APPLICATIONS OF DEEP LEARNING ON CARDIAC MRI: DESIGN APPROACHES FOR A COMPUTER AIDED DIAGNOSIS

Manuel Pérez Pelegrí

Dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy
Ph.D. in Technologies for Health and Well-Being

Supervisors:
Prof. Dr. David Moratal Pérez
Dr. José Vicente Monmeneu Menadas
Dr. María Pilar López Lereu

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

February 2023

Supervisors: **Prof. Dr. David Moratal Pérez**
Universitat Politècnica de València, Valencia, Spain
Dr. José Vicente Monmeneu Menadas
ASCIRES Biomedical Group
Dr. María Pilar López Lereu
ASCIRES Biomedical Group

The research described in this thesis was conducted in the Centre for Biomaterials and Tissue Engineering of the Universitat Politècnica de València in collaboration with:

- ASCIRES Biomedical Group
- Hospital Clínic Universitari de València

“I think that if you work as a radiologist you are like Wile E. Coyote in the cartoon. You’re already over the edge of the cliff, but you haven’t yet looked down. There’s no ground underneath. It’s just completely obvious that in five years deep learning is going to do better than radiologists. It might be ten years.”

Geoffrey Hinton, 2016

“AI won’t replace radiologists, but radiologists who use AI will replace those who don’t”.

Curtis Langlotz, 2019

“AI will certainly affect radiologist, they just won´t have to do any manual tasks and will only focus on diagnosis with more information than ever, and the best thing is they won´t even need to learn to use any new tool, that is the job of us biomedical engineers”.

Just a new PhD out there, 2022

Acknowledgements

A mis directores de tesis, David Moratal, María Pilar López y José Vicente Monmeneu.

A todos los coautores que participaron en este Proyecto y a aquellos compañeros cuyos consejos facilitaron el desarrollo del mismo.

A mis supervisores en Lincbiotech, por facilitarme la realización de este proyecto en el contexto de relación laboral.

Finalmente, a mis padres, mi hermana Lucia y a Julia. Sin su apoyo en la situación en la que se llevó a cabo el presente trabajo, este no hubiera podido llevarse a término. Especialmente a Lucia que proporcionó la ayuda necesaria para completar parte del mismo en momentos cruciales.

Abstract

Cardiovascular diseases are one of the most predominant causes of death and comorbidity in developed countries, as such heavy investments have been done in recent decades in order to produce high quality diagnosis tools and treatment applications for cardiac diseases. One of the best proven tools to characterize the heart has been *magnetic resonance imaging* (MRI), thanks to its high-resolution capabilities in both spatial and temporal dimensions, allowing to generate dynamic imaging of the heart that enable accurate diagnosis. The dimensions of the left ventricle and the ejection fraction derived from them are the most powerful predictors of cardiac morbidity and mortality, and their quantification has important connotations for the management and treatment of patients. Thus, cardiac MRI is the most accurate imaging technique for left ventricular assessment. In order to get an accurate and fast diagnosis, reliable image-based biomarker computation through image processing software is needed. Nowadays most of the employed tools rely in semi-automatic *Computer-Aided Diagnosis* (CAD) systems that require the clinical expert to interact with it, consuming valuable time from the professionals whose aim should only be at interpreting results. A paradigm shift is starting to get into the medical sector where fully automatic CAD systems do not require any kind of user interaction. These systems are designed to compute any required biomarkers for a correct diagnosis without impacting the physician natural workflow and can start their computations the moment an image is saved within a hospital archive system.

Automatic CAD systems, although being highly regarded as one of next big advances in the radiology world, are extremely difficult to develop and rely on *Artificial Intelligence* (AI) technologies in order to reach medical standards. In this context, *Deep learning* (DL) has emerged in the past decade as the most successful technology to address this problem. More specifically, *convolutional neural networks* (CNN) have been one of the most successful and studied techniques for image analysis, including medical imaging. In this work we describe the main applications of CNN for fully automatic CAD systems to help in the clinical diagnostics routine by means of cardiac MRI. The work covers the main points to take into account in order to develop such systems and presents different impactful results within the use of CNN to cardiac MRI, all separated in three different main projects.

The first project involves the problem of automatic segmentation of the main cardiac regions within MR images. We present a new type of CNN architecture called PSPU-net and compare its results against the classical and successful architecture 3D U-net. Both models could achieve state-of-the-art segmentation results, but the PSPU-net consistently surpassed the classic model in all settings and targeted tissues. The results demonstrate that the new PSPU-net model proposed can generalize the predicted segmentations better and with less computational resources. making it an excellent candidate to use on cardiac MRI to produce almost instant segmentations, enabling fast computation of the biomarkers of interest from them.

The second project treats the problem of automatic computation of biomarkers from the images without employing any in-between segmentation step. We focus the target on the volume of the left ventricle on cardiac MRI. This type of approach has the advantage of not requiring costly labels such as manual segmentations, however it cannot be effectively integrated in a radiological setting due to the lack of spatial contextual information produced, meaning that the expert will not know where the prediction generated by the system came from. We address this problem known as explainabilty using a weak-supervised learning approach that allows to train a neural network with only the targeted biomarker values and enables it to produce a segmentation mask that directly shows the region that the model used to predict the biomarker. The considered approach addresses two problems at the same type, producing an explainable model that can be trusted in the clinical scenario and additionally provides a way to train models for segmentation when only the volume of the target region is available, potentially broadening the number of image databanks that could be exploited. The trained model was capable of estimating the left ventricle volumes with very low errors and with excellent correlation. Additionally, the segmentation masks that it generated were always accurately located in the correct region with a good overall quality that allowed for a high explainabilty power.

The last project addresses the detection of the two major events in the cardiac cycle within the cardiac MRI, the *end-systole* and the *end-diastole*. Detecting these frames before any type of segmentation or biomarker estimation is required, as only these two time points are used for relevant cardiac contractility function calculations. In this project a fully convolutional neural network scheme is employed to treat both spatial and temporal analysis of the sequence. The key components of the developed model are the use of dilated convolutions for the temporal analysis and the use of the overlap *Dice loss* function for training, which has been very successful for segmentation tasks, but has not been employed for event detection on temporal data. We trained the model with this loss and compared its results with the same model trained with the classic *cross-entropy loss*. The results showed that the *Dice loss* has notorious superiority for this task. The final model obtained highly accurate results at detecting both the *end-systole* and *end-diastole,* making it suitable to be used in clinical contexts.

The full work presented describes novel and powerful approaches to apply CNN to cardiac MRI analysis. The work provides several key findings, enabling the integration in several ways of this novel but non-stop growing technology into fully automatic CAD systems that could produce highly accurate, fast and reliable results. The results described will greatly improve and impact the workflow of the clinical experts in the near future.

Keywords: cardiac magnetic resonance imaging, deep learning, convolutional neural networks, computer-aided diagnosis, image segmentation, explainable-AI, weak-supervised learning, dynamic imaging event detection

Resumen

Las enfermedades cardiovasculares son una de las causas más predominantes de muerte y comorbilidad en los países desarrollados, por ello se han realizado grandes inversiones en las últimas décadas para producir herramientas de diagnóstico y aplicaciones de tratamiento de enfermedades cardíacas de alta calidad. Una de las mejores herramientas de diagnóstico para caracterizar el corazón ha sido la *imagen por resonancia magnética* (IRM) gracias a sus capacidades de alta resolución tanto en la dimensión espacial como temporal, lo que permite generar imágenes dinámicas del corazón para un diagnóstico preciso. Las dimensiones del ventrículo izquierdo y la fracción de eyección derivada de ellos son los predictores más potentes de morbilidad y mortalidad cardiaca y su cuantificación tiene connotaciones importantes para el manejo y tratamiento de los pacientes. De esta forma, la IRM cardiaca es la técnica de imagen más exacta para la valoración del ventrículo izquierdo. Para obtener un diagnóstico preciso y rápido, se necesita un cálculo fiable de biomarcadores basados en imágenes a través de software de procesamiento de imágenes. Hoy en día la mayoría de las herramientas empleadas se basan en sistemas semiautomáticos de *Diagnóstico Asistido por Computador* (CAD) que requieren que el experto clínico interactúe con él, consumiendo un tiempo valioso de los profesionales cuyo objetivo debería ser únicamente interpretar los resultados. Un cambio de paradigma está comenzando a entrar en el sector médico donde los sistemas CAD completamente automáticos no requieren ningún tipo de interacción con el usuario. Estos sistemas están diseñados para calcular los biomarcadores necesarios para un diagnóstico correcto sin afectar el flujo de trabajo natural del médico y pueden iniciar sus cálculos en el momento en que se guarda una imagen en el sistema de archivo informático del hospital.

Los sistemas CAD automáticos, aunque se consideran uno de los grandes avances en el mundo de la radiología, son extremadamente difíciles de desarrollar y dependen de tecnologías basadas en *inteligencia artificial* (IA) para alcanzar estándares médicos. En este contexto, el *aprendizaje profundo* (DL) ha surgido en la última década como la tecnología más exitosa para abordar este problema. Más específicamente, las *redes neuronales convolucionales* (CNN) han sido una de las técnicas más exitosas y estudiadas para el análisis de imágenes, incluidas las imágenes médicas. En este trabajo describimos las principales aplicaciones de CNN para sistemas CAD completamente automáticos para ayudar en la rutina de diagnóstico clínico mediante resonancia magnética cardíaca. El trabajo cubre los puntos principales a tener en cuenta para desarrollar tales sistemas y presenta diferentes resultados de alto impacto dentro del uso de CNN para resonancia magnética cardíaca, separados en tres proyectos diferentes que cubren su aplicación en la rutina clínica de diagnóstico.

El primer proyecto implica el problema de la segmentación automática de las principales regiones cardíacas dentro de las imágenes de RM. Presentamos un nuevo tipo de arquitectura CNN llamada PSPU-net y comparamos sus resultados con la clásica y exitosa arquitectura 3D U-net. Ambos modelos logran resultados de segmentación de alta calidad, pero la PSPU-net supera sistemáticamente al modelo clásico en todos los contextos y tejidos específicos. Los resultados demuestran que el nuevo modelo PSPU-net propuesto puede generalizar mejor las segmentaciones predichas y con menos recursos computacionales, lo que lo convierte en un excelente candidato para su uso en imágenes de resonancia magnética cardíaca para producir segmentaciones casi instantáneas, permitiendo un cálculo rápido de los biomarcadores de interés a partir de las mismas.

El segundo proyecto trata el problema del cálculo automático de biomarcadores a partir de las imágenes sin emplear ningún paso de segmentación intermedio. Enfocamos el objetivo en el volumen del ventrículo izquierdo en la imagen de resonancia magnética cardíaca. Este tipo de enfoque tiene la ventaja de que no requiere etiquetas costosas de obtener como las segmentaciones manuales, sin embargo, no se puede integrar de manera efectiva en un entorno radiológico debido a la falta de información contextual espacial producida, lo que significa que el experto no sabrá de donde viene la predicción ofrecida por el sistema. Abordamos este problema conocido como *explicabilidad* utilizando un enfoque de *aprendizaje débilmente supervisado* que permite entrenar una red neuronal solo con los valores de biomarcadores específicos, y le permite producir una máscara de segmentación que muestra directamente la región que el modelo usó para predecir el biomarcador. El enfoque considerado aborda dos problemas diferentes pero relacionados, produciendo un modelo explicable en el que se puede confiar en el escenario clínico y, además, proporciona una forma de entrenar modelos para la segmentación cuando solo está disponible el volumen de la región objetivo, lo que podría ampliar el número de bancos de imágenes que podrían ser explotados. El modelo entrenado fue capaz de estimar los volúmenes del ventrículo izquierdo con errores bajos y con una excelente correlación. Además, las máscaras de segmentación que generaba siempre se ubicaban con precisión en la región correcta con una buena calidad general ofreciendo una alta capacidad de explicabilidad.

El último proyecto aborda la detección de los dos eventos temporales en el ciclo cardíaco dentro de la secuencia de resonancia magnética cardíaca, la *telesístole* (fin de la sístole) y la *telediástole* (fin de la diástole). Detectar estos eventos es un prerrequisito previo a cualquier tipo de segmentación o estimación de biomarcadores, ya que solo se pueden utilizan estos dos puntos temporales para los cálculos relevantes de la función de contractilidad cardíaca. En este proyecto se emplea un esquema de red neuronal convolucional para tratar el análisis espacial y temporal de la secuencia. Los componentes clave del modelo desarrollado son el uso de convoluciones dilatadas para el análisis temporal y el uso de la función de pérdida de solapamiento de Dice para el entrenamiento, la cual ha tenido mucho éxito para tareas de segmentación, pero no se ha empleado para la detección de eventos en datos temporales. Se entrenó el modelo con esta función de pérdida y se compararon los resultados con el mismo modelo entrenado con la clásica función de entropía cruzada. Los resultados mostraron que la función de pérdida de Dice es notablemente superior para la tarea. El modelo final obtuvo resultados muy precisos al detectar tanto el final de sístole como el final de diástole, lo que lo hace adecuado para su uso en contextos clínicos.

El trabajo completo presentado describe enfoques novedosos y de alto impacto para aplicar CNN al análisis de resonancia magnética cardíaca. El trabajo proporciona varios hallazgos clave, permitiendo varias formas de integración de esta reciente y creciente tecnología en sistemas CAD completamente automáticos que pueden producir resultados altamente precisos, rápidos y confiables. Los resultados descritos mejorarán e impactarán positivamente el flujo de trabajo de los expertos clínicos en un futuro próximo.

