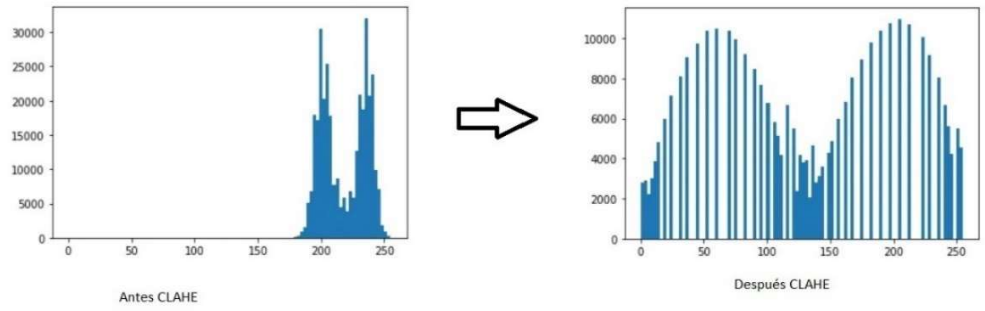

Figura 2.27 Ejemplo de histograma antes de ecualización con CLAHE (izda) y después (drcha.) (Senaratne, 2020).

2.5.5.2 Ecualización del histrograma

La ecualización del histograma se trata de un método que mejora el contraste de una imagen, con el fin de ampliar el rango de intensidades. (Huamán, 2022). En la imagen de la Figura 2.28 de la izquierda se puede ver que los píxeles parecen agrupados en torno al centro del rango disponible de intensidades. Lo que hace la ecualización de histograma es ampliar este rango. Tras aplicar la ecualización, obtenemos un histograma como el de la figura de la derecha.

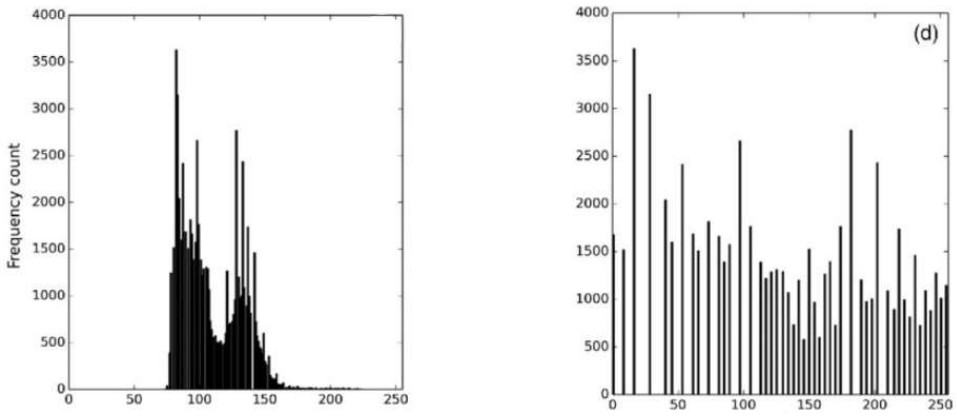

Figura 2.28 Ejemplo histograma antes de ecualización de histograma (izda) y después (drcha.) (imagen tomada de (Toet & Wu, 2014)).

2.6 Métricas de evaluación

Al igual que otros aspectos del aprendizaje profundo, la evaluación depende del tipo de problema que se esté abordando. Para el caso que nos ocupa, la segmentación multiclase de imágenes, se

pueden aplicar las métricas típicas empleadas en clasificación, entre las que destaca la exactitud u otras también derivadas de una matriz de confusión, como son precisión, cobertura, curva ROC o el área bajo la curva (AUC).

Esto es así ya que la segmentación de imágenes se puede entender como una aplicación de clasificación de píxeles, en la que cada píxel de una imagen toma un valor que corresponde a una categoría o clase a predecir.

En nuestros caso de estudio, un píxel puede clasificarse de 4 maneras:

1. ventrículo izquierdo,
2. miocardio,
3. aurícula izquierda y
4. otros.

Para cada imagen, la evaluación consiste en comparar la predicción de cada píxel con su correspondiente etiquetado por un experto humano (*ground truth*), considerándose acierto si ambos valores coinciden y fallo, en caso contrario. Así, es posible calcular las métricas y generar las herramientas gráficas descritas anteriormente.

Dos de las métricas de evaluación más populares para medir el solapamiento de regiones son el coeficiente de similaridad DICE, y el coeficiente de Jaccard, también llamado coeficiente IoU (I*ntersection over union*). Ambos permiten medir la exactitud con la que superponen dos conjuntos.

Se muestra a continuación la visualización de su cálculo (véase Figura 2.29 y Figura 2.30) y su correspondiente fórmula.

Coeficiente DICE:

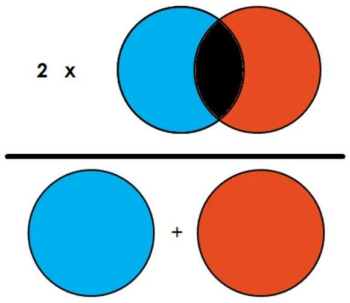

Figura 2.29 Visualización del cálculo del coeficiente DICE (Tiu, 2022).

$$DSC(A,B) = \frac{2|A \cap B|}{|A|+|B|}$$

Coeficiente IoU o de Jaccard:

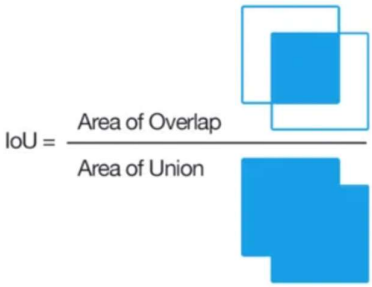

Figura 2.30 Visualización del cálculo de IoU (Wikipedia, 2023).

$$J(A,B) := \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

En el contexto de la segmentación, lo que se mide es la superposición de la predicción sobre el etiquetado real. Concretamente, la superposición de los píxeles uno a uno.

El coeficiente DICE no depende del número relativo de pixeles de cada clase, lo cual resulta muy indicado para problemas con clases desbalanceadas, como es el caso que nos ocupa.

Respecto al coeficiente Jaccard o coeficiente IoU, no se considera idóneo para problemas de aprendizaje automático con clases desbalanceadas, aunque sí que se considera como una medida realista de la bondad de un modelo de segmentación. Esto resulta lógico, ya que este coeficiente IoU mantiene cierta equivalencia con el coeficiente DICE, en tanto que crece y decrece en su misma dirección.

La siguiente fórmula muestra la relación entre ambos coeficientes:

$$J(A,B)^{-1} + 1 = 2DSC(A,B)^{-1}$$

$$DSC(A,B) = \frac{2J(A,B)}{1+J(A,B)}$$

2.6.1 Funciones de pérdida en segmentación semántica

Existe una gran variedad de funciones de pérdida que resultan convenientes para resolver los problemas de segmentación semántica en el entrenamiento de CNNs. Un buen resumen puede encontrase en (Jadon, 2020), que identifica las funciones de pérdida más comunes e idóneas en

segmentación de imágenes. La Tabla 2.1 resume las funciones de perdida más relevantes/utilizadas a la hora de realizar la segmentación semántica, junto con sus casos de uso.

Tabla 2.1 Funciones de pérdida para segmentación semántica (Jadon, 2020)

Función de pérdida	Casos de uso
Entropía cruzada binaria	Funciona mejor en escenarios de distribución equitativa de datos entre clases función de pérdida basada en la distribución Bernoulli.
Entropía cruzada ponderada	Muy utilizada con conjuntos de datos sesgados Pondera los ejemplos positivos mediante el coeficiente β
Entropía cruzada equilibrada	Similar a la entropía cruzada ponderada, muy utilizada con conjuntos de datos sesgados pondera tanto los ejemplos positivos como los negativos mediante β y 1 – β respectivamente
Pérdida focal	Funciona mejor con conjuntos de datos muy desequilibrados. Pondera a la baja la contribución de los ejemplos fáciles, lo que permite al modelo aprender ejemplos difíciles
Término de penalización por pérdida derivado del mapa de distancias	Variante de la entropía cruzada. Se utiliza para límites difíciles de segmentar
Pérdida DICE	Inspirada en el coeficiente DICE. Como el coeficiente DICE no es convexo por naturaleza resulta complicada su optimización en algoritmos de entrenamiento de redes neuronales. Se ha modificado para hacerlo más manejable, Esta versión modificada se llama pérdida DICE suavizada y emplea funciones sigmoideas para ser más adecuada para la optimización.
Pérdida de sensibilidad-especificidad	Inspirada en las métricas de sensibilidad y especificidad Se utiliza en los casos en los que hay que centrarse más en los verdaderos positivos
Pérdida de Tversky	Variante del coeficiente DICE. Añade peso a los falsos positivos y a los falsos negativos.
Pérdida Log-Cosh DICE	Variante de DICE Loss y enfoque log-cosh de regresión inspirado para suavizar. Las variaciones pueden utilizarse para conjuntos de datos sesgados
Pérdida por distancia de Hausdorff	Inspirada en la métrica de distancia de Hausdorff utilizada para evaluar la segmentación. La pérdida aborda la naturaleza no convexa de la métrica de distancia añadiendo algunas variaciones
Pérdida basada en la forma	Variación de la pérdida de entropía cruzada añadiendo un coeficiente basado en la forma. Se utiliza en casos de límites difíciles de segmentar.
Pérdida combinada	Combinación de pérdida de dados y entropía cruzada binaria se utiliza para clases ligeramente desequilibradas aprovechando las ventajas de BCE y pérdida DICE.
Pérdida logarítmica exponencial	Función combinada de pérdida de dados y entropía cruzada binaria. Se centra en los casos predichos con menos precisión

Pérdida por similitud estructural maximizada por correlación	Se centra en la estructura de segmentación. Se utiliza en casos de importancia estructural, como las imágenes médicas.

Como puede apreciarse en dicha tabla, muchas de estas funciones se basan en modificaciones o combinaciones de otras. Sus diferencias son sutiles y resultarán más o menos indicadas dependiendo del problema concreto a resolver y de las posibles características del conjunto de datos abordado.