Palabras clave: imagen por resonancia magnética cardíaca, aprendizaje profundo, redes neuronales convolucionales, diagnóstico asistido por computadora, segmentación de imágenes, IA explicable, aprendizaje débilmente supervisado, detección de eventos en imágenes dinámicas

Resum

Les malalties cardiovasculars són una de les causes de mort i comorbiditat més predominants als països desenvolupats, s'han fet grans inversions en les últimes dècades per tal de produir eines de diagnòstic d'alta qualitat i aplicacions de tractament de malalties cardíaques. Una de les tècniques millor provades per caracteritzar el cor ha estat la imatge per ressonància magnètica (IRM), gràcies a les seves capacitats d'alta resolució tant en dimensions espacials com temporals, que permeten generar imatges dinàmiques del cor per a un diagnòstic precís. Les dimensions del ventricle esquerre i la fracció d'ejecció que se'n deriva són els predictors més potents de morbiditat i mortalitat cardíaca i la seva quantificació té connotacions importants per al maneig i tractament dels pacients. D'aquesta manera, la IRM cardíaca és la tècnica d'imatge més exacta per a la valoració del ventricle esquerre. Per obtenir un diagnòstic precís i ràpid, es necessita un càlcul fiable de biomarcadors basat en imatges mitjançant un programa de processament d'imatges. Actualment, la majoria de les ferramentes emprades es basen en sistemes semiautomàtics de Diagnòstic Assistit per ordinador (CAD) que requereixen que l'expert clínic interaccioni amb ell, consumint un temps valuós dels professionals, l'objectiu dels quals només hauria de ser la interpretació dels resultats. S'està començant a introduir un canvi de paradigma al sector mèdic on els sistemes CAD totalment automàtics no requereixen cap tipus d'interacció amb l'usuari. Aquests sistemes estan dissenyats per calcular els biomarcadors necessaris per a un diagnòstic correcte sense afectar el flux de treball natural del metge i poden iniciar els seus càlculs en el moment en què es deixa la imatge dins del sistema d'arxius hospitalari.

Els sistemes CAD automàtics, tot i ser molt considerats com un dels propers grans avanços en el món de la radiologia, són extremadament difícils de desenvolupar i depenen de les tecnologies d'Intel·ligència Artificial (IA) per assolir els estàndards mèdics. En aquest context, l'aprenentatge profund (DL) ha sorgit durant l'última dècada com la tecnologia amb més èxit per abordar aquest problema. Més concretament, les xarxes neuronals convolucionals (CNN) han estat una de les tècniques més utilitzades i estudiades per a l'anàlisi d'imatges, inclosa la imatge mèdica. En aquest treball es descriuen les principals aplicacions de CNN per a sistemes CAD totalment automàtics per ajudar en la rutina de diagnòstic clínic mitjançant ressonància magnètica cardíaca. El treball recull els principals punts a tenir en compte per desenvolupar aquest tipus de

sistemes i presenta diferents resultats d'impacte en l'ús de CNN a la ressonància magnètica cardíaca, tots separats en tres projectes principals diferents.

El primer projecte implica el problema de la segmentació automàtica de les principals regions cardíaques dins d'imatges de RM. Presentem un nou tipus d'arquitectura CNN anomenada PSPU-net i comparem els seus resultats amb l'arquitectura clàssica 3D U-net. Els dos models aconsegueixen resultats de segmentació de alta qualitat, però la xarxa PSPU va superar el model clàssic en tots els entorns i teixits objectiu. Els resultats demostren que el nou model PSPU-net proposat pot generalitzar millor les segmentacions previstes i amb menys recursos computacionals. convertint-lo en un excel·lent candidat per utilitzar-lo en ressonància magnètica cardíaca per produir segmentacions quasi instantànies, permetent un càlcul ràpid dels biomarcadors d'interès d'ells.

El segon projecte tracta el problema del càlcul automàtic de biomarcadors a partir de les imatges sense emprar cap pas de segmentació intermèdia. Centrem l'objectiu en el volum del ventricle esquerre de la ressonància magnètica cardíaca. Aquest tipus d'enfocament té l'avantatge de no requerir etiquetes costoses com les segmentacions manuals, però no es pot integrar eficaçment en un entorn radiològic a causa de la falta d'informació contextual espacial produïda, de manera que l'expert no sabrà d’on prové la predicció generada pel sistema. Abordem aquest problema conegut com a explicabilitat mitjançant un enfocament d'aprenentatge supervisat feble que permet entrenar una xarxa neuronal només amb els valors de biomarcadors objectiu i li permet produir una màscara de segmentació que mostra directament la regió que el model va utilitzar per predir el biomarcador. L'enfocament considerat aborda dos problemes del mateix tipus, produint un model explicable en el que es pot confiar a l'escenari clínic i, a més, proporciona una manera d'entrenar models per a la segmentació quan només està disponible el volum de la regió objectiu, ampliant potencialment el nombre de bancs d’imatge i de dades que es podrien explotar. El model entrenat es capaç d'estimar els volums del ventricle esquerre amb errors molt baixos i amb una correlació excel·lent. A més, les màscares de segmentació que produides sempre estaven localitzades amb precisió a la regió correcta amb una bona qualitat general que permetia una gran capacitat d'explicació.

L'últim projecte aborda la detecció dels dos esdeveniments principals del cicle cardíac dins de la RM cardíaca, la telesístole (sístole final) i la telediàstole (diàstole final). La detecció d'aquests punts és necessària abans de qualsevol tipus de segmentació o estimació de biomarcadors, ja que només aquests dos punts de temps s'utilitzen per als càlculs rellevants de la funció de contractilitat cardíaca. En aquest projecte s'utilitza un

esquema de xarxa neuronal totalment convolucional per tractar tant l'anàlisi espacial com temporal de la seqüència. Els components clau del model desenvolupat són l'ús de cconvolucions dilatades per a l'anàlisi temporal i l'ús de la funció de pèrdua de superposament de Dice, que ha tingut molt èxit per a tasques de segmentació, però no s'ha utilitzat per a la detecció d'esdeveniments en seqüencies temporals. Es va entrenat el model amb aquesta pèrdua i vam comparat els seus resultats amb el mateix model entrenat amb la clàssica funció de pèrdua d'entropia creuada. Els resultats van mostrar que la pèrdua de Dice té una superioritat notòria per a aquesta tasca. El model final va obtenir resultats altament precisos a l'hora de detectar tant la sístole final com la diàstole final, el que el va apte per ser utilitzat en contextos clínics.

El treball complet presentat descriu enfocaments nous i potents per aplicar CNN a l'anàlisi de ressonància magnètica cardíaca. El treball proporciona diversos descobriments clau, que permeten la integració de diverses maneres d'aquesta tecnologia nova però en constant creixement en sistemes CAD totalment automàtics que podrien produir resultats altament precisos, ràpids i fiables. Els resultats descrits milloraran i afectaran considerablement el flux de treball dels experts clínics en un futur proper.

Paraules clau: ressonància magnètica cardíaca, aprenentatge profund, xarxes neuronals convolucionals, diagnòstic assistit per ordinador, segmentació d'imatges, IA explicable, aprenentatge supervisat feble, detecció d'esdeveniments d'imatge dinàmica

Abbreviations and Acronyms

Artificial intelligence and deep learning

ADAM	Adaptative Moment Estimator
ADAMW	Adaptative Moment Estimator with Weight decay
AI	Artificial Intelligence
AN	Artificial Neuron/s
ANN	Artificial Neural Network/s
BERT	Bidirectional Encoder Representation Transformer
BN	Batch Normalization
CNN	Convolutional Neural Network/s
CAM	Class Activation Mapping / Class Activation Map
GCU	Growing Cosine Unit
GPT-3	Third generation Generative Pre-trained Transformer
DL	Deep Learning
FCL	Fully Connected Layer/s
FCN	Fully Convolutional Neural Network/s
FCNN	Fully Connected Neural Network/s
GAN	Generative Adversarial Network/s
GDL	Generalized Dice loss
GELU	Gaussian Error Linear Unit
Grad-CAM	Gradient-Weighted Class Activation Mapping
GRU	Gated Recurrent Unit
LIME	Local Interpretable Model-Agnostic Explanation
LSTM	Long Short Term Memory
ML	Machine Learning
MLP	Multilayer Perceptron/s
NLP	Natural Language Processing
PSP	Pyramid Scene Parsing

ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network/s
SGD	Stochastic Gradient Descent
SHAP	SHapley Additive exPlanations
WDL	Weighted Dice Loss
XAI	Explainable Artificial Intelligence

Medical imaging

CMRI	Cardiac Magnetic Resonance Imaging
CT	Computed Tomography
DICOM	Digital Imaging and Communications in Medicine
MRI	Magnetic Resonance Imaging
NIfTI	Neuroimaging Informatics Technology Initiative
PACS	Picture Archiving and Communication System

Magnetic resonance imaging

$\mathbf{B_0}$	Magnetic Field
GRE	Gradient Echo
$\mathbf{^1H}$	Hydrogen atom
$\mathbf{M_0}$	Net magnetization
RF	Radio frequency
T	Tesla unit
T1	longitudinal relaxation time
T2	Transversal relaxation time
TE	Echo Time
TR	Repetition Time
SE	Spin-Echo
SSFP	Steady State Free Precession

Cardiovascular setting

CO	Cardiac Output
ECG	Electrocardiography

ED	End-Diastole
EF	Ejection Fraction
ES	End-Systole
LV	Left Ventricle
RV	Right Ventricle
SV	Stroke Volume
VI	Volume Index
WThg	Wall Thickening

Hardware and software

CAD	Computer Aided Diagnosis
CPU	Central Processing Unit
GB	Gigabyte
GPU	Graphics Processing Unit
IDE	Integrated Development Environment
RAM	Random Access Memory
TPU	Tensor Processing Unit
VRAM	Video Random Access Memory

Miscellaneous

2D	Two dimensions, two-dimensional or bi-dimensional
3D	Three dimensions or three-dimensional
4D	Four dimensions or fourth-dimensional
BSA	Body Surface Area
DC	Dice Coefficient
FOV	Field Of View
GDS	Generalized Dice Score
MAE	Mean Absolute Error
ml	Milliliters
NN	Neural Network
RMAE	Relative Mean Absolute Error
RMSE	Root Mean Square Error
ROI	Region Of Interest

XOR	Exclusive OR logical operator

Contents

Chapter 1. Introduction

1.1. Artificial intelligence and healthcare

Artificial intelligence (AI) has seen a great explosion in popularity in recent years. Its applications extend to any field where some type of automation might be useful. However, although AI has been around for many years, it became notoriously popular in the last decade due to the advent of most of the computing technology employed in deep learning (Figure 1.1).

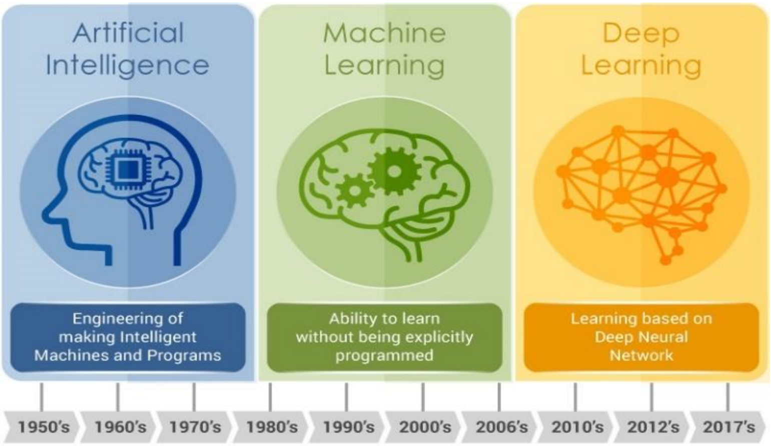

Artificial Intelligence
Engineering of making Intelligent Machines and Programs
Machine Learning
Ability to learn without being explicitly programmed
Deep Learning
Learning based on Deep Neural Network
1950's
1960's
1970's
1980's
1990's
2000's
2006's
2010's
2012's
2017's

Figure 1.1 Overview of the historical evolution of Artificial intelligence, from its beginning in the 1950's to the emergence of practical applications of deep learning in the 2010's. Image source: https://www.privatewallmag.com/inteligencia-artificial-machine-deep-learning/.

Deep learning (DL) encompasses all the AI algorithms based on artificial neural networks (ANN). Nowadays it is common to hear about AI when it is only referred to

ANN. However, ANN are only a subfield of it. More specifically AI involves any algorithm that can solve a problem in an intelligent manner, this includes any method that applies a fixed set of rules. Within AI, an important subfield is machine learning (ML), a branch of algorithms which make use of generic models that can learn to adjust themselves with the use of data in order to solve a specific problem. Within ML there are tons of algorithms, from the simple linear regression to the more complex random forest, support vector machines or artificial neural networks. ANN are one of the more powerful models within ML, these algorithms are further expanded in the deep learning subfield (encompassing all the deep neural networks). Then, within the DL field there are also a variety of different ANN architectures, convolutional neural networks (CNN) being one of the most important ones. CNN are a subtype of ANN designed for image processing, which they are especially good at. Figure 1.2 summarizes the general AI branches described.

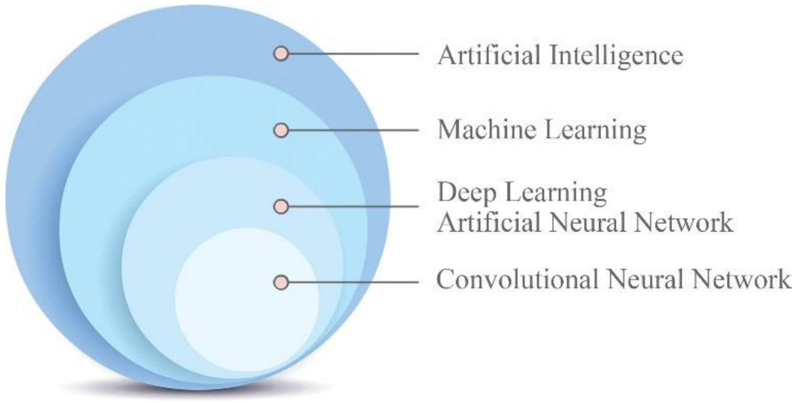


Figure 1.2 Schematic of the main branches of artificial intelligence nowadays. Machine learning is a specific type of AI algorithms that are capable of learning to improve themselves from examples. Deep learning is one type of machine learning algorithms that uses artificial neural networks. Finally, convolutional neural networks are those neural networks specially designed to treat computer vision problems [1].