Capítulo 3 Estado del arte

El análisis de imágenes ecocardiográficas 2D desempeña un papel crucial en la rutina clínica para medir la morfología y la función cardiacas y llegar a un diagnóstico. Dicho análisis se basa en la interpretación de índices clínicos que se extraen del procesamiento de imágenes de bajo nivel, como la segmentación. Por ejemplo, la extracción de la fracción de eyección (FE) del ventrículo izquierdo (VI) requiere delineación precisa del endocardio ventricular izquierdo tanto en fin de diástole (ED) y fin de sístole (ES). En la rutina clínica la anotación semiautomática o manual sigue siendo un trabajo cotidiano debido a la falta de precisión y reproducibilidad de los métodos de totalmente automáticos. Esto lleva a tareas que consumen mucho tiempo y son propensas a la variabilidad intraobservador e interobservador (Leclerc, Smistad, Pedrosa, Østvik, Cervenansky, Espinosa, Espeland, Rye Berg, et al., 2019).

Las dificultades inherentes a la segmentación de imágenes ecocardiográficas están bien documentadas: i) escaso contraste entre el miocardio y la sangre; ii) falta de homogeneidad del brillo; iii) variación en el patrón de moteado a lo largo del miocardio debido a la orientación de la sonda cardíaca con respecto al tejido; iv) presencia de trabéculas y músculos papilares con intensidades similares a las del miocardio; v) importante variabilidad de la ecogenicidad tejido dentro de la población; vi) variabilidad de la forma, intensidad y vi) movimiento de las estructuras patológicas.(Leclerc, Smistad, Pedrosa, Østvik, Cervenansky, Espinosa, Espeland, Rye Berg, et al., 2019)

EL primer conjunto de datos ecocardiográficos se publicó dentro del Challenge on Endocardial Three-dimensional Ultrasound Segmentation (CETUS), que tuvo lugar durante la conferencia MICCAI 2014. El conjunto de datos CETUS se compone de 45 secuencias ecocardiográficas 3D (15 para entrenamiento, 30 para pruebas) distribuidas equitativamente entre tres subgrupos diferentes: sujetos sanos, pacientes con infarto de miocardio previo examinados al menos 3 meses después del evento y pacientes con miocardiopatía dilatada. Los datos se proporcionan con dos mallas de referencia del endocardio del ventrículo izquierdo (LVEndo) por paciente (una en ED y otra en ES), correspondiendo cada referencia a la forma media calculada a partir de las anotaciones de tres cardiólogos experimentados diferentes. En este desafío se evaluaron cinco métodos totalmente automáticos (modelos deformables, bosque aleatorio de Hough, filtro de Kalman, modelo de apariencia activa) y cuatro semiautomáticos (método de corte de grafos, bosque aleatorio estructurado, enfoques multiatlas y de conjunto de niveles). Ningún participante implementó una red neuronal profunda. El resultado del desafío reveló que las mejores puntuaciones generales las obtuvo la superficie activa explícita B-spline, un método totalmente automático propuesto por Barbosa et al. (Barbosa et al., 2014). Este método fue mejorado

posteriormente por Pedrosa et al. (Pedrosa et al., 2017)) gracias a la integración de una forma a priori (*shape prior*) obtenida mediante un esquema convencional de análisis de componentes principales. Al hacerlo, los autores obtuvieron las siguientes puntuaciones para la segmentación del endocardio del ventrículo izquierdo 3D: i) valores DICE medios de 0,909 (ED) y 0,875 (ES); ii) distancias de Hausdorff medias de 6,3 mm (ED) y 6,9 mm (ES) y iii) distancias absolutas medias de 1,8 mm (ED) y 2,0 mm (ES) (Leclerc, Smistad, Pedrosa, Østvik, Cervenansky, Espinosa, Espeland, Rye Berg, et al., 2019).

Se han propuesto varios estudios de métodos de segmentación ecocardiográfica mediante técnicas tradicionales de aprendizaje no profundo, tanto en 2D (Noble & Boukerroui, 2006), (Carneiro et al., 2012) como en 3D (Bernard et al., 2016), (Leung & Bosch, 2010). La mayoría de los métodos reportados se enfocaron en la segmentación del borde del endocardio del ventrículo izquierdo. Entre esas revisiones, solo la de Bernard et al. (2016) aplicó diferentes técnicas al mismo conjunto de datos, lo que permitió una comparación justa (Bernard et al., 2016). En este estudio, los autores enumeraron los resultados obtenidos por nueve técnicas diferentes. Los métodos reportados se pueden dividir en dos categorías principales: aquellos con una información previa débil y aquellos con una información previa fuerte. El primer grupo implica suposiciones débiles, relativas a información espacial, de intensidad, de movimiento o anatómica. Incluye técnicas basadas en imágenes (filtro de cuadratura multiescala) (Wang et al., 2014), un método basado en el movimiento (filtro de Kalman) (Smistad & Lindseth, 2014), modelos deformables (BEAS, level-set) (Barbosa et al., 2014), (Wang et al., 2014) y un enfoque basado en gráficos (*graph-cut*) (Bernier et al., 2014). El segundo grupo utiliza enfoques con información previa fuerte, como un conocimiento previo de forma (bosque de Hough) (Milletari et al., 2014), un modelo de apariencia activa (van Stralen et al., 2014), un método basado en atlas (Oktay et al., 2014) y un algoritmo de aprendizaje automático (bosque aleatorio) (Milletari et al., 2014), (Keraudren et al., 2014), (Domingos et al., 2014), cada uno de los cuales requiere un conjunto de entrenamiento anotado manualmente (Leclerc, Smistad, Pedrosa, Østvik, Cervenansky, Espinosa, Espeland, Rye Berg, et al., 2019).

Los métodos de aprendizaje profundo se han aplicado con éxito a la segmentación del endocardio del ventrículo izquierdo en ecocardiografía. En 2012, Carneiro et al. Desarrollaron un método de aprendizaje profundo en dos etapas para la segmentación del endocardio del ventrículo izquierdo para imágenes ecocardiográficas 2D restringidas a adquisiciones en vista de cuatro cámaras (Carneiro et al., 2012). Basándose en un marco máximo a posteriori, los autores formularon el problema de segmentación del ventrículo izquierdo (VI) mediante dos pasos sucesivos: i) la selección automática de varias regiones en la imagen analizada donde el endocardio del ventrículo izquierdo está totalmente presente; ii) la extracción automática del contorno del endocardio del ventrículo izquierdo a partir de las regiones previamente seleccionadas. Estos dos

pasos implicaban una red de creencia profunda. El método se entrenó con 400 imágenes de 12 secuencias de pacientes con diversas patologías y se probó con 50 imágenes de 2 secuencias de sujetos sanos. Obtuvieron una distancia Hausdorff media de ∼18 mm y una distancia absoluta media de ∼8 mm para el endocardio del ventrículo izquierdo . En 2017, Smistad et al. (Smistad et al., 2017) demostraron que el método U-Net CNN (Ronneberger et al., 2015a) podía entrenarse para segmentar con éxito el ventrículo izquierdo en imágenes de ultrasonido 2D. Sin embargo, debido a la falta de datos, la red se entrenó con la salida de un método de segmentación de modelo deformable de última generación (Smistad & Lindseth, 2014). En un conjunto de pruebas segmentadas manualmente, los resultados mostraron que la red y el modelo deformable obtuvieron la misma precisión, con una puntuación DICE de 0,87. Posteriormente, Oktay et al. (Oktay et al., 2018) utilizaron CNNs para segmentar la estructura 3D del endocardio del ventrículo izquierdo un enfoque denominado red neuronal anatómicamente restringida (ACNN). El núcleo de su red neuronal se basaba en una arquitectura similar a la U-Net 3D (Çiçek et al., 2016), cuya salida de segmentación está restringida para ajustarse a una representación compacta no lineal de la anatomía subyacente derivada de una red autocodificadora. El rendimiento de su método se evaluó en el conjunto de datos CETUS. Obtuvieron las siguientes puntuaciones para la segmentación de la estructura 3D del endocardio del ventrículo izquierdo: i) valores DICE medios de 0,912 (ED) y 0,873 (ES); ii) distancias Hausdorff medias de 7,0 mm (ED) y 7,7 mm (ES) y iii) distancias absolutas medias de 1,9 mm (ED) y 2,1 mm (ES) (Oktay et al., 2018), que son bastante parecidos a los obtenidos por Pedrosa et al. (Pedrosa et al., 2017). Además, el uso de solo 15 pacientes durante la fase de entrenamiento ilustra el gran potencial de las técnicas de aprendizaje profundo para analizar imágenes ecocardiográficas (Leclerc, Smistad, Pedrosa, Østvik, Cervenansky, Espinosa, Espeland, Rye Berg, et al., 2019).

Con el éxito del aprendizaje profundo en numerosos campos de la imagen médica, varios investigadores explotaron la aplicación de métodos basados en el aprendizaje profundo en la segmentación ecocardiográfica. Inspirándose en la eficacia de la información previa en los algoritmos tradicionales, se han incorporado conocimiento previo de forma (Oktay et al., 2018) y atlas (Dong et al., 2020) en marcos de aprendizaje profundo para proporcionar conocimiento de la estructura anatómica, y lograron mejoras alentadoras en comparación con los modelos sin información previa (Liu et al., 2021). También se han desarrollado métodos que combinan el aprendizaje profundo con modelos deformables para la segmentación de estructuras cardíacas en dos pasos (Veni et al., 2018) y en un paso (Nascimento & Carneiro, 2020). Los patrones de movimiento en secuencias de ultrasonido cardíaco se han explorado mediante filtrado de partículas (Carneiro & Nascimento, 2013) y flujo óptico (Jafari et al., 2018) para mantener la coherencia temporal para una segmentación consistente y precisa en diferentes marcas de tiempo. Además, varios investigadores se han centrado en la utilización de datos no etiquetados

2012; (Yu et al., 2018) y multidominio (H. Chen et al., 2016) en la fase de entrenamiento para reducir el requisito de modelos basados en aprendizaje profundo para grandes conjuntos de datos de imágenes médicas (Liu et al., 2021).

A pesar de estos esfuerzos, quedan dos grandes problemas sin resolver en el campo de la segmentación en ecocardiografía. En primer lugar, el bajo contraste entre los tejidos del miocardio y la pérdida de bordes son comunes en la ecocardiografía 2D. Por lo tanto, se requiere un diseño de modelo específico para mejorar las características de las regiones de bajo contraste basándose en contextos vecinos y reduciendo al mismo tiempo el impacto negativo de los ruidos. En segundo lugar, los métodos actuales de segmentación basados en aprendizaje profundo suelen predecir la categoría para cada píxel de forma independiente, es decir, la predicción para un píxel se realiza sin considerar explícitamente otros resultados de predicción de píxeles vecinos. Por lo tanto, carecen del mecanismo de aprendizaje para la coherencia de etiquetas entre diferentes ubicaciones en una imagen ecocardiográfica 2D, lo que probablemente sea subóptimo y reduzca la calidad de la segmentación (L.-C. Chen et al., 2018) (Liu et al., 2021).

Capítulo 4 Materiales y métodos

4.1 Descripción general

La preparación de los experimentos explicados a continuación se ha desarrollado en varias fases:

- Selección de arquitecturas de redes CNN que se iban a entrenar con el dataset disponible.
- Selección de los preprocesados que se iban a realizar a las imágenes contenidas en el dataset.
- Desarrollo del código correspondientes a las arquitecturas y métricas
- Desarrollo del código correspondiente al preprocesado de imágenes y generación de diferentes datasets que contenían las imágenes preprocesadas.
- Entrenamiento de las arquitecturas con los datos sin tratar y posteriormente extenderlo a los dataset con las imágenes preprocesadas.

4.2 Entorno computacional y tecnología

En el desarrollo del proyecto se han utilizado tres entornos computaciones distintos:

- Equipo personal: En este equipo se han ejecutado las partes de código correspondientes a los diferentes preprocesados realizados en las imágenes y se han generado los datasets con las imágenes preprocesadas en él.
- Google Colab: Se utilizó una cuenta gratuita para realizar los desarrollos de los modelos y métricas. Al intentar hacer los entrenamientos aparecían constantemente mensajes indicando que se habían gastado los recursos de GPU y TPU para ese periodo, por lo que fue necesario solicitar acceso a los servidores de la UNED.

- Servidor UNED: Este servidor dispone de una GPU NVIDIA GeForce RTX 2070 SUPER, además de 52 GB de RAM. En este servidor se ha realizado el entramiento de los modelos con los diferentes preprocesados.

4.3 Datasets

CAMUS son las siglas de « Cardiac Acquisitions for Multi-structure Ultrasound Segmentation ». Contiene secuencias ecocardiográficas 2D con vistas de dos y cuatro cámaras de 500 pacientes que se adquirieron con el mismo equipo en el mismo centro médico. Para cada paciente se exportaron secuencias de vistas apicales 2D de cuatro cámaras y de dos cámaras. Estas vistas cardíacas estándar se eligieron para este estudio para permitir la estimación de los valores de fracción de eyección del ventrículo izquierdo basados en el método biplano de discos de Simpson.

Se adquirió al menos un ciclo cardiaco completo para cada paciente en cada vista, lo que permitió la anotación manual de las estructuras cardiacas en ED (fin de diástole) y ES (fin de sístole). Cabe destacar que dentro del dataset CAMUS únicamente encontramos máscaras para las imágenes ES (final de sístole) y ED (final de diástole).

El tamaño de este conjunto de datos y su estrecha conexión con cuestiones clínicas cotidianas ofrecen la posibilidad de entrenar métodos de aprendizaje profundo para analizar automáticamente datos ecocardiográficos. Además, CAMUS incluye anotaciones manuales de expertos para el endocardio del ventrículo izquierdo (LVEndo), el miocardio (contorno del epicardio más específicamente, denominado LVEpi) y la aurícula izquierda (LA). El objetivo de este conjunto de datos clínicos es (Leclerc, Smistad, Pedrosa, Østvik, Cervenansky, Espinosa, Espeland, Rye Berg, et al., 2019):

- permitir entrenar adecuadamente modelos de aprendizaje automático / profundo;
- permitir una comparación significativa entre los métodos más avanzados;
- evaluar hasta dónde pueden llegar los métodos de aprendizaje supervisado en la evaluación de imágenes ecocardiográficas 2D, es decir, segmentar estructuras cardiacas y estimar índices clínicos.

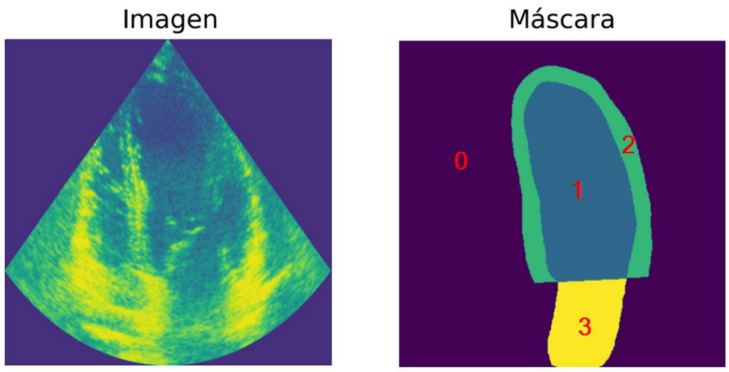

Figura 4.1 Segmentación multiestructura vista como un problema de clasificación multietiqueta. Imagen ecocardiográfica 2D con las estructuras a la izquierda y la máscara de verdad básica a la derecha. 1- Ventrículo izquierdo, 2- Miocardio, 3- Aurícula izquierda,0-otros

En el conjunto CAMUS se emplean cuatros etiquetas para diferenciar las estructuras cardiacas. Cada etiqueta está representada por un número del 0 al 3: Ventrículo izquierdo (1), miocardio (2), aurícula izquierda (3), otros (0). (véase Figura 4.1).

Para reforzar el realismo clínico de los datos del *dataset* por parte de los autores no se establecieron prerrequisitos ni realizaron selección de datos. En consecuencia:

- algunos casos fueron difíciles de rastrear
- el conjunto de datos presenta una amplia variabilidad de ajustes de adquisición;
- para algunos pacientes, partes de la pared no eran visibles en las imágenes;

- en algunos casos, la recomendación de orientación de la sonda para obtener una vista rigurosa de cuatro cámaras fue imposible de seguir y, en su lugar, se obtuvo una vista de cinco cámaras

De este modo se obtuvo un conjunto de datos muy heterogéneo, tanto en términos de calidad de imagen como de casos patológicos, lo que es típico de los datos de la práctica clínica diaria. Para mantener el realismo clínico el dataset contiene imágenes de buena, media y baja calidad. Alrededor de un 19% de las imágenes tienen baja calidad.

Los datos de los 500 pacientes del dataset se han dividido en conjunto de entrenamiento, compuesto por datos de 450 pacientes, y conjunto de prueba, que contiene datos de los 50 restantes. En la Figura 4.2 se muestran ejemplos de imágenes extraídas del conjunto de datos CAMUS.

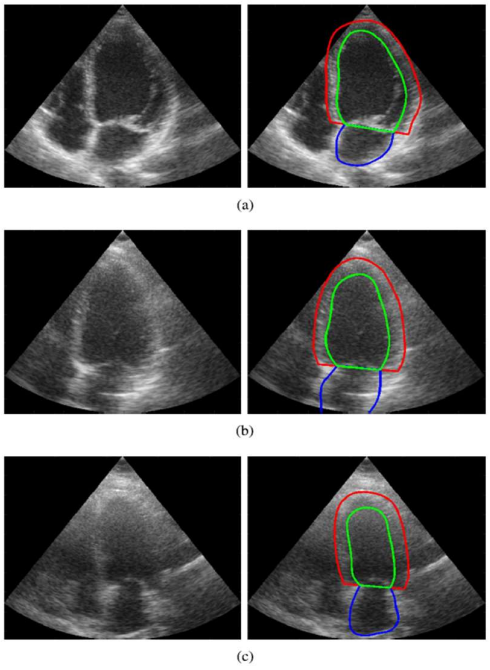

Figura 4.2 Imágenes típicas extraídas del conjunto de datos CAMUS. Endocardio y epicardio del ventrículo izquierdo y la pared de la aurícula izquierda se ven en verde, rojo y azul, respectivamente. En la parte izquierda se ven las imágenes de entrada, y en la parte derecha las anotaciones manuales correspondientes. (a) Buena calidad de imagen. (b) Calidad media de imagen. (c) Mala calidad de imagen. (Leclerc, Smistad, Pedrosa, Østvik, Cervenansky, Espinosa, Espeland, Rye Berg, et al., 2019)

Para realizar el entrenamiento del modelo, he utilizado el dataset CAMUS. Dentro de sus vistas he empleado la apical de 4 cámaras y he empleado las máscaras correspondientes a fin de diástoles (ED).

4.3.1 Preparación de los datos

Como se ha comentado anteriormente, el conjunto de datos se encontraba divido en 450 imágenes de entrenamiento y 50 imágenes de prueba. Además de esta división se dividió el

conjunto de imágenes entrenamiento en dos subconjuntos: entrenamiento y validación, aplicando un porcentaje del 80%, resultando en un conjunto de entrenamiento de 360 imágenes y conjunto de validación de 90.

Además, el tamaño de las imágenes y de las máscaras se redujo de 778×549 a 384×384. Para ello he transformado la imagen en un array y posteriormente he redimensionado este array a 384x384 empleando interpolación bicúbica mediante la librería openCV.

4.4 Arquitecturas y metodologías de evaluación

En esta sección se describen las cuatro arquitecturas propuestas para la segmentación de las imágenes:

- **U-Net:** He creado una red U-Net con la arquitectura mostrada en la Figura 2.23.
- **ResUNet:** He creado una red ResUNET con la arquitectura mostrada en la Figura 2.25.
- **LadderNet:** He creado una red Laddenet con la arquitectura mostrada en la Figura 2.24.
- **CNN:** He desarrollado una arquitectura de red convolucional similar a U-Net, como se muestra en la Figura 4.3, pero más sencilla, que ha servido para validar los diferentes preprocesados realizados en las imágenes y que ha servido a su vez para poder validar todos los desarrollos de código realizados. Me ha servido como punto de partida dado para poder ir desarrollando las arquitecturas más complejas que luego he utilizado.

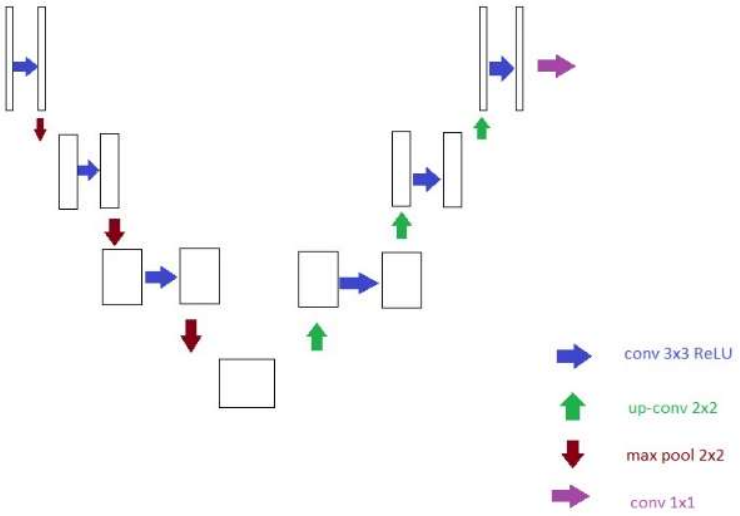

Figura 4.3 Arquitectura CNN desarrollada.