Deep learning has demonstrated a great versatility and quality in its applications. Healthcare is one of the main fields where DL has gained much attention, and more specifically in the radiology field due to the more widespread data available and the CNN great success in image processing tasks. However, DL and AI in general is entirely dependent on the data available, and if the data used in not a good representative of the real population or there exists some intrinsic bias the algorithms will reproduce these issues in their response. Taking this into account any medical-related problem can be specially challenging. First because of the difficulty in the data accessibility (due to patient's privacy) and second for the possible bias present within the data that if not taken into account can lead to discriminative and fatal results [2,3]. Additionally, most of ANN are hard to interpret, and are employed as a black box that "just works". This makes adoption of these systems harder in the medical setting, where there are several tasks (such as diagnosis or treatment recommendations) that require some explanation on the decision taken in order to be accepted by the medical community.

Even with the multiple challenges that DL needs to address, it is incredibly powerful and has demonstrated being capable of solving complicated health-related tasks, and it has had great success at different applications including triages, radiological diagnosis, drug interaction predictions, telemedicine applications or electronic health records management as some important examples. This thesis is focused on the application of these models to a specific problem that involves the radiology and cardiovascular medical branches.

1.2. Motivations

A good clinical assessment based on radiological information often requires measurements of regions of interest (ROIs) from which radiological biomarkers can be extracted. This step typically involves segmentation (that is, classifying each pixel within the image in different categories for the different ROIs and the background) which can be accomplished using manual, semiautomatic or fully automatic tools.

Manual segmentation is a hard and especially time-consuming task, furthermore it is also a monotonous task. This limits manual segmentation of medical images to very expert clinicians that have very good knowledge on the images treated. On the other hand, semiautomatic segmentation typically makes use of software capable of performing general segmentation tasks as long as the user provides some sort of input to help the system perform the most complicated and laborious parts, still this usually implies that the clinical expert must learn to properly use this tool and validate its results. Often these systems also include correction tools to refine the results in case some errors were made. Finally, fully automatic segmentation allows to obtain the ROIs segmentation without any interaction from the user. These systems need to be extremely accurate and fast if it is meant to avoid posterior manual correction by the health professional, and for these reasons each segmentation problem usually requires a dedicated system especially designed to perform that specific task.

In the clinical setting it is not a widespread practice to perform extensive analysis of the radiological information due to the requirement of the user interaction with a software that usually requires experience (and much less if this also requires a full manual segmentation). This means that a lot of valuable information is lost in the diagnostic workflow. Furthermore, even in the specific cases where the clinical expert is trained to use a software to obtain more detailed information, they are still in most cases semi-automatic approaches that makes the diagnostic workflow not very efficient.

Automatic computer-aided diagnosis (CAD) is a relatively new paradigm in the diagnostic workflow which leaves the clinical user outside of the biomarker computation work. This is accomplished by integrated fully automatic systems that start processing the images the moment they are acquired and stored and provide the biomarker analysis results along the original acquisition to the final user, making the diagnostic workflow seamless. Additionally, by avoiding the intervention of the user, these systems are reproducible in different environments, as their results are user-independent. This is a relatively new paradigm that has seen great growth with the development of machine learning and deep learning technologies. Fig 1.3 summarizes the main differences between the two main diagnostic paradigms that allow for advance biomarker integration into the clinical workflow that exist today.

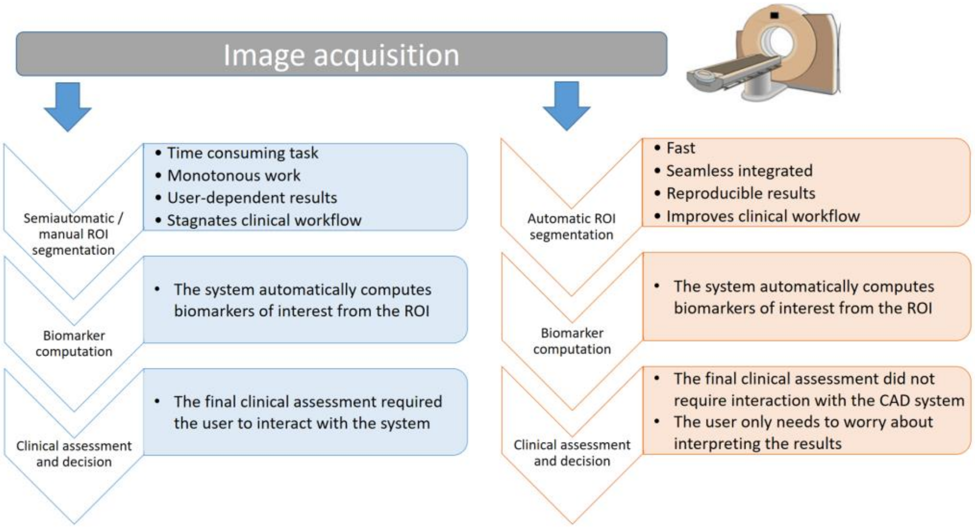


Image acquisition
Semiautomatic / manual ROI segmentation
• Time consuming task
• Monotonous work
• User-dependent results
• Stagnates clinical workflow
Biomarker computation
• The system automatically computes biomarkers of interest from the ROI
Clinical assessment and decision
• The final clinical assessment required the user to interact with the system
Automatic ROI segmentation
• Fast
• Seamless integrated
• Reproducible results
• Improves clinical workflow
Biomarker computation
• The system automatically computes biomarkers of interest from the ROI
Clinical assessment and decision
• The final clinical assessment did not require interaction with the CAD system
• The user only needs to worry about interpreting the results

Figure 1.3 Comparison of the two clinical diagnosis paradigms that make use of advance imaging biomarkers. The classic method is to use manual or semiautomatic tools to help in the segmentation of the region to analyze. The CAD paradigm completely erases any user intervention by automatically processing the images acquired. Ideally, the CAD system is integrated within the PACS system of the hospital, making the diagnosis a fast and seamless workflow.

From this perspective it is very clear that radiology can be highly impacted by CAD systems that make use of DL techniques. On one hand the development of such systems add value to the diagnostic capabilities offered by the original images by providing biomarkers that otherwise would not be available, and on the other hand allows for the prediction of these biomarkers without requiring interaction from clinical users, whose actual focus falls on the final diagnosis. In the end this will save time, allowing for more patient’s being diagnosed in the same time frame and with more information available.

The described context is especially important in those cases where the original images do not provide by themselves enough information to have a good diagnosis, and where some type of biomarker measurement is a must. One of such cases is the routinely clinical assessment of the heart by means of magnetic resonance imaging (MRI). Even in the simplest cardiac MRI studies (CMRI), it is required to have the images analyzed to extract parameters of interest. These types of studies aim at imaging the heart`s motion, which in turn allows for the calculation of certain dynamic parameters of the heart that derive from the volume values of certain regions of the heart during its cycle.

Most specifically the LVEF (left ventricle ejection fraction) is the most relevant to characterize the heart function, and to compute it the LV volumes at both end-systole and end-diastole are necessary. To compute these parameters, it is first required to select the regions of interest. In clinical practice this is usually done by employing semiautomatic software that can make most of the work, but still require the user to manually adjust the final results. As it can be seen, this is a perfect clinical scenario where deep learning-based CAD systems can be applied to improve the clinical workflow. This is even more true considering that cardiac-related conditions are one of the main causes of death in modern societies nowadays [4-6], which implies that a great demand of these techniques is assured.

1.3. Objectives

The overall aim of this thesis consisted on studying, implementing and testing DL techniques in order to replicate the typical clinical procedure applied by clinical professionals in assessing the heart condition, and more specifically the LV function employing cardiac MRI. With this approach, it was intended to investigate and develop the key components that would allow a satisfactory workflow from beginning to end based on DL technology. Besides this principal objective, a secondary one was to study additional DL methods to tackle more specific situations that could also be encountered in medical imaging in general and more specifically in cardiac MRI.

In particular, three specific objectives related to the context's problem were defined:

1. Automatic segmentation of the main regions of interest present within the images whose analysis is used to characterize the heart. This includes the segmentation of the left ventricle myocardium and the left ventricle and right ventricle inner chambers.
2. Automatic estimation of biomarker volume values of the left ventricle without employing segmentations. Additionally, this problem setting overlaps with the challenging explainable-AI problem (XAI), making this a dual objective, automatic estimation of the values and providing an explanation of the cause that produced the prediction by the artificial neural network.
3. Automatic detection of end-systole and end-diastole within the CMR images. This involves working with whole dynamic acquisitions and classifying each time point.

Besides these specific objectives, and additional objective of the thesis was to test new neural network architectures and methodologies that could be exploited in general DL settings. In this sense, all the developments were intended to be applicable to any problem of similar nature to that of CMRI analysis and provide new contributions directly involving the DL field.

1.4. Contributions to knowledge

This thesis offers three main novel contributions for the automatic assessment of the LV volume and function with conventional short-axis CMRI employing CNN. Additionally, the thesis also derives other novel contributions that are generalizable to the DL field itself, which involves different sub-fields including Weak-supervised training, loss function choice, explainable-AI and architecture design optimization.

The first contribution is that convolutional neural networks for segmenting the main regions of interest in short-axis CMRI can be reliably employed, offering excellent quality in their results. More specifically, this contribution also determines that CNN that process images in 2D can obtain better results with less parameters than those that treat the images in 3D when more specific and optimum designs are employed, allowing the use of faster and lighter models.

The second contribution is that left ventricle volume values and segmentations can be estimated with great precision with CNN when some restrictions are applied to the learning schedule, even when only trained to target the volume values with no segmentations available. The method described by this contribution allows to design weak-supervised training methods in CNN that both improves the estimated biomarkers and offers a segmentation as an estimation of the region within the image that the network targeted to calculate the biomarkers. This methodology also contributes to solve the explainability problem within the XAI field, in this case specifically targeted for the clinical user's understanding.

The third and last contribution is that CNN can automatically detect the ES and ED within short-axis CMRI sequences with a variable number of frames and of slices per frame. Concretely, models employing dilated convolutions instead of recurrent layers show a great potential to process the temporal information of these sequences with great capacity even with a reduced number of parameters. Furthermore, this contribution includes the finding that employing loss functions usually employed for training segmentation CNN can be used in problems regarding temporal classification with greater results than classic loss functions.

By combining the information offered by the three contributions this thesis derives an additional one, which is that a fully automatic CNN system to estimate the main information of interest required in the assessment of short-axis CMRI can be designed by coupling the third with either the first or second contributions. This system would allow to generate the results with high accuracy and in a matter of seconds provided it could be used with the required hardware. This would allow for a great increase in the speed of diagnosis in the clinical setting.

1.5. Thesis Structure

This thesis is structured in 10 chapters. Chapter 1 presents a summary of the general and specific objectives of the thesis and the novel contributions to knowledge. Chapters 2 to 3 describe the theoretical background that is essential for understanding the experimental studies. Chapter 4 describes the data employed for the different experiments, as well as a description of the hardware and software used in them. Chapters 5 to 7 present the experimental projects performed. Chapter 8 present the final overall conclusions of this thesis, along the limitations encountered and future lines of work. Chapter 9 covers all the referenced bibliography within the thesis. Finally, chapter 10 covers all the publications derived from this thesis. A summary of the main chapters of this thesis is introduced below:

Chapter 1: Introduction

The current chapter gives a background on motivations and objectives proposed for the thesis development. The main contributions to knowledge and the overall thesis structure is described here as well.

Chapter 2: Background on cardiac MRI

This chapter gives a background on the principles of cardiac MRI, with a focus on the techniques used to assess patients with suspected heart pathologies. The chapter begins with a summary of the anatomical and physiological principles of the heart, followed by an introduction of the general principles of MRI physics, and finishes with a detailed description of short axis cine sequences.

Chapter 3: Image processing with deep learning

In this chapter an overview on the history and the principal features and components of CNN is presented. It addresses the use of these types of DL models for image processing, and more specifically for medical image processing.

Chapter 4: Materials

This chapter cover all the materials employed in the experiments. It covers a full detailed description of the data available. The chapter also covers a description on the hardware and software used to produce all the results.

Chapter 5: Automatic semantic segmentation

This chapter describes the experimental study aimed to evaluate the use of CNN for segmenting the main regions of the left and right ventricles in the images. Two different types of CNN were compared, a more basic model that could process data in 3D and a novel architecture that only processed the data in 2D but that included several additional capabilities. The experiments aim at determining how good the segmentation results are for the different targeted tissues and how different they are between the two models.

Chapter 6: Automatic Biomarker Estimation and Explainability

In this chapter the problem of automatic estimation of biomarkers from the images is addressed. More specifically the direct estimation of the LV volume. The experiments consisted on the design and training of a model that was only fed with the LV volume but that could produce both the estimated volume and a segmentation of the region where it based this estimation. With this setting the experiment also covers the topics of weak-supervised training and explainability within the DL field.

Chapter 7: Automatic End-Systole and End-Diastole detection

The chapter cover the work done in the design and implementation of a CNN that could detect the ES and ED frames within the image sequences. The experiments included a new way of preprocessing the images to train the model more efficiently, a model design employing dilated convolutions to process the temporal information of the sequence, a final postprocessing methodology to assign a final classification based on the probabilities generated by the model and a training schedule using an overlap loss function compared to a classical classification loss function.

Chapter 8: Final conclusions

This chapter comprises the final conclusions reached throughout the thesis project, a deep discussion on the technologies described, the adequacy of the work as a whole, and its potential application in real-world conditions. Limitations and future lines of work are also described in this chapter.

Chapter 2.
Background on cardiac magnetic resonance imaging

2.1. Introduction

The heart is the organic muscle in charge of pumping blood throughout the entire human body, a heart failure can easily result in death making it one of the most important organs within the human body. This organ is the most important element when studying all major cardiovascular diseases, which are one of the main causes of death in developed countries [4-6].