- **VGG16:** He creado una arquitectura de red UNET empleando en la parte del codificador una red VGG16 preentrenada con el conjunto de imágenes ImageNet (ver Figura 4.4):

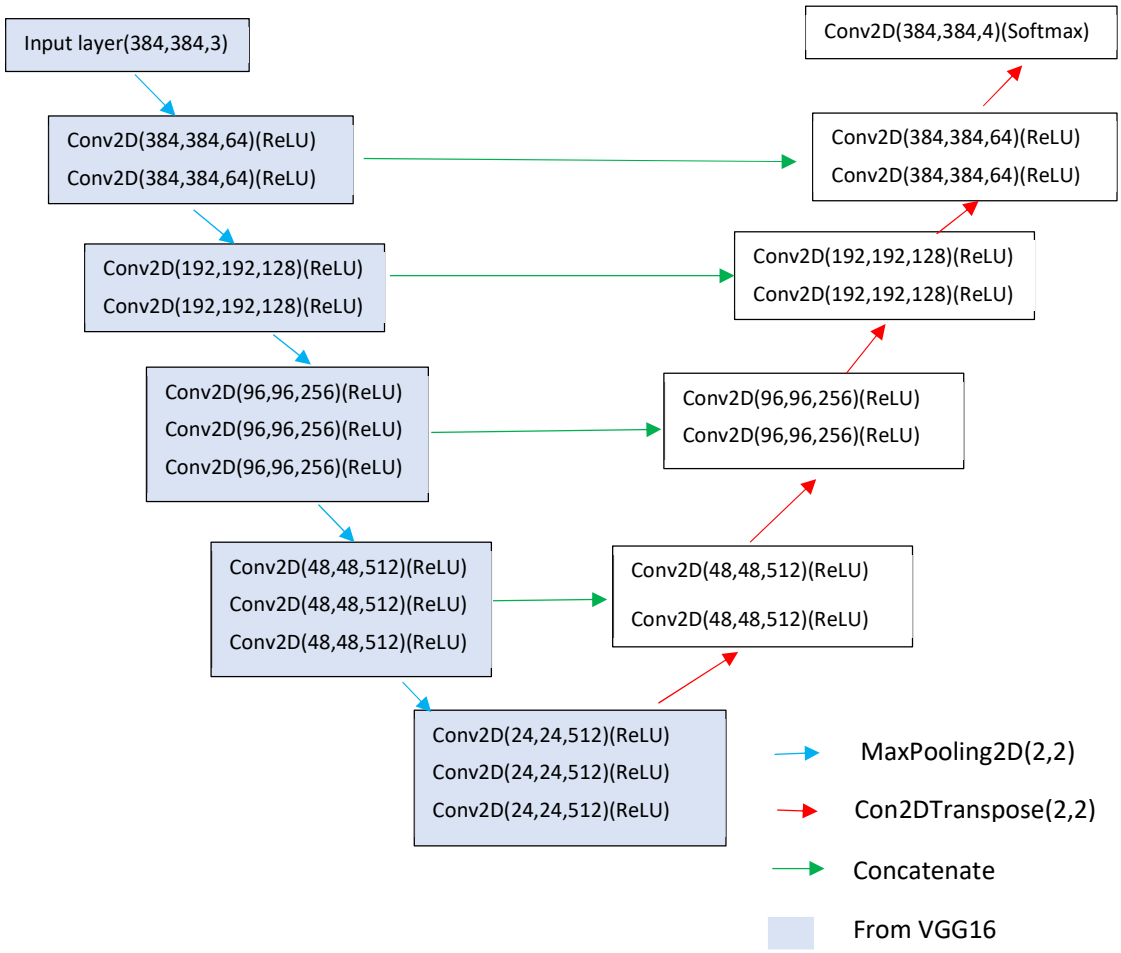

Figura 4.4 Arquitectura UNET preentrenada con codificador VGG16.

Para comparar los diferentes modelos he usado las métricas Accuracy, IoU y coeficiente DICE.

4.5 Experimentos

4.5.1 Tamaño de Kérnel

He probado a ejecutar las arquitecturas descritas anteriormente con 3 tamaños de kérnel diferentes: 3x3,4x4 y 2x2.

4.5.2 Preprocesado

A continuación, se presentan varios filtros de preprocesado aplicados a las imágenes con el objetivo de mejorar el rendimiento de los modelos.

4.5.2.1 CLAHE

Con este experimento queríamos ver si la Ecualización Adaptativa del Histograma con Contraste Limitado (CLAHE) mejoraba el rendimiento. He usado una cuadrícula de 8x8 pixels y un límite de recorte de 2. En la Figura 4.5, se muestra un ecocardiograma antes y después de aplicar la ecualización con CLAHE del histograma.

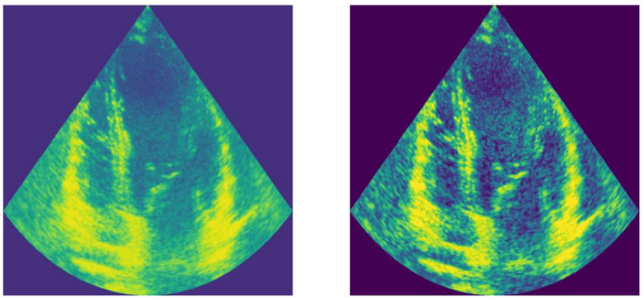

Figura 4.5 Imagen antes CLAHE (izda.), después CLAHE (drcha).

4.5.2.2 Ecualización del histograma

En la Figura 4.6, se muestra un ecocardiograma antes y después de aplicar la ecualización del histograma.

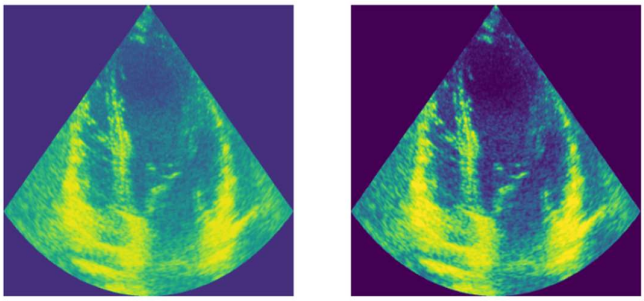

Figura 4.6 Imagen antes filtro Ecualización Histograma(izda.), después Ecualización Histograma (drcha).

4.5.2.3 Filtro gaussiano

En las pruebas he aplicado un filtro gaussiano de tamaño 7x7. En la Figura 4.7, se muestra un ecocardiograma antes y después de aplicar un filtro gaussiano.

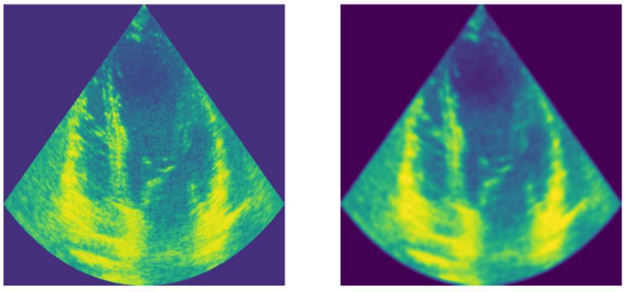

Figura 4.7 Imagen antes filtro Gaussiano(izda.), después filtro Gaussiano (drcha).

4.5.2.4 Filtro Sobel

He hecho dos pruebas, a aplicar el filtro de Sobel con tamaño de kérnel de 5 solo en el eje X y a aplicarlo con tamaño de kérnel de 5 en ambos ejes. En la Figura 4.8 se muestra un ecocardiograma antes y después de aplicar el filtro de Sobel en el eje X.

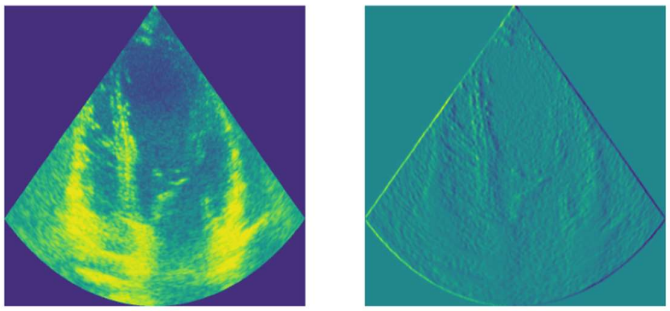

Figura 4.8 Imagen antes filtro Sobel(izda.), después filtro Sobel (drcha) solo eje X.

En la Figura 4.9 se muestra un ecocardiograma antes y después de aplicar el filtro de Sobel en ambos ejes.

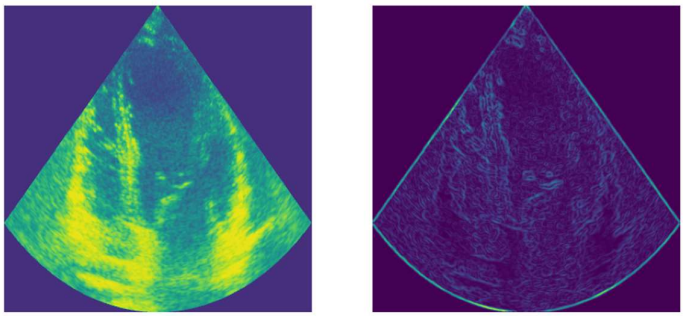

Figura 4.9 Imagen antes filtro de Sobel (izda.), después filtro Sobel(drcha) en ambos ejes.

4.5.2.5 Aumento de datos

La idea del aumento de datos es aplicar diversas transformaciones sobre las entradas originales, obteniendo muestras ligeramente diferentes pero iguales en esencia, lo que permite a la red desenvolverse mejor en la fase de inferencia.

He utilizado la función ImageDataGenerator de Keras para realizar estas transformaciones con la siguiente configuración: giro de ángulo de 10º, movimiento vertical y horizontal de 0.1, distorsión de 0.2 y zoom de 0.2. En la Figura 4.10Figura 4.10 se puede ver el resultado de aplicar diferentes transformaciones a una imagen y su máscara:

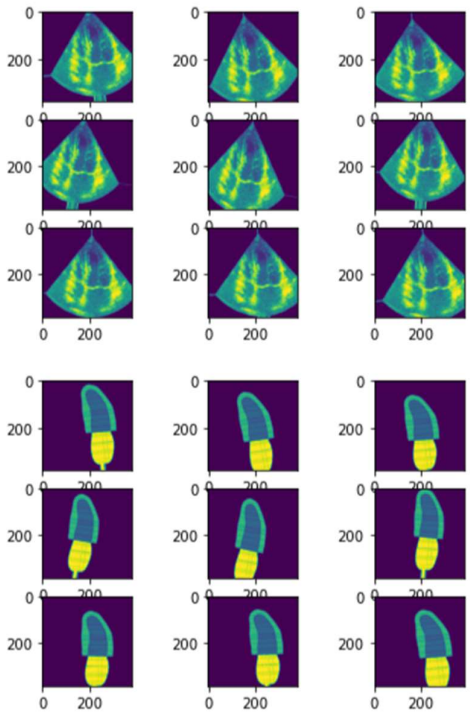

Figura 4.10 Ejemplo de resultado aumento de datos aplicando 9 transformaciones a una imagen original.

4.6 Experimentación

En los experimentos he combinado todas las arquitecturas con todos los preprocesados de imágenes mencionados, así como con aumento de datos. También he probado las arquitecturas con diferentes tamaños de kérnel 2, 3 y 4. Los resultados se exponen en el apartado 5.

He entrenado el modelo usando 150 épocas, pero para evitar excesivo tiempo de entrenamiento he utilizado EarlyStopping, con una paciencia, entendida esta como el número de épocas o iteraciones completas adicionales que se permite que el modelo continúe entrenando después de que una métrica de validación haya dejado de mejorar antes de detener el entrenamiento, de 5 épocas sin mejora, monitorizando la métrica del coeficiente DICE. He empleado un tamaño de lote (*batch*) de 5.

He utilizado una tasa de aprendizaje inicial de 1e-3, aunque he aplicado ReduceLROnPlateau para reducir la tasa de aprendizaje cuando la métrica del coeficiente DICE no mejora durante dos épocas.

Como función de pérdida he utilizado sparse_categorical_crossentropy. En nuestro caso vemos que las clases estás desbalanceadas y por ello hemos empleado funciones de pérdida que penalizan en mayor medida los fallos en las clases minoritarias.

Como optimizador de la función de pérdida he utilizado Adam (Adaptive Moment Estimation), un método estocástico de descenso de gradiente que utiliza una estimación adaptativa del momento(Kingma & Ba, 2014).

4.7 Evaluación con otros conjuntos de ecocardiogramas

Una vez concluida la parte de experimentación anteriormente descrita he elegido el modelo (con o sin procesado) que mejor resultado obtiene para CAMUS y lo he evaluado con otros conjuntos de ecocardiogramas.

El dataset elegido se trata del *dataset* TED, que tiene secuencias de ecocardiografías de 4 cámaras en dos dimensiones de 98 pacientes (Painchaud et al., 2023).

Para cada paciente dispone de una carpeta en la de que además de las secuencias con diferentes imágenes (*frames*) y sus máscaras disponemos de un fichero .cfg. Este fichero .cfg nos da información de la posición donde se encuentra la imagen ED, ES y el número de imágenes por cada paciente.

Para realizar la evaluación he organizado la información en 3 conjuntos de datos, uno que contiene todos los frames de las secuencias, uno que únicamente corresponde al ED y otro que únicamente contiene los frames correspondientes al ES.

Capítulo 5 Resultados

5.1 Experimentos entrenamiento y validación

A continuación, se muestran los resultados con los conjuntos de datos de entrenamiento y validación para el conjunto CAMUS.

5.1.1 U-Net

Tabla 5.1 Resultados de los conjuntos de entrenamiento y validación con la arquitectura U-Net.

	Entrenamiento			Validación		
	DICE Mean (std)	IOU Mean (std)	Accuracy Mean (std)	DICE Mean (std)	IOU Mean (std)	Accuracy Mean (std)
Kérnel3	0,908 (0,073)	0,839 (0,102)	0,963 (0,043)	0,771 (0,238)	0,676 (0,260)	0,823 (0,264)
Kérnel4	0,905 (0,064)	0,832 (0,093)	0,964 (0,029)	0,773 (0,201)	0,667 (0,227)	0,870 (0,162)
Kérnel2	0,840 (0,080)	0,733 (0,109)	0,940 (0,037)	0,666 (0,207)	0,532 (0,211)	0,842 (0,158)
Aumento datos	**0,919 (0,050)**	**0,854 (0,075)**	**0,970 (0,022)**	**0,858 (0,144)**	**0,772 (0,166)**	**0,923 (0,135)**
CLAHE	0,918 (0,067)	0,855 (0,094)	0,967 (0,037)	0,757 (0,216)	0,650 (0,245)	0,834 (0,219)
Equalized	0,912 (0,058)	0,844 (0,084)	0,966(0,029)	0,830 (0,140)	0,728 (0,162)	0,912 (0,134)
Gaussiano	0,900 (0,062)	0,825 (0,088)	0,961 (0,034)	0,734 (0,272)	0,637 (0,282)	0,819 (0,261)
Sobel eje X	0,915 (0,059)	0,849 (0,087)	0,967 (0,030)	0,851 (0,100)	0,751 (0,121)	0,928 (0,094)
Sobel ambos ejes	0,908 (0,069)	0,839 (0,095)	0,964 (0,035)	0,834 (0,153)	0,737 (0,163)	0,909 (0,162)

5.1.2 U-Net preentrenada

Tabla 5.2 Resultados de los conjuntos de entrenamiento y validación con la arquitectura U-Net preentrenada.

	Entrenamiento			Validación		
	DICE Mean (std)	IOU Mean (std)	Accuracy Mean (std)	DICE Mean (std)	IOU Mean (std)	Accuracy Mean (std)
Kérnel3	0,855 (0,065)	0,752 (0,086)	0,947 (0,025)	0,671 (0,347)	0,585 (0,314)	0,898 (0,099)
Aumento datos	**0,867 (0,038)**	**0,768 (0,051)**	**0,953 (0,013)**	**0,816 (0,192)**	**0,718 (0,179)**	**0,939 (0,050)**
CLAHE	0,819 (0,070)	0,699 (0,085)	0,934 (0,028)	0,667 (0,272)	0,548 (0,241)	0,891 (0,068)
Equalized	0,866 (0,065)	0,769 (0,089)	0,952 (0,023)	0,746 (0,270)	0,649 (0,257)	0,916 (0,073)
Gaussiano	0,023 (0,056)	0,015 (0,036)	0,760 (0,003)	0,0 (0,0)	0,0 (0,0)	0,740 (0,0)
Sobel eje X	0,853 (0,099)	0,756 (0,127)	0,949 (0,034)	0,781 (0,189)	0,670 (0,187)	0,923 (0,051)
Sobel ambos ejes	0,838 (0,085)	0,729 (0,103)	0,941 (0,032)	0,775 (0,171)	0,656 (0,171)	0,924 (0,043)

5.1.3 ResUnet

Tabla 5.3 Resultados de los conjuntos de entrenamiento y validación con la arquitectura ResUnet.

	Entrenamiento			Validación		
	DICE Mean (std)	IOU Mean (std)	Accuracy Mean (std)	DICE Mean (std)	IOU Mean (std)	Accuracy Mean (std)
Kérnel3	0,547 (0,338)	0,447 (0,284)	0,867 (0,077)	0,331 (0,319)	0,244 (0,253)	0,718 (0,206)
Kérnel4	0,839 (0,101)	0,734 (0,123)	0,942 (0,036)	0,711 (0,282)	0,607 (0,264)	0,892 (0,116)
Kérnel2	0,769 (0,203)	0,662 (0,191)	0,924 (0,053)	0,621 (0,325)	0,519 (0,301)	0,846 (0,175)
Aumento datos	0,734 (0,119)	0,608 (0,130)	0,955 (0,022)	0,640 (0,215)	0,522 (0,205)	0,888 (0,216)
CLAHE	0,860 (0,076)	0,761 (0,096)	0,949 (0,028)	0,791 (0,194)	0,686 (0,201)	0,923 (0,068)
Equalized	0,724 (0,313)	0,640 (0,284)	0,917 (0,074)	0,616 (0,347)	0,522 (0,315)	0,842 (0,189)
Gaussiano	**0,845 (0,113)**	**0,745 (0,139)**	**0,942 (0,043)**	**0,717 (0,266)**	**0,613 (0,265)**	**0,840 (0,224)**
Sobel eje X	0,644 (0,360)	0,557 (0,318)	0,898 (0,082)	0,579 (0,373)	0,489 (0,324)	0,879 (0,085)

Sobel ambos ejes	0,567 (0,362)	0,472 (0,308)	0,874 (0,081)	0,513 (0,374)	0,419 (0,310)	0,864 (0,077)

5.1.4 CNN

Tabla 5.4 Resultados de los conjuntos de entrenamiento y validación con la arquitectura CNN.

	Entrenamiento			Validación		
	DICE Mean (std)	IOU Mean (std)	Accuracy Mean (std)	DICE Mean (std)	IOU Mean (std)	Accuracy Mean (std)
Kérnel3	0,642 (0,176)	0,493 (0,151)	0,879 (0,047)	0,623 (0,170)	0,470 (0,143)	0,874 (0,043)
Kérnel4	**0,717 (0,159)**	**0,578 (0,144)**	**0,905 (0,042)**	**0,720 (0,154)**	**0,579 (0,139)**	**0,904 (0,040)**
Kérnel2	0,480 (0,170)	0,330 (0,126)	0,831 (0,038)	0,475 (0,174)	0,326 (0,129)	0,828 (0,041)
Aumento datos	0,633 (0,172)	0,484 (0,157)	0,873 (0,053)	0,649 (0,147)	0,497 (0,142)	0,876 (0,047)
CLAHE	0,660 (0,180)	0,515 (0,158)	0,886 (0,048)	0,678 (0,167)	0,532 (0,149)	0,894 (0,046)
Equalized	0,672 (0,165)	0,525 (0,149)	0,890 (0,046)	0,685 (0,111)	0,532 (0,114)	0,887 (0,043)
Gaussiano	0,511 (0,165)	0,359 (0,126)	0,822 (0,034)	0,535 (0,131)	0,375 (0,102)	0,829 (0,032)
Sobel eje X	0,725 (0,113)	0,579 (0,106)	0,906 (0,030)	0,716 (0,066)	0,563 (0,072)	0,901 (0,022)
Sobel ambos ejes	0,678 (0,088)	0,519 (0,074)	0,888 (0,021)	0,664 (0,085)	0,503 (0,069)	0,888 (0,018)

5.1.5 Laddernet

Tabla 5.5 Resultados de los conjuntos de entrenamiento y validación con la arquitectura Laddernet.