Nowadays, the best technique to characterize the heart is cardiac MRI (CMRI), due to both its great spatial and temporal resolution capabilities [7-10]. CMRI encompasses any MRI acquisition that targets to image the heart and its activity.

This chapter starts with a description of key information regarding the heart anatomy and physiological function, followed by an overview of the basic elements that characterize MRI technology and ends up describing with more detail the specific application of CMRI.

2.2. Cardiac anatomy and physiology

2.2.1. Anatomical structure

The heart is an organ composed mainly of muscle tissue. It is located within the thoracic cavity, in a space known as mediastinum, placed between the lungs. It is surrounded by the pericardial sac, composed of different layers. The pericardial sac is in contact with the heart wall. The heart wall is made up of three layers: endocardium (most inner layer), myocardium (middle layer) and epicardium (most external layer, shared with the pericardial sac). The thicker layer of the heart wall is the myocardium, composed of muscular tissue (composed of cardiac fibers) in charge of its contractility function. This muscle layer is surrounded by the epicardium layer on its external face and by the endocardium on its internal face (inner layer) [11, 12].

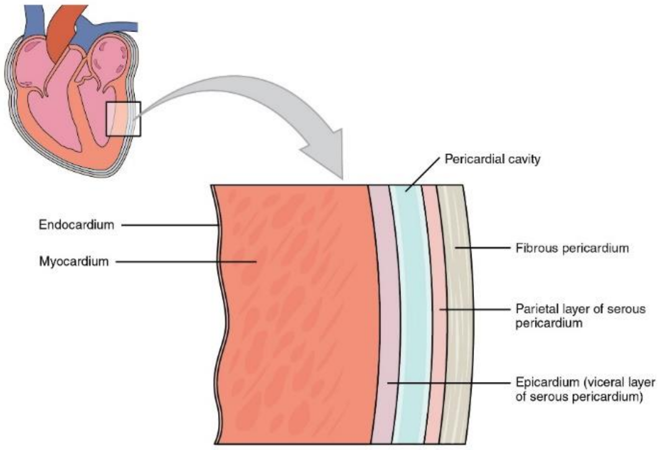

Pericardial cavity
Endocardium
Myocardium
Fibrous pericardium
Parietal layer of serous pericardium
Epicardium (viceral layer of serous pericardium)

Figure 2.1. Cardiac layers of heart that surround the myocardium by the outside and inside [11].

The heart is composed of four inner chambers or cavities: right atrium, right ventricle (RV), left atrium and left ventricle (LV). The atrium cavities receive blood from the outside while the ventricles pump it to the outside. The atrium cavities are above their respective ventricles and separated to them by the tricuspid (right) and mitral (left) valves. These valves allow the pass of blood only from the atrium to the ventricle and are controlled by the papillary muscles. The cardiac wall separating the right chambers from the left ones is the septum. The myocardium layer thickness varies across these chambers and it has its greatest thickness around the left ventricle.

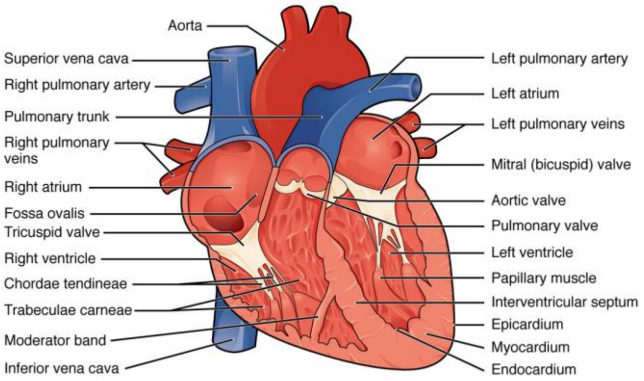


Figure 2.2 Full anatomic representation of the heart including the different valves and chambers [11].

2.2.2. Heart function

The heart's principal function is to pump oxygenated blood to the rest of the body and to recover the venous deoxygenated blood to pump it to the lungs for oxygenation. This is accomplished by its constant contraction and relaxation in the cardiac cycle which encompasses several phases (Fig 2.3).

Overall the cardiac cycle can be divided in the diastolic and the systolic phases. The diastolic phase or diastole is the state when the heart is relaxed, in contrast the systolic phase or systole is the contraction state of the heart. Both the atrial and ventricular chambers have their own systolic and diastolic phases. The entire cardiac cycle starts with the atrial filling at the end of the ventricular systole. After the filling, the ventricles start their relaxation (beginning of ventricular diastole) while the atrium cavities start their contraction (beginning of atrial systole). At the end of the atrial systole the blood passes from the atrium to the ventricles through the mitral and tricuspid valves. After this, the ventricular systole begins (the ventricles start contracting) while the atrium cavities start their diastole phase. At the end of the ventricular systole the ventricles pump all the blood to the outside of the heart and the cycle starts again.

The ventricular systole is remarkably different between the left and right ventricles. The main reason is that the RV pumps the blood to the lungs which are proximal to the heart. In contrast the LV pumps the blood throughout the entire body until reaching again the heart. Thus, the ejection force of the LV is greater, which is accomplished by its thicker cardiac wall.

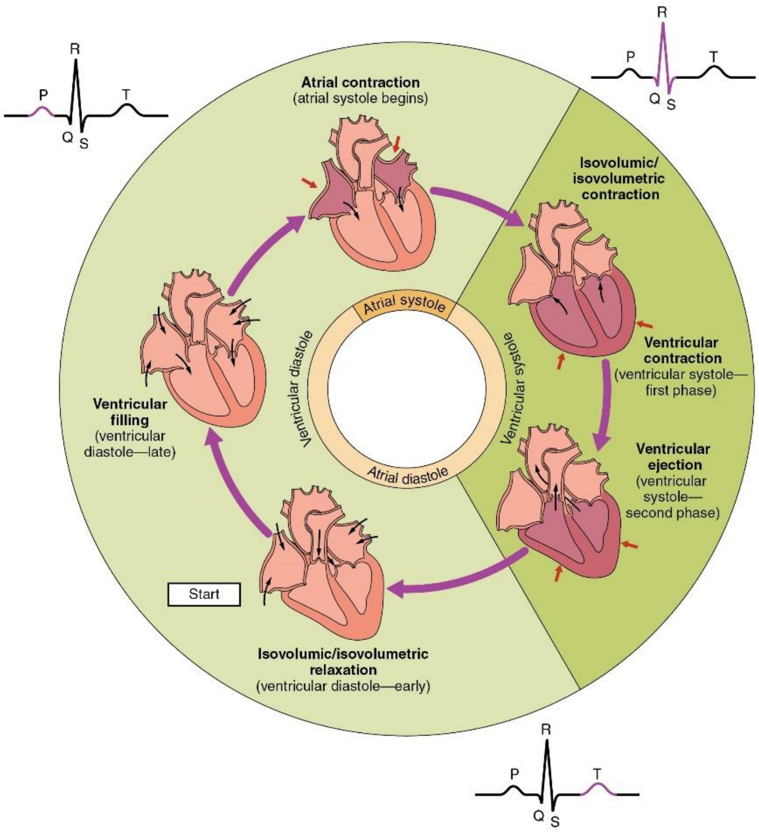


Figure 2.3 Schematic of the cardiac cycle [11].

There are different cardiac parameters that measure the functionality and give valuable information regarding its contractility function. Since the left ventricle is in charge of pumping the blood to the body and this is the main heart’s function, these parameters are focused on the LV function, but they are equally applicable to the RV. The most important ones are the following:

- **End-systolic volume:** the volume of blood measure within the ventricle at the end of contraction measured in *ml*. It represents the minimum amount of blood present within the ventricle. Normal values range from 22-66 ml and 26-86 ml for women and men respectively for the LV [13].
- **End-diastolic volume:** the volume of blood measured within the ventricle at the end of relaxation measured in *ml*. It represents the maximum amount of blood when the ventricle is filled. Normal values range from 86-178 ml and 106-214 ml for women and men respectively for the LV [13].
- **Myocardium volume:** the volume of the myocardium. Measured at end-diastole. Usually only measured for the LV. Needed to derive the ventricle mass.
- **Stroke volume (*SV*):** the blood volume pumped by the ventricle in one heartbeat. The normal resting value for stroke volume is approximately 70 ml/beat [12].

$$SV\ (ml) = End\ diastolic\ volume - End\ sistolic\ volume \qquad \text{Equation 2.1}$$

- **Cardiac output (*CO*):** the volume of blood pumped by the ventricle in one minute. An average healthy adult has on average 75 beats/min, so in the same conditions the cardiac output will be 5250 ml/min [12].

$$CO\ (ml/min) = Stroke\ volume \times heart\ rate \qquad \text{Equation 2.2}$$

- **Ejection fraction (*EF*):** the percentage of blood pumped by the ventricle from the volume of blood when the ventricle is filled in one beat. The normal value is around 60% (normal range 57-77%) [13, 14].

$$EF\ (\%) = \frac{stroke\ volume\ \times 100}{end\ diastolic\ volume} \qquad \text{Equation 2.3}$$

- **Volume index (*VI*):** a normalized measurement of the volumes. Applicable to end-systolic, end-diastolic and stroke volumes. The volume values vary depending on body size, so this measure normalizes the volume taking into account the body surface area (*BSA*). There are several formulations proposed to express BSA [15], but the the Mosteller formula [16] presented in Equation 2.4 is one of the most extended ones.

$$BSA\ (m^2) = \sqrt{height(cm)\ weight(kg)/3600} \qquad \textit{Equation 2.4}$$

$$VI\ (ml/m^2) = \frac{Volume}{BSA} \qquad \textit{Equation 2.5}$$

- **Wall thickening (*WThg*):** the thickening of the myocardium during wall motion, between end-diastole (*WTh*$_{ED}$) and end-systole (*WTh*$_{ES}$). In regional analysis, a segment with wall thickening < 2mm is considered affected [17].

$$WTh\ (mm) = WTh_{ES} - \ WTh_{ED} \qquad \textit{Equation 2.6}$$

- **Ventricle mass:** the ventricle mass (also referred to as myocardium mass), computed by multiplying the myocardium volume by the myocardium density, which normally assumed to be 1.055 g/ml in clinical analysis [18-20], although some recent works have discussed that slightly higher values might be more precise [21]. Ventricle mass is usually only measured for the LV. Normal values for LV mass are 56-140 g and 92-176 g for women and men respectively [13]. The normalized mass by BSA have normal values ranging 41-85 g/m^2 and 49-85 g/m^2 [13].

Out of all these measurements, the EF is overall the most informative and important one as it is a relative index of the contractility power of the LV. In order to measure all these parameters with great fidelity and obtain a robust diagnosis CMRI is usually employed due to its high imaging resolution capabilities, allowing for direct measurement of all these biomarkers.

2.3. Magnetic Resonance Imaging

2.3.1. Physical principles

Spin is a property described by quantum mechanics that is present in subatomic particles [22, 23] and is defined by the presence of an uneven number of neutrons and/or protons. This creates a magnetic moment that when in presence of a strong magnetic field, can be altered by radiofrequency signals whose oscillation frequency matches that of the nuclei in a process known as resonance [22, 23]. This phenomenon is called magnetic resonance.

Within the human body the most abundant atom is hydrogen (^{1}H) due to its abundant presence in water. The spin orientation of the protons is randomly distributed within the body, but when in presence of a strong magnetic field (B_0), they get the same orientation as the magnetic field and start precessing around it at a specific frequency (Figure 2.4).

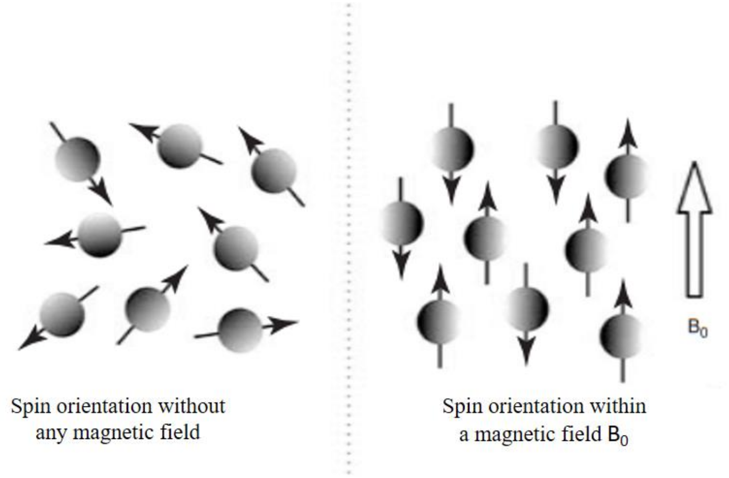

B0
Spin orientation without any magnetic field
Spin orientation within a magnetic field B0

Figure 2.4 Modification of the spin orientation under the influence of a magnetic field. In their natural state the orientations are randomly distributed. When applying a magnetic field, the orientations will align to that of the magnetic field, some pointing to the same direction and others in the opposite, creating a net magnetization vector. Image modified [24].

The moments, although aligned with the magnetic field, do not share the same direction. In this situation, the sum of all the spins creates a net magnetization vector M_0. This vector is the MR signal and its strength depends on the number of spins aligned to the magnetic field, which in turn depends on the strength of this magnetic field.

The magnetic moment can then be excited with a radiofrequency (RF) pulse that will change its orientation in a specific angle (Figure 2.5). After this excitation the spins lose the energy received by the RF pulse in a process known as relaxation. In this process M_0 will return to its original orientation and is during this process that the information that will form the images is acquired.

The are two types of relaxation: longitudinal and transversal. Longitudinal relaxation is the recovery of the longitudinal magnetization direction of M_0. Transversal relaxation is the loss of the magnetization in the M_{xy} plane. Longitudinal relaxation is defined by the time it takes to recover 63% of the equilibrium state, this time is referred to as T1. The transverse relaxation is defined by the time it takes the magnetization on the M_{xy} plane to lose 37% of its value after the excitation [25]. T1 and T2 vary across different tissues within the body, as they are dependent on its chemical composition. The information provided by these times is what is ultimately captured by the MRI systems in order to generate the final images.