	Entrenamiento			Validación		
	DICE Mean (std)	IOU Mean (std)	Accuracy Mean (std)	DICE Mean (std)	IOU Mean (std)	Accuracy Mean (std)
Kérnel3	**0,847 (0,104)**	**0,747 (0,126)**	**0,945 (0,044)**	**0,783 (0,206)**	**0,680 (0,214)**	**0,865 (0,226)**
Kérnel4	0,871 (0,054)	0,775 (0,067)	0,954 (0,023)	0,827 (0,138)	0,723 (0,143)	0,916 (0,146)
Kérnel2	0,755 (0,112)	0,617(0,119)	0,910(0,057)	0,665(0,203)	0,529 (0,199)	0,814 (0,206)
Aumento datos	0,757 (0,088)	0,630(0,099)	0,956(0,021)	0,665(0,212)	0,547 (0,203)	0,913 (0,157)
CLAHE	0,868 (0,076)	0,775(0,097)	0,951(0,036)	0,796(0,208)	0,699 (0,215)	0,872 (0,232)

Equalized	0,840 (0,088)	0,734(0,108)	0,942(0,038)	0,727(0,246)	0,618 (0,247)	0,807 (0,283)
Gaussiano	0,868 (0,064)	0,771(0,077)	0,953(0,027)	0,849(0,120)	0,751 (0,125)	0,930 (0,130)
Sobel eje X	0,838 (0,099)	0,733(0,121)	0,940(0,043)	0,731(0,240)	0,622 (0,249)	0,838 (0,247)
Sobel ambos ejes	0,826 (0,104)	0,716(0,125)	0,939(0,042)	0,730(0,203)	0,608 (0,206)	0,849 (0,212)

5.2 Test

A continuación, se muestran los resultados obtenidos en el conjunto de datos de pruebas o test.

5.2.1 U-Net

Tabla 5.6 Resultados del conjunto de test con la arquitectura U-Net.

	DICE	IOU	Accuracy
Kérnel3	0,907	0,830	0,968
Kérnel4	0,907	0,830	0,968
Kérnel2	0,839	0,723	0,944
Aumento datos	**0,911**	**0,836**	**0,969**
CLAHE	0,857	0,751	0,943
Equalized	0,754	0,609	0,919
Gaussiano	0,834	0,716	0,943
Sobel X	0,260	0,149	0,764
Sobel ambos ejes	0,0	0,0	0,760

5.2.2 U-Net preentrenada

Tabla 5.7 Resultados del conjunto de test con la arquitectura U-Net.

	DICE	IOU	Accuracy
Kérnel3	**0,894**	**0,809**	**0,963**
Aumento datos	0,892	0,805	0,963
CLAHE	0,742	0,591	0,899
Equalized	0,815	0,703	0,928
Gaussiano	0,0	0,0	0,760

Sobel eje X	0,0	0,0	0,759
Sobel ambos ejes	0,001	0,0	0,757

5.2.3 ResUnet

Tabla 5.8 Resultados del conjunto de test con la arquitectura ResUnet.

	DICE	IOU	Accuracy
Kérnel3	0,783	0,643	0,910
Kérnel4	**0,895**	**0,810**	**0,963**
Kérnel2	0,882	0,789	0,960
Aumento datos	0,870	0,769	0,956
CLAHE	0,699	0,537	0,907
Equalized	0,764	0,618	0,910
Gaussiano	0,460	0,300	0,827
Sobel X	0, 048	0,024	0,713
Sobel ambos ejes	0,049	0,025	0,760

5.2.4 CNN

Tabla 5.9 Resultados del conjunto de test con la arquitectura CNN.

	DICE	IOU	Accuracy
Kérnel3	0,767	0,623	0,924
Kérnel4	**0,805**	**0,673**	**0,935**
Kérnel2	0,284	0,166	0,767
Aumento datos	0,789	0,652	0,930
CLAHE	0,656	0,488	0,888
Equalized	0,675	0,511	0,894
Gaussiano	0,573	0,402	0,820
Sobel X	0,021	0,010	0,733

Sobel ambos ejes	0,0	0,0	0,760

5.2.5 Laddernet

Tabla 5.10 Resultados del conjunto de test con la arquitectura Laddernet.

	DICE	IOU	Accuracy
Kérnel3	**0, 890**	**0,802**	**0,962**
Kérnel4	0,887	0,797	0,961
Kérnel2	0,826	0,705	0,942
Aumento datos	0,850	0,739	0,950
CLAHE	0,864	0,761	0,953
Equalized	0,827	0,707	0,940
Gaussiano	0,864	0,760	0,954
Sobel X	0,144	0,077	0,769
Sobel ambos ejes	0,196	0,109	0,761

5.2.6 Resumen de las 5 arquitecturas

Tabla 5.11 Tabla resumen de las 5 arquitecturas

	DICE	IOU	Accuracy
U-Net	**0,911**	**0,836**	**0,969**
U-Net preentrenada	0,894	0,809	0,963
ResUnet	0,895	0,810	0,963
CNN	0,805	0,673	0,935
Laddernet	0,890	0,802	0,962

5.2.7 Evaluación mejor modelo con otros ecocardiogramas

Tras los resultados obtenidos veo que el mejor modelo es del de la red U-Net con kérnel 3x3 y aumento de datos.

He recogido los resultados de este modelo y lo he aplicado en los 3 datasets generados a partir de los datos del dataset TED con los siguientes resultados:

Tabla 5.12 Resultados del conjunto de test TED.

	DICE	IOU	Accuracy
Dataset TED completo	0,765	0,622	0,921
Solo frames ED	**0,844**	**0,730**	**0, 938**
Solo frames ES	0,733	0,580	0,912

Adicionalmente se han analizado cualitativamente los resultados de aplicar el mejor modelo a imágenes de otros dos conjuntos de datos, uno formado por imágenes extraídas de una sonda portable Lumify en Sierra Leona y otro formado por imágenes tomadas con una sonda General Electrics s70 en el Hospital Universitario Infanta Leonor. Dado que de estos conjuntos de datos no se dispone de máscaras que permitan evaluar cuantitativamente la bondad del modelo, la evaluación que se ha realizado es únicamente cualitativa. Para realizar esta evaluación se han enviado los resultados del modelo a un experto, que ha emitido un juicio subjetivo en cuanto al desempeño del modelo. Los resultados más significativos, así como el análisis hecho por el experto se presentan en la siguiente sección.

Capítulo 6 Discusión y conclusiones

El objetivo de mi trabajo ha sido desarrollar un modelo que permita segmentar correctamente las imágenes ecocardiográficas, detectando las siguientes regiones: Ventrículo izquierdo (1), miocardio (2), aurícula izquierda (3), otros (0).

Decidimos utilizar DL para analizar las imágenes por los prometedores resultados que ha demostrado en tareas de procesamiento de imágenes y en análisis de ecocardiografía en los últimos años. Para poder realizar este proyecto han sido fundamentales los conocimientos adquiridos en la asignatura de Deep Learning cursada en el Máster, pero cabe destacar que en esta asignatura no había contenido específico relacionado con la segmentación de imágenes y para poder realizar este proyecto he debido formarme específicamente en ello.

En el desarrollo de este proyecto he empleado el dataset CAMUS dado que se trata de uno de los más amplios disponibles con ecocardiografías etiquetadas.

He comparado diferentes arquitecturas para la segmentación de imagen ecocardiográfica, así como diferentes técnicas de preprocesado de imagen.

En el conjunto de pruebas con el conjunto de imágenes CAMUS, hemos logrado un coeficiente DICE de $0{,}805$ para la arquitectura CNN con kérnel de tamaño 4x4, esta red se trata de una red neuronal convolucional, se trata de una red con una arquitectura similar a la red U-Net, pero más sencilla que me ha servido como punto de partida para validar el funcionamiento de los diferentes preprocesados realizados. He obtenido un coeficiente DICE de 0,895 para la ResUnet con kérnel 4x4, 0,890 para Laddernet con kérnel 3x3, 0,894 para una red U-Net preentrenada empleando como codificador una red VGG16 y de 0,911 para la arquitectura U-Net con tamaño de kérnel 3x3 y con aumento de datos un coeficiente DICE.

Posteriormente he aplicado mi mejor modelo, que se trata de una arquitectura U-Net con tamaño de kérnel de 3x3 y aumento de datos, al conjunto de ecocardiogramas TED (Painchaud et al., 2023), descrito en la sección 4.7 para ello he organizado los frames de este dataset, que contiene imágenes de todo el ciclo cardiaco, en 3 datasets: uno que contiene todos los frames, otro que únicamente contiene los correspondientes al final de la diástoles (ED) y otro que únicamente contiene los correspondientes al final de la sístole (ES). He evaluado mi modelo con este dataset y he podido comprobar que sus resultados son de un coeficiente DICE de 0,765 cuando incluyo todos los frames, 0,844 para los frames correspondientes a ED y 0,733 para los frames correspondientes a ES. Como era de esperar, mi modelo, que ha sido entrenado únicamente con imágenes correspondientes a ED, obtiene mejor resultado para estas imágenes.

También he aplicado el modelo a imágenes tomadas con una sonda portable y con ecógrafo de hospital. Al aplicar el modelo a las imágenes procedentes de Sierra Leona y tomadas con la sonda Lumify, podemos observar que en general el desempeño es bastante correcto, siendo la posición anatómica de las máscaras correcta en casi todos los casos. En general las máscaras cubren más espacio del debido, tomando en algunos casos parte de las paredes del corazón e incluso de las venas pulmonares. En la Figura 6.1 se puede observar un ejemplo de aplicación con las imágenes más significativas.

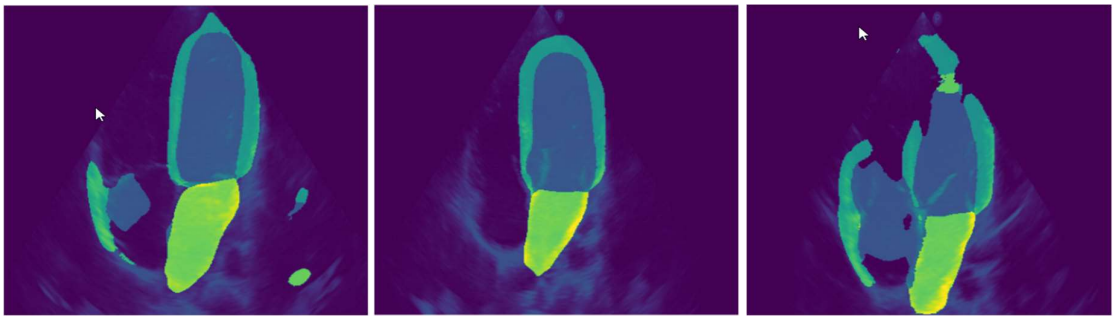

Figura 6.1 Ejemplo de tres desempeños diferentes del modelo, de izquierda a derecha, un ejemplo de desempeño medio, un ejemplo de desempeño bueno y un ejemplo de desempeño malo.

En cuanto a las imágenes obtenidas en el Hospital Universitario Infanta Leonor, se puede apreciar en la Figura 6.2 que el desempeño es bastante pobre, al segmentar incorrectamente zonas inconexas sin ningún sentido anatómico.

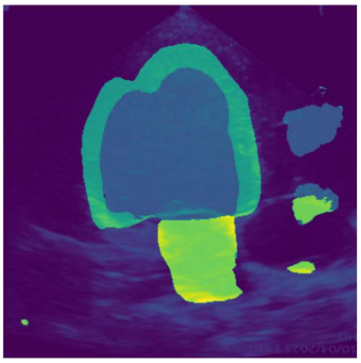

Figura 6.2 Ejemplo segmentación de imágenes obtenidas en el Hospital Universitario Infanta Leonor.

Como hemos dicho, el mejor de mis modelos, la arquitectura U-Net con tamaño de kérnel 3x3 y aplicando aumento de datos, obtiene un DICE de 0,911, que es inferior al 0,939 obtenido en (Leclerc, Smistad, Pedrosa, Østvik, Cervenansky, Espinosa, Espeland, Berg, et al., 2019). Esto podría deberse a diferencias en el preprocesamiento, ya que en mi estudio he modificado el tamaño de las imágenes a 384x384 pixeles y le he aplicado diferentes filtros. Así mismo el artículo de Leclerc et al. no detalla las arquitecturas aplicadas; tan sólo comenta que han aplicado dos arquitecturas diferentes: U-Net 1 optimizada para la velocidad y U-Net 2 optimizada para la

precisión. Esto da lugar a dos arquitecturas diferentes (que difieren de la original propuesta por Ronneberger et al. (Ronneberger et al., 2015a)) con sus propios hiperparámetros.

Mi mejor modelo no ha conseguido obtener mejores resultados que los ya existentes hoy en día en la literatura. Durante el desarrollo de este TFM he evaluado 5 arquitecturas diferentes para la segmentación de ecocardiografías del dataset CAMUS, aplicándoles diferentes filtros, aumento de datos y diferentes tamaños de kérnel. Todos los resultados indican que la red U-Net destaca por sus resultados en este tipo de tareas, también indica que el aumento de datos permite mejorar los resultados del modelo.

6.1 Trabajos futuros

Como posibles líneas de trabajo en un futuro destacaría:

- Preprocesado de imagen: Podría resultar interesante probar algún filtro adicional, sobre todo alguno de paso alto.
- Aprendizaje federado: Dado que uno de los problemas de este tipo de sistemas es disponer de pocos datos de entrenamiento y que estos se encuentren distribuidos en diferentes localizaciones, sujetos a la normativa de protección de datos, una posible área de trabajo sería el aprendizaje federado.
- Entrenamiento y evaluación: Otra posible línea de trabajo sería evaluar otras funciones de pérdida, diversos optimizadores, así como emplear validación cruzada.
- Implementar una prueba de concepto (PoC) para validar la viabilidad técnica de la solución, aunque estaba en los objetivos del proyecto no ha sido posible llevarlo a cabo.

Capítulo 7 Bibliografía

Alom, M. Z., Hasan, M., Yakopcic, C., & Taha, T. M. (2017). *Inception Recurrent Convolutional Neural Network for Object Recognition* (arXiv:1704.07709). arXiv. https://doi.org/10.48550/arXiv.1704.07709

Amazon Machine Learning. (2015). *Model Fit: Underfitting vs. Overfitting—Amazon Machine Learning*. https://docs.aws.amazon.com/machine-learning/latest/dg/model-fit-underfitting-vs-overfitting.html

Attia, D., & Benazza-Benyahia, A. (2018). Left ventricle detection in echocardiography videos. *2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, 1-6. https://doi.org/10.1109/ATSIP.2018.8364476

Barbosa, D., Friboulet, D., D'hooge, J., & Bernard, O. (2014). Fast Tracking of the Left Ventricle Using Global Anatomical Affine Optical Flow and Local Recursive Block Matching. *The MIDAS Journal*. https://doi.org/10.54294/9oybn9

Bernard, O., Bosch, J. G., Heyde, B., Alessandrini, M., Barbosa, D., Camarasu-Pop, S., Cervenansky, F., Valette, S., Mirea, O., Bernier, M., Jodoin, P.-M., Domingos, J. S., Stebbing, R. V., Keraudren, K., Oktay, O., Caballero, J., Shi, W., Rueckert, D., Milletari, F., … D'hooge, J. (2016). Standardized Evaluation System for Left Ventricular Segmentation Algorithms in 3D Echocardiography. *IEEE Transactions on Medical Imaging*, *35*(4), 967-977. https://doi.org/10.1109/TMI.2015.2503890

Bernier, M., Jodoin, P.-M., & Lalande, A. (2014). Automatized Evaluation of the Left Ventricular Ejection Fraction from Echocardiographic Images Using Graph Cut. *The MIDAS Journal*. https://doi.org/10.54294/fi9kgd

Carneiro, G., & Nascimento, J. C. (2012). The use of on-line co-training to reduce the training set size in pattern recognition methods: Application to left ventricle segmentation in

ultrasound. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 948-955. https://doi.org/10.1109/CVPR.2012.6247770

Carneiro, G., & Nascimento, J. C. (2013). Combining Multiple Dynamic Models and Deep Learning Architectures for Tracking the Left Ventricle Endocardium in Ultrasound Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(11), 2592-2607. https://doi.org/10.1109/TPAMI.2013.96

Carneiro, G., Nascimento, J. C., & Freitas, A. (2012). The segmentation of the left ventricle of the heart from ultrasound data using deep learning architectures and derivative-based search methods. *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, *21*(3), 968-982. https://doi.org/10.1109/TIP.2011.2169273

Chen, H., Zheng, Y., Park, J.-H., Heng, P.-A., & Zhou, S. K. (2016). Iterative Multi-domain Regularized Deep Learning for Anatomical Structure Detection and Segmentation from Ultrasound Images. En S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, & W. Wells (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016* (pp. 487-495). Springer International Publishing. https://doi.org/10.1007/978-3-319-46723-8_56

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(4), 834-848. https://doi.org/10.1109/TPAMI.2017.2699184

Chollet, F. (2021). *Deep Learning with Python, Second Edition*. Simon and Schuster.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation* (arXiv:1606.06650). arXiv. https://doi.org/10.48550/arXiv.1606.06650

Domingos, J. S., Stebbing, R. V., Leeson, P., & Noble, J. A. (2014). Structured Random Forests for Myocardium Delineation in 3D Echocardiography. En G. Wu, D. Zhang, & L. Zhou (Eds.),

Machine Learning in Medical Imaging (pp. 215-222). Springer International Publishing. https://doi.org/10.1007/978-3-319-10581-9_27

Dong, S., Luo, G., Tam, C., Wang, W., Wang, K., Cao, S., Chen, B., Zhang, H., & Li, S. (2020). Deep Atlas Network for Efficient 3D Left Ventricle Segmentation on Echocardiography. *Medical Image Analysis*, *61*, 101638. https://doi.org/10.1016/j.media.2020.101638

Engelman, D., Watson, C., Remenyi, B., & Steer, A. (2014). *05: Vistas del Corazón en Ecocardiografía—Parte 1* [Curso online]. Echocardiographic Diagnosis of Rheumatic Heart Disease Nurse Training Modules. https://www.wiredhealthresources.net/EchoProject/modules/spanish/05A/story_html5.html

Fernandez de Toro Espejel, B. (2023). *Application of Deep Learning techniques to the analysis of echocardiographic images for detecting damage in cardiac valves.*

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. https://learning.oreilly.com/library/view/hands-on-machine-learning/9781492032632/

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

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition [Book]. (s. f.). Recuperado 21 de mayo de 2023, de https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition* (arXiv:1512.03385). arXiv. https://doi.org/10.48550/arXiv.1512.03385

Hernández, P. (2020). *¿Qué es y para qué sirve un ultrasonido Doppler?* 409602790-DRA PATRICIA HERNÁNDEZ CORDERO. https://www.drapatriciahernandezginecologa.com/blog/articles/que-es-y-para-que-sirve-un-ultrasonido-doppler

Huamán, A. (2022). *OpenCV: Histogram Equalization*. https://docs.opencv.org/4.x/d4/d1b/tutorial_histogram_equalization.html

Hussein, Z. R., Rahmat, R. W., Abdullah, L. N., Saripan, M. I., & Zamrin, D. M. (2011). Contour extraction of echocardiographic images based on pre-processing. *IOP Conference Series: Materials Science and Engineering*, *17*, 012042. https://doi.org/10.1088/1757-899X/17/1/012042

ITelligent. (2018). Deep learning & Convolutional Neuronal Network: Qué es y en qué consiste. *ITELLIGENT INFORMATION TECHNOLOGIES*. https://itelligent.es/es/deep-learning-convolutional-neuronal-network-cnn-consiste/

Jadon, S. (2020). A survey of loss functions for semantic segmentation. *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, 1-7. https://doi.org/10.1109/CIBCB48159.2020.9277638

Jafari, M., Girgis, H. Y. A., Liao, Z., Behnami, D., Abdi, A. H., Vaseli, H., Luong, C., Rohling, R., Gin, K., Tsang, T., & Abolmaesumi, P. (2018). *A Unified Framework Integrating Recurrent Fully-Convolutional Networks and Optical Flow for Segmentation of the Left Ventricle in Echocardiography Data: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings* (pp. 29-37). https://doi.org/10.1007/978-3-030-00889-5_4

Jauregui, A. F. (2020). Cómo crear Red Convolucional en Keras. *Ander Fernández*. https://anderfernandez.com/blog/que-es-una-red-neuronal-convolucional-y-como-crearlaen-keras/