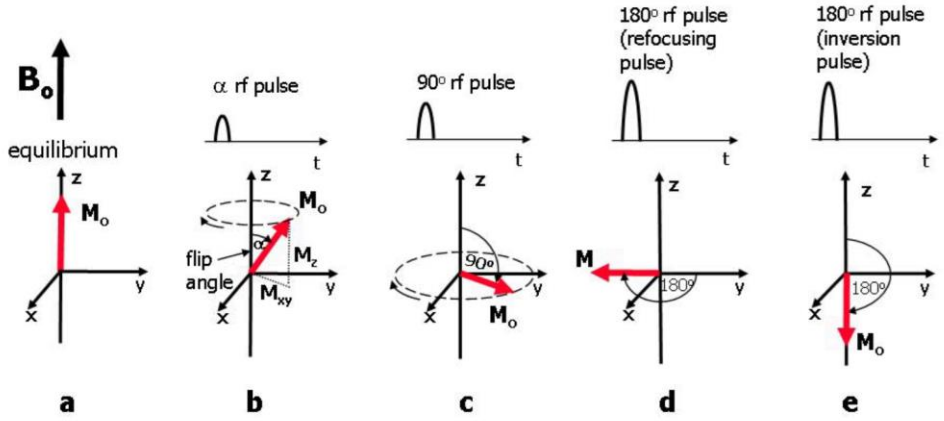


Figure 2.5 Magnetization vector flipping. a) In the equilibrium state the magnetization vector is aligned with the magnetic field, in this state the M_0 only has a component in the z-axis. b) Applying a RF pulse will change the vector's direction with a certain angle (flip angle), at this point M_0 is defined by components in x and y axis depending on the flip angle. The vector then starts to recover its equilibrium state while precessing around the z-axis, the energy loss in this process can be captured as information to reconstruct MRI images. c) Applying a 90° RF pulse makes the vector lie in the xy plane, losing its z-component entirely. d) A refocusing pulse is a 180° RF pulse usually applied at some point when the magnetization vector is recovering after a previous pulse, which allows to instantaneously flip the transverse component. e) An inversion pulse is a 180° RF pulse to the magnetization vector in its equilibrium state, which flips the longitudinal component of the vector [26].

2.3.2. The MRI technology

The MRI system is a scanner machine that uses the physical properties of MR to acquire data to generate images from the inner body. An MRI system has three key elements:

- **Magnet:** produces the magnetic field B_0. It is characterized by the strength of the magnetic field generated, which is measured in Tesla units (T). Generally, the stronger the magnetic field, the higher the contrast and resolution of the images produced. Most MRI machines employ 1.5T in the clinical setting. 3T scanners were initially restricted to research settings, but nowadays they are more available and employed in the clinical practice [27]. MRI with stronger magnetic fields exists, reaching up to 7T [28, 29] and 10.5T [30], but are limited to highly restricted experimental contexts. There is also additional research focusing on developing MRI with even higher magnetic fields [31].
- **RF coils:** in charge of emitting RF pulses to excite the magnetized particles (emitter coils) and receiving the signals produced by the magnetic relaxation (receiving coils). As the MR signals are very weak, a strong shielding is needed to avoid any kind of electromagnetic interference that could affect the signals.
- **Gradient coils:** they introduce spatial variations of the magnetic field along the scanner (gradient magnetic field). This allows for the excitation of specific regions within the scanner. Stronger gradients allow the detection of smaller features within the body, thus improving the spatial resolution of the final images. Three gradient coils are used in order to produce the gradient magnetic field in each spatial direction.

Besides its main components, MRI acquisitions are also characterized by a number of different parameters that mainly define the RF Pulse sequences. These are temporal sequences that define a succession of RF pulses that allows for the generation of different image contrast. As different tissues have different T1 and T2 relaxation times depending on its inner chemical characteristic, specific pulse sequences can be employed to produce relaxation signal that are more predominant on its T1 (T1-weighted image contrast) or T2 components (T2-weighted image contrast) in order to differentiate more some tissue properties than others. The most important sequences are grouped in either of two categories: *spin-echo* and *gradient-echo* [24, 25]:

- **Spin-echo (SE):** SE pulse sequences were one of the earliest developed and is still widely employed. The pulse sequence timing can be adjusted to give T1-weighted, T2-weighted images and proton-density images. The main variables in SE sequences are the repetition time (TR) and the echo time

(TE). All spin echo sequences include a slice selective 90-degree pulse followed by one or more 180 degree refocusing pulse.

- **Gradient-echo (GRE):** These are a widely used alternative to SE sequences. They differ from them in that they employ gradient fields to generate transverse magnetization and flip angles lower than 90 degrees. These additional variables allow for a greater versatility in the sequence design, and this allows GRE pulses to reduce acquisition times of the signals.

2.3.3. Cardiac MRI principles

Cardiac MRI (CMRI) includes all MRI acquisition methods to generate images of the heart to visualize its anatomy and functionality. Compared to other cases, CMRI has the added difficulty that the heart beat moves the tissue relative spatial location in a matter of milliseconds, thus the acquisition speed of the data needs to be fast enough to capture it. Increasing acquisition speed implies reducing spatial resolution and/or losing signal-to-noise ratio [32, 33], depriving of valuable anatomical information with it. To solve this problem image acquisition is usually done retrospectively by coupling and synchronizing the data acquisition with an electrocardiography (ECG) signal of the patient [26, 34, 35]. This method requires that the acquisition is taken in several cardiac cycles and normally in breath-hold conditions, as the lung movement also alter the heart’s motion. The resulting image can either be a stationary image of the heart at a specific time point (still imaging) or a dynamic image where several time steps of the cardiac cycle are obtained to reproduce the cardiac motion (cine imaging) [26]. Still acquisitions are employed to view specific anatomical elements with greater detail (such as coronary arteries) and the images produced by these sequences are generally “black blood” images [26], meaning the intensity of pixels of blood is set to lower values. In contrast cine imaging are employed to evaluate general cardiac function as well as overall anatomic features. Cine acquisitions give the best overall view on the heart’s condition in the clinical setting, as a great number of parameters of the cardiac state depend on the analysis of its motion. As this work is related to the use of cine acquisitions, the following descriptions will focus on them.

There are multiple acquisition protocols in cine CMRI, most of them are based in the SSFP sequence (steady-state free precession), which is a variation of the gradient-echo sequence [26, 36, 37]. These sequences may have different configuration settings depending on the vendor and the MRI machine. Some common ones are “TrueFisp” for

Siemens, “balanced FFE” for Philips, or “FIESTA” for General Electrics. Cine CMRI produce images weighted in both T1 and T2. Specifically, they are generated from the ratio between T2 and T1. The resulting images have both high spatial and temporal resolution (in the order of tens of milliseconds). They present great contrast between the blood pool within the chambers (viewed as white) and the myocardium (viewed as black). As blood has a brighter signal in these images they are also referred to as “bright blood”. The final generated image is a stack of slices (or a single slice) of the heart at different time points or “frames” resulting in a dynamic image that can display the motion of the heart with great quality.

The images can be acquired from different cardiac anatomical planes. These planes are defined from its orientation to the left ventricle’s longer axis, which is the line that connects the apex to the mitral valve [38]. Thus there are a total of three main imaging planes (Figure 2.6), each giving different spatial information.

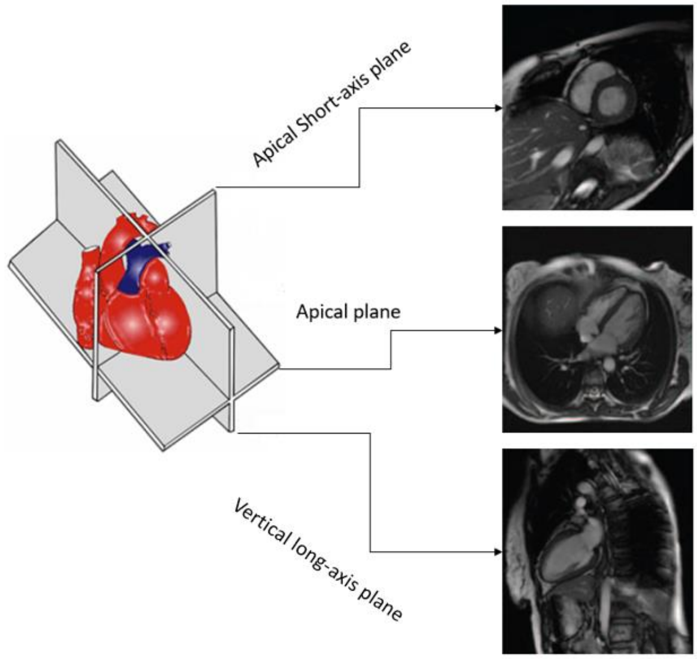


Figure 2.6 Different cardiac planes and their correspondence in cardiac imaging visualization in cine CMRI (bright blood images). Image adapted from [38, 39].

Out of these planes, the short-axis one allows for the view of both right and left ventricles at the same time, this gives a great view of the myocardium contractility along the heart's insides. This plane allows to obtain a great number of functional and structural parameters, and as such is the most employed one to analyze the heart condition as a whole. The image types employed in this work are short-axis cine CMRI.

2.3.4. Short-axis cine CMRI

Short-axis cine CMRI gives the most overall information regarding the heart's function. It allows the dynamic visualization of the right and left ventricles in the same image plane, making them easier to analyze compared to other cardiac imaging planes [40]. The final acquisition consists of a dynamic set of volumetric stacks or frames of the heart comprising the cardiac cycle. This volumetric stack can either contain a specific region of the heart (sometimes comprising only one slice per frame), or a complete view of it. The cardiac image regions are the apical region (close to the apex), mid region and basal region. Within the acquisition some slice that fall outside the heart region may also me be present. Figure 2.7 shows the main regions that are usually imaged, and Figure 2.8 shows a set of different time frames for the same slice including the end-systole (ES) and the end-diastole (ED).

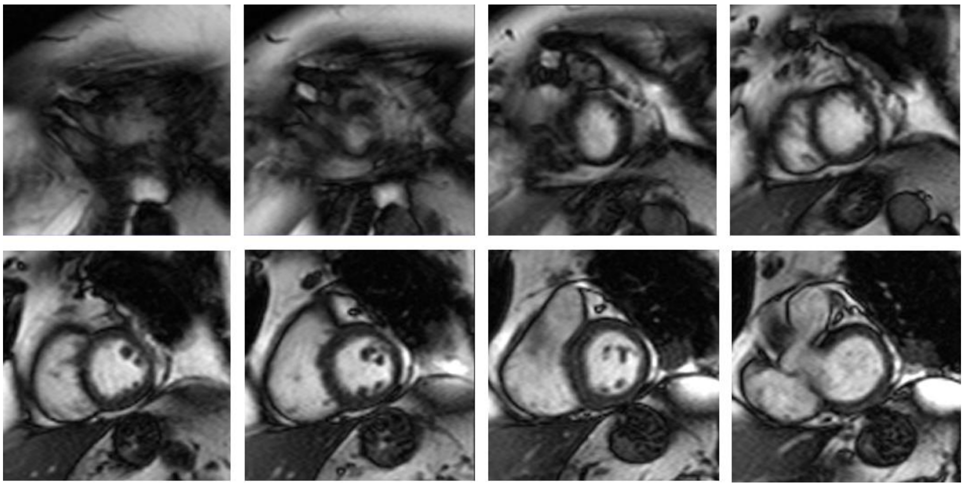


Figure 2.7 Example of slices from a full short-axis CMR. The main regions are present, following the order left to right and up to down: slices outside of the heart (first image), apical slice (second image), mid region slices (third to sixth images) and basal slice (seventh image).

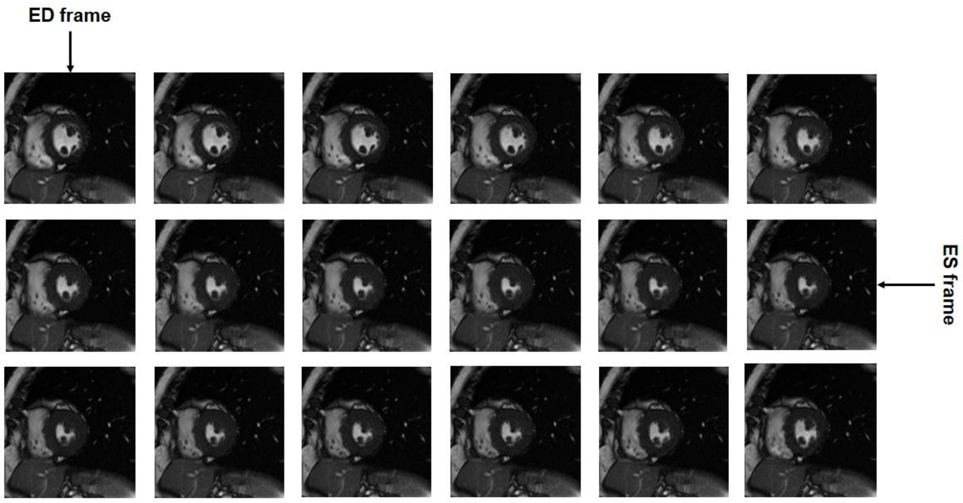


Figure 2.8 Different frames corresponding to the same slice in a short-axis CMRI acquisition. The dynamic contraction of the endocardium is clearly visible. End-systolic and end-diastolic frames of the sequence are indicated.

The short-axis imaging plane allows for the visualization of certain important cardiac elements with great detail. Mainly: LV and RV myocardium and blood pools and the papillary muscles. The LV and RV cavities appears in the image as bright. The myocardium appears as a dark layer surrounding the LV (as a thick layer) and RV (slim layer). The papillary muscles appear as dark stains within the ventricle chambers. Figure 2.9 shows a mid-region slice with all these elements.

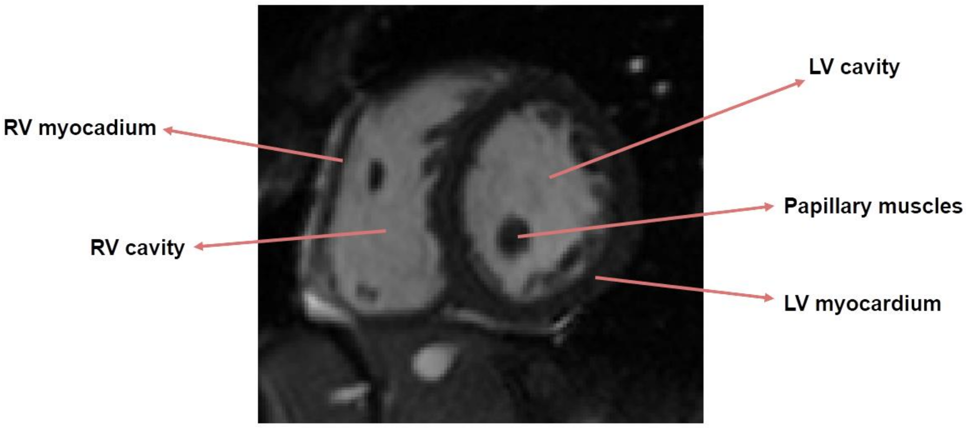


Figure 2.9 Zoom on the cardiac region of a short-axis CMRI image. The image corresponds to a mid-slice of the heart. The different key regions of interest are indicated [41].

Chapter 3. Image processing with deep learning

3.1. Introduction to deep learning

Deep learning (DL) is a subfield of machine learning that has gained an explosive popularity and growth in recent years. The algorithms employed in deep learning are artificial neural networks (ANN), whose design is based on how the human brain learns and recognizes information.

Some of the main reasons on the popularity of deep learning are the high quality on the results offered compared to other ML algorithms and their capacity to work as feature extractors. In most ML systems a previous and meticulous feature extraction process is required to get meaningful inputs from the data to feed the algorithms. This is not the case for ANN, where they can learn to extract important features from the data by themselves as well as to solve the problem handed to them when enough data is available [42]. At the same time, even as powerful as they are, ANN will normally require larger amounts of data to reach their optimum performance (Fig 3.1).

One of the fields that has been greatly impacted by DL is image processing and computer vision. Images are a composition of data values with a spatial relation among them called pixels (or voxels in the case of 3D). This includes the medical imaging field where great contribution has been made.

This chapter briefly covers the advances and important milestones in the DL field, the main components of ANN, their principles of operation and their application to image analysis and to the medical image field.

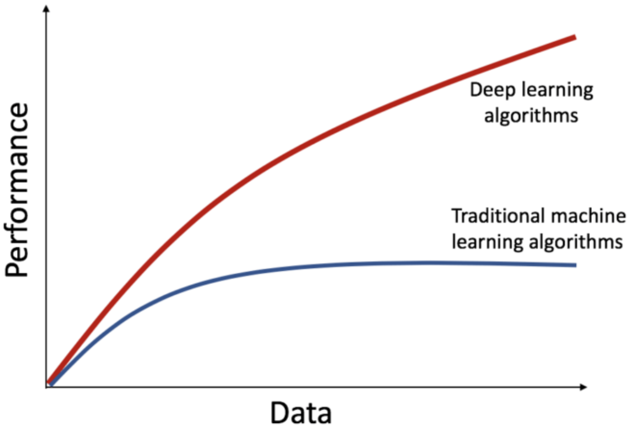


Figure 3.1 Summary graph of the overall differences on performance and data required for classic machine learning and deep learning algorithms. Classic machine learning performance scales with data but its limited by its power capabilities, determining a maximum possible performance. Deep learning may require larger amounts of data to reach high levels of performance, but it is less limited by plateau limits compared to classic machine learning algorithms, which allows for creation of extremely powerful models once enough data is available [43].

3.2. Artificial neural networks

As previously stated, ANN are based on biological NN. This type of algorithmic design was first described in 1943 when the first artificial neuron (AN) based on a basic computational model of a biological neuron was proposed [44]. Since then, a lot of developments have been done in the evolution of ANN. The current design of the majority of AN is based on the perceptron [45]. This first computational model of a neuron applied a linear operation to all the inputs to the neuron through several learnable weights, plus a learnable bias that will remain independent to the inputs. These weights represent the connection and are analogous to the biological synapses. This weighted sum is then transformed with some non-linear function (activation function). The original perceptron applied a step function to obtain binary outputs, but many others have been developed and proposed for different tasks, with different pros and cons. The basic structure of the perceptron is presented in Figure 3.2, and Figure 3.3 displays some common activation functions employed nowadays.

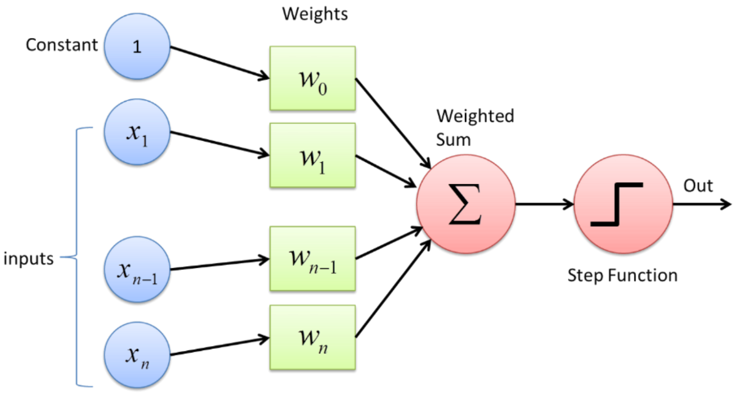


Figure 3.2 Graphic representation of a basic perceptron. The different inputs are each multiplied by a weight and summed. The weighted sum then goes through an activation function (step function) to produce the output [46].

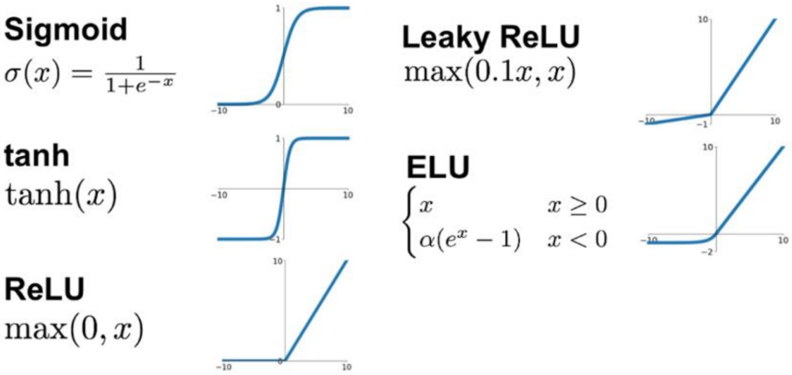


Figure 3.3 Examples of activation functions used in artificial neural networks. Sigmoid is usually employed in the final layer to produce an output between 0 and 1 to represent a probability. Tanh is similar to sigmoid, but its output range extends to negative values. ReLU and its many variants are mostly used in hidden layers [47].

The basic artificial neuron can solve simple tasks, but it is insufficient to solve complex problems, reaching its limit at solving the XOR logical operation [48], although recently it has been proposed that some oscillatory activation functions seem to be capable of tackling this limit [49]. To improve performance, multiple neurons are stacked together in what is called a “hidden layer” where each neuron generates an output, this are often referred to as fully connected layers (FCL). Additionally, multiple layers can be stacked together to generate a full neural network usually known as fully connected neural network (FCNN). These are also referred to as multilayer perceptrons (MLP), even though their units might not be strictly perceptrons (they may not use the original step function as activation) [50]. Figure 3.4 shows a schematic of a fully connected neural network.

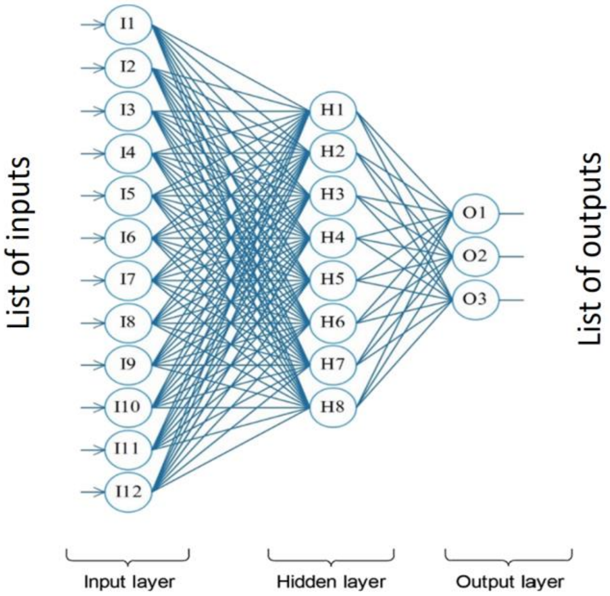


Figure 3.4 Example of a simple artificial neural network. The input layer includes all the inputs, the hidden layer takes the input layer and produces the outputs that are given to the output layer. Many configurations are possible, with multiple hidden layers that have different number of neurons. Image modified [51].

These ANN are already capable of solving many regression and classification tasks, however even though all these algorithms have been around for many decades, they did not catch the attention of the scientific community due to many problems. Among them shined the problems in defining an efficient way to train these models. These changed with the introduction of the backpropagation algorithm [52], a method first described in 1972 [53], but implemented efficiently to train ANN in 1987 [54]. This algorithm is the foundation of ANN training even today. It works by propagating the error from the latest layers backwards throughout the network and assigns a corresponding associated error to every learnable weight. However, even with the addition of this learning method and other important contributions like more efficient activation functions such as ReLU [55], ANN were still difficult to optimize, mainly due to the high computational resources they required, and were still not widely employed until the 2010s when an exponential growth in computational power was made available with the development of cheap and efficient graphical processing units (GPU). This historical pattern can be viewed by exploring the number of publications related to the field. A search in PubMed (which is focused on biomedical science and technology) with the keywords “artificial neural network” shows that until 1990 there are barely any publications on the topic. From that point onwards the interest grows at a very slow and linear pace. This tendency changes dramatically to an exponential growth around 2017 and continues today. A graphic of this trend can be viewed in Figure 3.5.

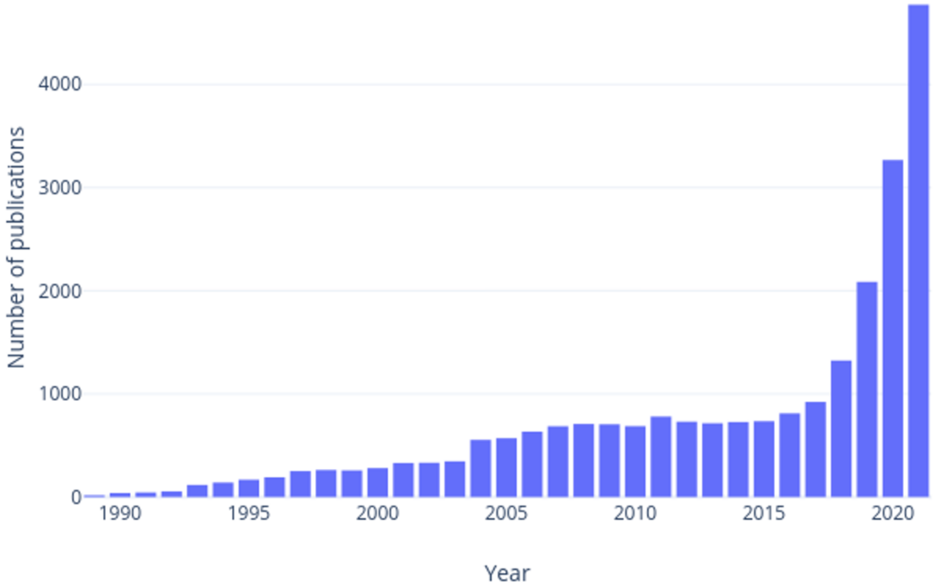


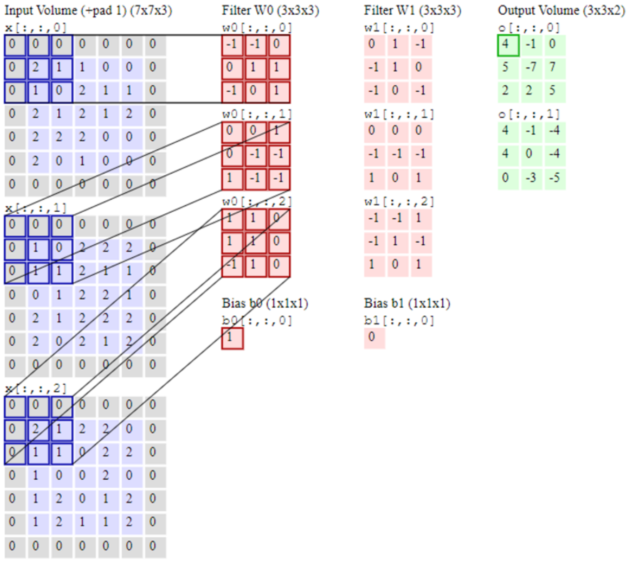
Figure 3.5 Number of publications listed in PubMed (*https://pubmed.ncbi.nlm.nih.gov/*) obtained after searching the keywords “artificial neural network”. Before 1990 there was barely any publication available. In the following decades an increasing linear tendency is visible, reaching 1000 publications in 2017. In the following years (up to 2021) the increase is exponential.

3.3. Convolutional neural networks

Convolutional neural networks (CNN) are a specialized type of neural network specially designed to process images. The neurons from the visual cortex of the brain have a reduced receptive field, meaning that a neuron can only process a certain region that is captivated by the eyes [56, 57]. By overlapping the receptive field of several neurons the full image can be analyzed by the visual cortex.