Kalaivani, S., & Seetharaman, K. (2022). A three-stage ensemble boosted convolutional neural network for classification and analysis of COVID-19 chest x-ray images. *International Journal of Cognitive Computing in Engineering*, *3*, 35-45. https://doi.org/10.1016/j.ijcce.2022.01.004

Keraudren, K., Oktay, O., Shi, W., Hajnal, J., & Rueckert, D. (2014). *Endocardial 3D Ultrasound Segmentation using Autocontext Random Forests*. https://doi.org/10.13140/2.1.2194.6889

Khoshdeli, M., Cong, R., & Parvin, B. (2017). *Detection of Nuclei in H&E Stained Sections Using Convolutional Neural Networks*.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lan, K., Liu, L., Li, T., Chen, Y., Fong, S., Marques, J., Wong, R., & Tang, R. (2020). Multi-view convolutional neural network with leader and long-tail particle swarm optimizer for enhancing heart disease and breast cancer detection. *Neural Computing and Applications*, *32*. https://doi.org/10.1007/s00521-020-04769-y

Le, K. (2021, diciembre 8). An overview of VGG16 and NiN models. *MLearning.Ai*. https://medium.com/mlearning-ai/an-overview-of-vgg16-and-nin-models-96e4bf398484

Leclerc, S., Smistad, E., Pedrosa, J., Østvik, A., Cervenansky, F., Espinosa, F., Espeland, T., Berg, E. A. R., Jodoin, P.-M., Grenier, T., Lartizien, C., D'hooge, J., Lovstakken, L., & Bernard, O. (2019). Deep Learning for Segmentation Using an Open Large-Scale Dataset in 2D Echocardiography. *IEEE Transactions on Medical Imaging*, *38*(9), 2198-2210. https://doi.org/10.1109/TMI.2019.2900516

Leclerc, S., Smistad, E., Pedrosa, J., Østvik, A., Cervenansky, F., Espinosa, F., Espeland, T., Rye Berg, E. A., Jodoin, P.-M., Grenier, T., Lartizien, C., D'hooge, J., Lovstakken, L., & Bernard, O. (2019). Deep Learning for Segmentation using an Open Large-Scale Dataset in 2D Echocardiography. En *arXiv e-prints*. https://doi.org/10.48550/arXiv.1908.06948

Leung, K. Y. E., & Bosch, J. G. (2010). Automated border detection in three-dimensional echocardiography: Principles and promises. *European Journal of Echocardiography: The Journal of the Working Group on Echocardiography of the European Society of Cardiology*, *11*(2), 97-108. https://doi.org/10.1093/ejechocard/jeq005

Liu, F., Wang, K., Liu, D., Yang, X., & Tian, J. (2021). Deep pyramid local attention neural network for cardiac structure segmentation in two-dimensional echocardiography. *Medical Image Analysis*, *67*, 101873. https://doi.org/10.1016/j.media.2020.101873

Long, J., Shelhamer, E., & Darrell, T. (2015). *Fully Convolutional Networks for Semantic Segmentation* (arXiv:1411.4038). arXiv. http://arxiv.org/abs/1411.4038

M. Pierre. (2020). Cardiologie: Généralités sur le coeur, Rythme cardiaque, Structure Anatomique. *Santé sur le Net, l'information médicale au cœur de votre santé*. https://www.sante-sur-le-net.com/maladies/cardiologie/generalites-coeur/

Michelucci, U. (2019). *Advanced Applied Deep Learning: Convolutional Neural Networks and Object Detection*. Apress. https://doi.org/10.1007/978-1-4842-4976-5

Milletari, F., Yigitsoy, M., Navab, N., & Ahmadi, S.-A. (2014). Left Ventricle Segmentation in Cardiac Ultrasound Using Hough-Forests With Implicit Shape and Appearance Priors. *midas*. https://doi.org/10.54294/y9qm6j

Nascimento, J. C., & Carneiro, G. (2020). One Shot Segmentation: Unifying Rigid Detection and Non-Rigid Segmentation Using Elastic Regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(12), 3054-3070. https://doi.org/10.1109/TPAMI.2019.2922959

Noble, J. A., & Boukerroui, D. (2006). Ultrasound image segmentation: A survey. *IEEE Transactions on Medical Imaging*, *25*(8), 987-1010. https://doi.org/10.1109/TMI.2006.877092

Oktay, O., Ferrante, E., Kamnitsas, K., Heinrich, M., Bai, W., Caballero, J., Cook, S., de Marvao, A., Dawes, T., O'Regan, D., Kainz, B., Glocker, B., & Rueckert, D. (2018). Anatomically Constrained Neural Networks (ACNN): Application to Cardiac Image Enhancement and Segmentation. *IEEE Transactions on Medical Imaging*, *37*(2), 384-395. https://doi.org/10.1109/TMI.2017.2743464

Oktay, O., Shi, W., Keraudren, K., Caballero, J., & Rueckert, D. (2014, septiembre 14). *Learning Shape Representations for Multi-Atlas Endocardium Segmentation in 3D Echo Images*. https://doi.org/10.13140/2.1.3767.5522

Painchaud, N., Duchateau, N., Bernard, O., & Jodoin, P.-M. (2023). *TED project*. https://humanheart-project.creatis.insa-lyon.fr/ted.html

Patterson, J., & Gibson, A. (2017). *Deep Learning [Book]*. https://www.oreilly.com/library/view/deep-learning/9781491924570/

Pedrosa, J., Queirós, S., Bernard, O., Engvall, J., Edvardsen, T., Nagel, E., & D'hooge, J. (2017). Fast and Fully Automatic Left Ventricular Segmentation and Tracking in Echocardiography Using Shape-Based B-Spline Explicit Active Surfaces. *IEEE Transactions on Medical Imaging*, *36*(11), 2287-2296. https://doi.org/10.1109/TMI.2017.2734959

Ronneberger, O., Fischer, P., & Brox, T. (2015a). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. *9351*, 234-241. https://doi.org/10.1007/978-3-319-24574-4_28

Ronneberger, O., Fischer, P., & Brox, T. (2015b). *U-Net: Convolutional Networks for Biomedical Image Segmentation* (arXiv:1505.04597). arXiv. https://doi.org/10.48550/arXiv.1505.04597

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533-536.

Serna, M. (2019). *PLANOS ECOCARDIOGRAFICOS*. https://ecocritic.es/wp-content/uploads/2019/10/PLANOS-ECOCARDIOGRAFICOS.pdf

Shah, D. (2022). *The Essential Guide to Data Augmentation in Deep Learning*. https://www.v7labs.com/blog/data-augmentation-guide

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Smistad, E., & Lindseth, F. (2014). *Real-time Tracking of the Left Ventricle in 3D Ultrasound Using Kalman Filter and Mean Value Coordinates*. https://doi.org/10.13140/2.1.1330.6888

Smistad, E., Østvik, A., Haugen, B. O., & L⃝vstakken, L. (2017). 2D left ventricle segmentation using deep learning. *2017 IEEE International Ultrasonics Symposium (IUS)*, 1-4. https://doi.org/10.1109/ULTSYM.2017.8092573

Toet, A., & Wu, T. (2014). Efficient contrast enhancement through log-power histogram modification. *Journal of Electronic Imaging*, *23*, 063017. https://doi.org/10.1117/1.JEI.23.6.063017

Universidad de Murcia. (2005). *Técnicas de filtrado. Tema 6*. Universidad de Murcia. https://www.um.es/geograf/sigmur/teledet/tema06.pdf

Van Steenkiste, G. (2020). *Equine electrocardiography revisited: 12- lead recording, vectorcardiography and the power of machine intelligence*.

van Stralen, M., Haak, A., Leung, K., Burken, G., & Bosch, J. G. (2014). Segmentation of Multi-Center 3D Left Ventricular Echocardiograms by Active Appearance Models. *MIDAS*. https://doi.org/10.54294/cnimu5

Vasilev, I., Slater, D., Spacagna, G., Roelants, P., & Zocca, V. (2019). *Python deep learning: Exploring deep learning techniques and neural network architectures with PyTorch, Keras, and TensorFlow* (Second edition). Packt Publishing Limited.

Veni, G., Moradi, M., Bulu, H., Narayan, G., & Syeda-Mahmood, T. (2018). Echocardiography segmentation based on a shape-guided deformable model driven by a fully convolutional network prior. *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, 898-902. https://doi.org/10.1109/ISBI.2018.8363716

Vicente, F. R. (2023). *Historia de la IA: Frank Rosenblatt y el Mark I Perceptrón, el primer ordenador fabricado específicamente para crear redes neuronales en 1957*. Telefónica Tech. https://telefonicatech.com/blog/historia-de-la-ia-frank-rosenblatt-y-e

Wang, C., Wang, C., & Smedby, O. (2014). Model-based left ventricle segmentation in 3D ultrasound using phase image. *The MIDAS Journal*. https://doi.org/10.54294/y53rnw

Wikipedia. (2023a). Bias–variance tradeoff. En *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Bias%E2%80%93variance_tradeoff&oldid=1176219028

Wikipedia. (2023b). Corazón humano. En *Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Coraz%C3%B3n_humano&oldid=151210373

Wikipedia. (2023c). Dendrita. En *Wikipedia, la enciclopedia libre*. https://es.wikipedia.org/w/index.php?title=Dendrita&oldid=151172645

Wikipedia. (2023). Medical imaging. En *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Medical_imaging&oldid=1174740060

Yap, M. H., Edirisinghe, E., & Bez, H. (2010). Processed images in human perception: A case study in ultrasound breast imaging. *European Journal of Radiology*, *73*(3), 682-687. https://doi.org/10.1016/j.ejrad.2008.11.007

Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., & Sang, N. (2018). BiSeNet: Bilateral Segmentation Network for Real-Time Semantic Segmentation. En V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer Vision – ECCV 2018* (pp. 334-349). Springer International Publishing. https://doi.org/10.1007/978-3-030-01261-8_20

Zhang, A., C.Lipton, Z., Li, M., & J.Smola, A. (2019). *7.2. Convolutions for Images—Dive into Deep Learning 1.0.3 documentation*. http://d2l.ai/chapter_convolutional-neural-networks/conv-layer.html

Zhang, Z., Liu, Q., & Wang, Y. (2018). Road Extraction by Deep Residual U-Net. *IEEE Geoscience and Remote Sensing Letters*, *15*(5), 749-753. https://doi.org/10.1109/LGRS.2018.2802944

Zhuang, J. (2019). *LadderNet: Multi-path networks based on U-Net for medical image segmentation* (arXiv:1810.07810). arXiv. http://arxiv.org/abs/1810.07810