These mechanisms inspired the design of CNN which employ convolutional and pooling layers as their most important components. They work differently to typical fully connected neural networks. The key concept is that in convolutional layers each neuron only has access to a limited number of features to analyze, instead of the full set of input features. This allows to learn spatial patterns between features as long as they have a spatial relationship between them. The weights in this case are composing a filter that scans all the input feature map and is shared among all the neurons in the layer, allowing to learn spatial patterns that are independent of the location. Each one of these filters also has a number of kernels (or channels) that expand the processing dimension to that of the channels of the input (for example, an RGB image will be processed using filters with 3 kernels, each for each channel within the input). This operation is most like that of classical convolutions where a sliding kernel applies the operations throughout the signal. Actually, the convolution layers defined in most CNN apply the cross-correlation operation, which is equivalent to a convolution but without flipping the kernel [58]. See Figure 3.6 for an example on how convolutional layers work.

By connecting several convolutional layers, a bigger receptive field can be successively reached by deeper neurons. This, in summary, means that the first convolutional layers usually learn small and low-level features, while deeper layers will be able to learn the aggregated patterns that compose more contextual information within the image. This can be expanded more by adding pooling layers. These are simply layers that compress the resulting feature maps into smaller ones, making the next convolutional layers gain higher receptive fields. The most employed pooling layers are the maxpooling layers (which collapses a local region to the maximum value within it), however others also exist like average pooling. Maxpooling is the usual choice since it allows to capture regions which contained big activations and maintain that level of activation when compressing the feature map, while operations like the average could dilute high intensity features if the surrounding has low intensities. However, the final choice will entirely depend on the objective. Figure 3.7 shows a schematic on how pooling layers operate.



Input Volume (+pad 1) (7x7x3)

x[:,:,0]

0	0	0	0	0	0	0
0	2	1	1	0	0	0
0	1	0	2	1	1	0
0	2	1	2	1	2	0
0	2	2	2	0	0	0
0	2	0	1	0	0	0
0	0	0	0	0	0	0

x[:,:,1]

0	0	0	0	0	0	0
0	1	0	2	2	2	0
0	1	1	2	1	1	0
0	0	1	2	2	1	0
0	2	1	2	2	2	0
0	2	0	2	1	2	0
0	0	0	0	0	0	0

x[:,:,2]

0	0	0	0	0	0	0
0	2	1	2	2	0	0
0	1	1	0	2	2	0
0	1	0	0	2	0	0
0	1	2	0	1	2	0
0	1	2	1	1	2	0
0	0	0	0	0	0	0

Filter W0 (3x3x3)

w0[:,:,0]

-1	-1	0
0	1	1
-1	0	1

w0[:,:,1]

0	0	1
0	-1	-1
1	-1	-1

w0[:,:,2]

1	1	0
1	1	0
-1	1	0

Bias b0 (1x1x1)

b0[:,:,0]

1

Filter W1 (3x3x3)

w1[:,:,0]

0	1	-1
-1	1	0
-1	0	-1

w1[:,:,1]

0	0	0
-1	-1	-1
1	0	1

w1[:,:,2]

-1	-1	1
-1	1	-1
1	0	1

Bias b1 (1x1x1)

b1[:,:,0]

0

Output Volume (3x3x2)

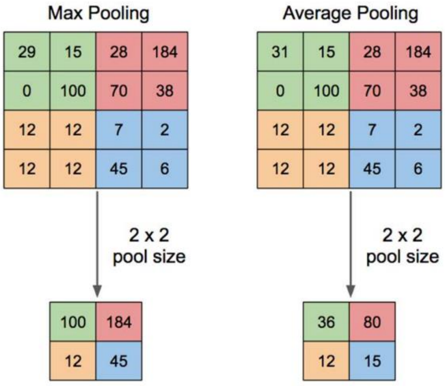
o[:,:,0]

4	-1	0
5	-7	7
2	2	5

o[:,:,1]

4	-1	-4
4	0	-4
0	-3	-5

Figure 3.6 Example of a 2D convolutional layer applied on a 3-channel input. The convolution layer consists of 2 filters (w0 and w1) whose 3rd dimension (number of kernels) corresponds to the number of the input's channels. Each of the filters produces one output channel, in this example the filter w0 and w1 generate the output O first and second channel respectively. Image captured from the demo available at *https://cs231n.github.io/convolutional-networks/*.



Max Pooling

29	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

2 x 2 pool size

100	184
12	45

Average Pooling

31	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

2 x 2 pool size

36	80
12	15

Figure 3.7 Max and average pooling operations. The pooling operation reduces the number of features in the inputs according to the pooling size. Max pooling produces the maximum of the captured values within the kernel while average pooling produces their mean [59].

The mechanisms of operation described allow convolutional layers to generate more abstract and complex features the more the image is compressed along the network. CNN demonstrated its potential after the AlexNet convolutional network significantly outperformed all the rival algorithms at the annual ImageNet competition in 2012 [60, 61]. After this, CNN have kept receiving more attention in the image processing community, including the medical imaging field. This also encompasses cardiac imaging.

These type of neural networks are useful to solve a diverse number of image problems. They can be used for image classification, where usually after the convolutional layers the features are passed to a fully connected layer that will use these extracted features to infer the classification. A similar design can also be employed to tackle regression problems, where the objective might be to obtain a value from the image, for example to calculate a specific biomarker value from a medical image.

An additional key application of CNN is segmentation, which aims to classify each pixel within the image in a class. Segmentation is one of the main topics within the image processing field, and thus is one of the main applications for which CNN are employed. In the case of segmentation, the neural network is a fully convolutional neural network (FCN). As the final output is also an image representing the different labels, this type of networks normally only employ convolutional and pooling layers, and thus their name. Segmentation neural networks have been continuously evolving since the first FCN was described [62] and is one of the most employed architecture types in the medical imaging community, largely due to the fact that most medical imaging analysis involves segmentation in some manner.

Additionally, more complex CNN can be exploited for harder tasks, including image to image translation [63, 64], image enhancing [65, 66], or synthetic image generation [67, 68]. It is also important to note that CNN can also be employed to process 1D data as long as the input variables have some positional relation between them, for example in time series data. These are usually treated with recurrent neural networks (a type of neural network specialized in temporal analysis) with the famous LSTM (long short term memory) [69] or GRU (gate recurrent unit) [70] layers. However, it has been demonstrated that 1D convolutional neural networks can be exploited for these tasks as well [71, 72].

In more recent years a novel type of architecture different from convolutional neural networks has emerged as a competitive substitute, showing even state of the art results and even surpassing CNN in some tasks. These are the transformers [73], a type of architecture that was originally designed for natural language processing, in whose field they have proven to be extremely superior to the previously used recurrent layers. The transformer has also been adapted to be used for image tasks. These are generally known as vision transformers [74], and they have shown a great capacity at treating images, albeit with some limitations, like having more difficulties in implementation with large image sizes and tasks related to dense predictions (like segmentations) [75]. Overall the transformers base architecture focuses on the use of positional encodings and the multi-head attention, a novel way to introduce an efficient global attention mechanism [73]. Although in recent years vision transformers have been extensively employed and studied as a way to surpass convolutional based architectures, the best architectures still employed some kind of hybrid design that incorporate operations innate to CNN [75, 76]. Very recently it was demonstrated that CNN could also be improved by using some of the features that characterize transformers and that CNN can still outperform state of the art vision transformers with the appropriate architecture and training designs [75].

3.4. Considerations in medical imaging

The medical imaging processing field is one that has seen great development in recent years thanks in part to the growth of ANN, and specially CNN. However, the training schedules for DL needs specific considerations that may not be required in regular image problems. This section covers some important factors that need special attention when applying DL model to medical imaging.

3.4.1. Signal intensity

Depending on the image acquisition scanner and protocol the intensity values of the signal displayed in the image may vary. Some imaging modalities like Computed tomography (CT) employ universal scales (Hounsfield units in the case of CT). However, even in this situation different machines may provide slight differences depending on the acquisition specifications [77]. This is more evident in the case of MRI where every different acquisition protocol and/or machine may provide very different signal value ranges.

Neural networks, like any other ML algorithm perform better when its variable inputs are normalized. This applies for the pixel value distributions within the images as well. Very different ranges may hurt the algorithm performance by a great margin.

Additionally, many initialization schemes assign random values to the learnable parameters within certain distributions that are often gravitating around low values close to zero [78]. This also means that inputs with low value ranges will probably allow the network to learn faster. For these reasons, intensity normalization is required before feeding the images to the CNN. There exist numerous intensity normalization methods [79]. Their appropriateness will depend on the nature of the images. Some important examples are:

- **Min-max normalization**: it modifies the range of values within the image to the range 0-1. Easy to interpret but is sensitive to outliers. Only suitable when working with images that come from the same source.
- **Z-score normalization**: it modifies the distribution of values within the image, so it has a zero mean and one standard deviation. This method is robust against outliers but the final range will be less interpretable. Suitable when working with images from different sources.
- **Clip normalization**: these methods clip and set the lower and upper values that pass a certain threshold at some specific level. Usually defined by some low and high percentile of the distribution. After this it is common to apply min-max normalization or Z-score normalization. Using min-max normalization in this setting yields a range of values between 0 and 1 and provides robustness against outliers thanks to the clipping.
- **Histogram equalization**: this type of technique is not focused in the intensity values themselves, but in the histogram of the image. By equalizing the histogram, the result is an improvement in the contrast of the image [80].

3.4.2. Resolution

An important feature of medical images is the resolution they present. Resolution corresponds to the amount of space that each pixel (or voxel) occupies within the real world in every coordinate axis and is defined by the spacing value (distance in mm from the center of one pixel/voxel to the adjacent ones for each direction). As medical images are normally 2D or 3D they will have either 2 or 3 spacing values. Dynamic 4D images also exists, in these cases the fourth spacing value should correspond to the time between images (note that for dynamic images with only one plane per frame the third value will correspond to the time axis, as they will be saved as 3D images).

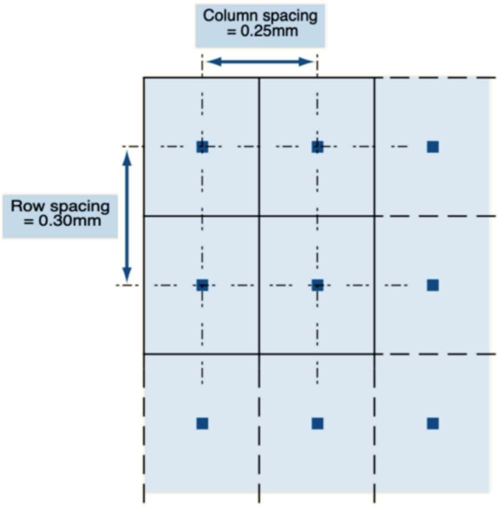


Figure 3.8 Representation of the pixel spacing within a 2D matrix. Different axis may have different spacing, usually indicated in mm. The spacing corresponds to the distance in the real world between the centers of adjacent pixels/voxels [81].

Different acquisitions may have different spacing values. This can also impact the performance of the algorithms, as they are not being fed with consistent voxel information. Additionally, it has also been described that using anisotropic spacing usually yield worse results [82], so whenever possible the spatial resolution difference along each axis should be as close to zero as possible. For these reason standardizing the resolution of the images employed will usually make the CNN perform better. Doing this may require resampling the images of the dataset, the resampling should be made into a resolution space suited for the images employed and will depend completely on the nature of the problem.

3.4.3. Matrix size

The matrix size of the image is also an important factor. Although CNN do not strictly need to be designed to work with specific image sizes, from a practical perspective, most problems require the images to have a constant one, in classification tasks usually the final feature map is fed to a fully connected layer, which require a specific dimension, which in turn depends on the original size of the image. Another situation is segmentation, where the images are downsampled with pooling operations and the final segmentation size is recovered to match that of the original image, this can only be done when the input images have a set size. Still, there are advanced layers that can be used to transform feature maps or vectors into constant sized vectors [83]. Intelligent conditional padding and cropping could also be applied in the case of fully convolutional neural networks, so there are still some design methods that can tackle the size limitation as well.

The matrix size of the image along its resolution determines the whole spatial field of view (FOV) captured within the image, which is another factor whose variability can have an impact in training CNN. As both factors are related, resolution and size normalization are done sequentially. First the image is resampled to get the desired resolution and then the image is resized without modifying the pixel/voxel resolution. This can be achieved with padding or cropping methods. Cropping involves the erasure of image borders to reduce its size. Padding involves adding values along the borders to increase the size. Padding can be accomplished in several ways, including extending the border value, mirror the image border or set a constant value, zero-padding being the particular case where the padded values are zero. Padding is a commonly performed technique, however it should be used with caution, as very big paddings can lead to worse performance [84].

Finally, it is also noteworthy that CNN normally requires large amounts of memory for training. In this setting image size is a crucial factor in memory management and it may limit several things, including the network size (determining its potential fitting capabilities) and the batch size (the number of samples used to train the network per iteration). Normally bigger networks can solve harder tasks [85] and moderate to big batch sizes can improve both performance and training speed [86, 87]. Since 3D medical images are pretty common compared to other image domains, they are often large and this needs to be meticulously addressed in order to get satisfactory results out of the neural networks.

When the image size is too big for the hardware available a common approach is to use patch-based analysis. This is basically the application of the model to certain chunks of the image that can fit in memory and is common for segmentation problems. This process, however should be used with caution, as the FOV changes when taking only a portion of the image, and it should still keep enough spatial information for the CNN to obtain meaningful information. As an example, in short-axis CMRI images, if one applies patch-based analysis for some heart-related segmentation task, the patches should be big enough to capture as much of the heart region as possible in order to keep the most important information. When this is also a limitation, strategies that allow overlapping patches can still lead to improved results [82]. Figure 3.9 exemplifies the concept of patch-based analysis.

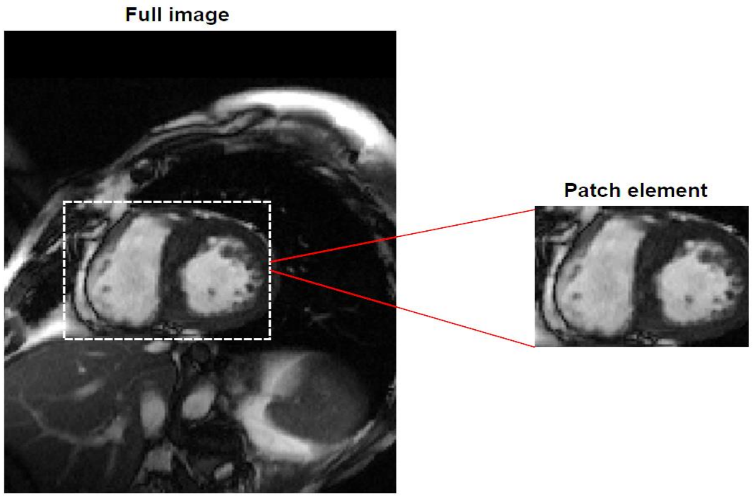


Figure 3.9 Example of patch extraction from a whole short-axis CMRI slice. Applying patch-based analysis makes use of the extracted patches as inputs for the model, instead of employing the entire image. A patch element is defined with a specific size that can extract several sections to use as new inputs. The patch size should be big enough to capture important contextual information. In the case of short-axis CMRI the patch size should allow to get as much of the heart region as possible [41].

3.4.4. Overfitting

Overfitting is a very well-known problem within machine learning. An overfitted model is one that has learned to do the task correctly with the training data, but it is incapable of generalizing well with unseen data [88]. One can relate this to memorizing the training data without learning meaningful information. This process is normally due to either having a very limited dataset or a model that is so powerful that it is capable of modeling even noise within the data and fit it to the objective (or a combination of both conditions).

As DL is a very powerful type of algorithm that employs thousands of hundreds, millions or even Billion parameters [89], it suffers from overfitting to a greater degree, and it is one of the main issues when training deep neural networks. At the same time, medical images are hard to obtain due to them being health-related information, and as such they are strictly protected by regulations. This constrain implies that most medical imaging dataset are not very big. Under those conditions, it is especially important to consider the overfitting problem and take countermeasures. There are several ways to address overfitting, these are known as regularization techniques. Some important regularization methods that involve the model itself are:

- **L1 and L2 regularization**: these methods apply an additional term to the loss function during training. The term is a weighted sum of the value of the parameters within the model. This will force the parameter to get lower values which in turn will reduce the model's power. This penalization may be applied to all or only a part of the model`s parameters. L1 applies the penalization on the absolute value of the parameters while L2 applies it to the squared values. L1 tends to suppress entirely some of the model's parameters while L2 pushes the entirety of parameters to have close to zero values.
- **Activity regularization**: activity regularization consists in applying an additional term to the loss function. In this case, the term is defined by the sum of the neuron's values (the output of a neural network layer). This will enforce a reduction on the activations. This can be applied through L1 or L2 methods. However, in neural networks, it makes more sense to use L1 since this will suppress entirely some of the activations, which will give a more relative importance to the remaining ones and thus enforce the network to learn more compact and representative features.
- **Early stop**: early stop is another classical method to avoid overfitting. In this case a small part of the training set is separated for validation (validation set). The quality of the model is evaluated on the validation

set after each epoch with the validation set and if it starts decreasing after some iterations, the training is halted, as this loss in quality is a common trait when overfitting start to happen [90, 91].

- **Max-Norm normalization:** this technique consists of applying a constraint to the weights associated to every neuron. This is done by clipping the weights after each training step if the L2 norm of the vector weights exceeds a previously specified value.
- **Batch normalization:** batch normalization (BN) [92] normalizes the features after each layer using the values of different instances within the batches during training. It works similarly to Z-score normalization in the sense that it scales the activation features for them to have a zero mean and a standard deviation of 1 across instances. BN has some additional parameters to enable its application at inference times (where only a single instance may be used within the batch). In the case of CNN, the normalization happens for each feature channels across all available instances. Batch normalization is additionally helpful to improve convergence speed.
- **Dropout:** dropout is an important method for regularizing neural networks. During training, a layer with dropout will "switch off" some of its neurons randomly based on an established probability. In practice, this makes the training work as a model averager, as only random portions work at each step. It can also be applied during inference, making the model a system for Bayesian inference [93]. In convolutional layers they are rarely employed compared to FCL. This is because switching random elements from the feature maps can impact the convolution's ability to extract spatial patterns. To address this some other forms of dropout, exist. Dropblock for example applies the dropout in chunks within the feature map, thus suppressing entire small areas [94]. Another popular method is the spatial dropout which applies the dropout randomly to entire feature maps [95].

Apart from these, there exist more methods to limit ANN in their learning capabilities. Besides these regularization methods, treating the dataset itself in specific manners can also help leverage the overfitting problem as described in the following sections.

3.4.5. Dataset size and variability

As stated at the beginning of the chapter, one of the advantages of neural networks, and more specifically convolutional neural networks, is their incredible performance capabilities. However, the counterpart of this is that they require large datasets that are a good representative of the problem's real distribution.

For training these algorithms it is recommended to employ as much data as possible, however this is an important limitation in medical imaging in various senses. First, access to the data is hard and legal requirements need to be addressed as they are medical information which is strictly confidential and protected by law. Second, even if one has access to a certain hospital database, the usual is to solve a very specific problem. If the problem involves targeting some specific diseases it is most probable that the final data available will be very unbalanced with respect to healthy subjects and/or other conditions. Lastly, most images saved in hospitals are not annotated to be employed to train machine learning systems. The labeling process (either segmentation, classification, or any other kind) will require a clinical expert doing it manually, such task may be hard and time consuming, with few people desiring to do this work.

As explained, there are a lot of difficulties in applying ANN to medical images, however some public datasets still exist and access to other data sources is possible (although slow) provided that a legal course is taken. In general, the key factors to consider would be acquiring a dataset as large as possible and selecting the images meticulously so that the distribution of conditions present within them match that of the problem to avoid data imbalances. Additionally, these images will need to be labelled by experts to ensure quality matching that of knowledgeable clinical experts on the problem.

When working with image datasets it is also important to consider the possible unbalance of categories to predict. The presence of imbalances can potentially lead any learning algorithm to be biased towards one of the categories. There are many ways to compensate this. Some of them are the use of weighted loss functions that compensate on the category's imbalance, oversampling (increasing the number of cases of the underrepresented categories) and undersampling (removing cases from the overrepresented categories). Oversampling and undersampling should be used with cautions and meticulously, as with undersampling we are effectively removing valuable information and with oversampling we could induce overfitting by duplicating cases. In the case of oversampling, it is good idea to combine it with data augmentation techniques in order to modify the replicated images and help avoid overfitting.

3.4.6. Data augmentation

As one of the main problems in medical imaging is the number of samples available, data augmentation plays a key role in the process of training CNN. Data augmentation in the image domain encompasses all the methods focused on creating artificial image instances to enlarge the dataset.

Data augmentation is applied in a variety of ways, but the most usual is to apply transformations to the images available, including any type of affine transformation (which includes translations, rotations, shear and zoom), mirroring the image, adding certain types of noise, modifying the contrast, applying elastic deformations, etc. These transformations need to be applied in ways that modify the original image enough to make it considerable different from the source, but at the same time they should keep consistency with the source, especially in the case of medical images where normally the tissues and organs imaged have some natural shape that should not be lost in the process (for example in short-axis CMRI mirroring would reverse the relative location of the LV and RV with respect each other. An example of some transformations applied to short-axis CMRI can be viewed in Figure 3.10.

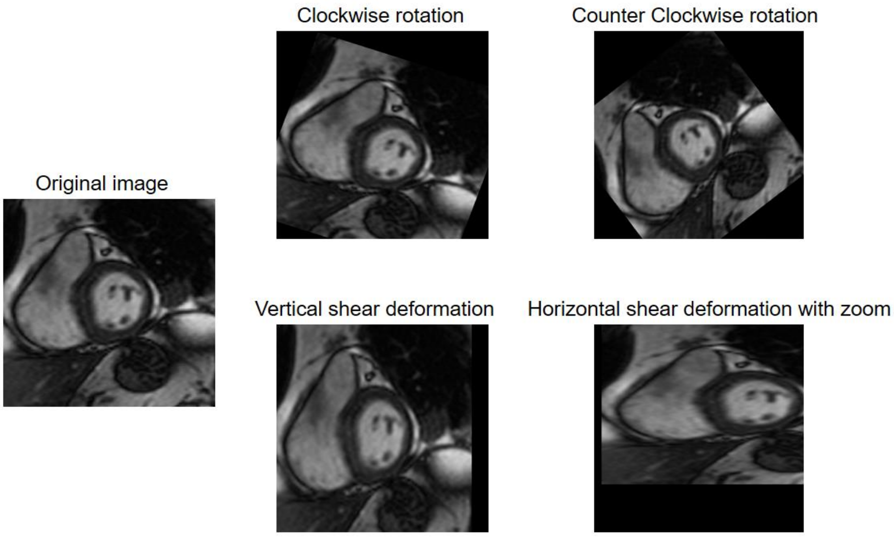

Clockwise rotation
Counter Clockwise rotation
Original image
Vertical shear deformation
Horizontal shear deformation with zoom

Figure 3.10 Examples of different image transformations that serve for data augmentation purposes. A single transformation may be applied but mixing different transformations will increase the variability of the new instances [41].

Besides the referenced imaged transformations, there exists other novel techniques to address data augmentation. Generative neural networks are a type of ANN that can generate new synthetic data samples based on a previously seen dataset. This has been accomplished with great results with variational autoencoders (VAE) [96, 97], generative adversarial networks (GAN) [98] and diffusion model [99]. Some works have already demonstrated the potential of synthetic medical images to improve data augmentation [100].

Chapter 4.
Materials

4.1. Dataset

For the experiments performed on this thesis the data employed consisted on short-axis cine CMRI sequences. The images were acquired from the Unidad de Resonancia Magnética (ASCIRES) del Hospital Clínic Universitari de València (València, Spain). All the patient's whose images were used had previously given written consent to be used for the studies, which was approved by the hospital's Medical Ethical Committee. All images were anonymized while acquiring them from the hospital's PACS system. Images were in DICOM format, with .IMA extension. This is a special type of DICOM extension employed in some MRI scanners from the SIEMENS brand.

The images are comprised of 4D stack of cine short-axis acquisitions with the image FOV focused on the LV and RV. The dataset comprised a total of 399 image volumes from 399 different subjects. The demographic distribution was of 272 men and 127 women, with age 64.51 ± 12.35 years (63.28 ± 11.97 years for men, 67.42 ± 12.75 years for women) (mean ± standard deviation). There were both healthy cases and cardiac patients. A diversity of pathologies was found within the dataset, the most predominant being myocardial fibrosis, necrosis, ischemia and LV systolic dysfunction (ejection fraction lower than normal and/or regional wall motion abnormalities). Healthy patients were defined as those without risk factors or previous conditions, normal ECG readings and normal cardiac MRI parameters. The pathology distribution of the patients is summarized in table 4.1.

Table 4.1 Diagnostic classification for all cases available in the dataset collected.

Categories	**Number of cases**
Normal cases, no pathology	*48*
Presence of necrosis	*14*
Presence of fibrosis	*12*
Presence of ischemia	*10*
Functional affection of LV (ejection fraction lower than normal and/or affected segmental contractility)	*23*
Functional affection of RV (ejection fraction lower than normal and/or affected segmental contractility)	*2*
Functional affection of LV and RV	*137*
Functional affection of LV and presence of fibrosis/necrosis/ischemia	*45*
Functional affection of RV and presence of fibrosis/necrosis/ischemia	*4*
Functional affection of RV and LV and presence of fibrosis/necrosis/ischemia	*95*
Other cases that do not fall in any other category	*9*

Imaging was performed in every case under breath-hold conditions using a 1.5T MRI scanner (Sonata Magnetom, Siemens, Erlangen, Germany). Image characteristics were obtained by analyzing the files metadata contained on the DICOM headers. The general specifications of the acquisition protocol were: flip angle: between 49 and 58 degrees (with the vast majority having 58°); repetition time: between 51.66 and 56.80 ms; echo time: between 1.25 to 1.34 (with the vast majority being at 1.25). The image in-plane resolution varied among the cases, ranging from 0.57 × 0.57 mm^2 to 1.09 × 1.09 mm^2. The slice resolution was constant with a slice thickness of 7 mm and a spacing between slices of 3 mm. Image matrix sizes varied from 144 × 144 to 256 × 256, being the latter the most common size. The number of slices was variable as well, with a range from 8 to 14. The number of temporal frames and the spatial resolution in each sequence was not constant in the dataset, the vast majority had 35 frames (366 cases, 92% of the dataset) with a constant temporal resolution of 0.023 s. The remaining cases had a number of frames between 14 to 25 frames with a temporal resolution between 0.062 and 0.078 depending on the case.

The images had been labeled by 2 expert cardiologists with more than 15 years of experience. The labels consisted of painted contours of the RV (endocardium, leaving inside the blood pool chamber) and LV (endocardium and epicardium, leaving inside both the myocardium and the blood pool chamber) on the ED frames. For the ES frames label information differed in that the epicardium contour was not included, leaving only the LV and RV blood pool chambers. These contours represent the segmentation boundaries of the different regions of interest at both maximum relaxation and contraction and were generated semi-automatically with the help of the software Syngo.via version SYNGO MR A30 4VA30A from SIEMENS (*https://www.siemens-healthineers.com/es/magnetic-resonance-imaging/advanced-imaging-applications/syngo-via*). The segmentation's information was coded within the files as DICOM overlays, which could be accessed through the processing of the "overlay data" header (header value |6000, 3000|). After thoroughly exploring the dataset, it was found that 2 out of the 399 did not have segmentation data for the ES. These cases were still employed in the experiments described in chapter 5, but were discarded for the ones in chapter 6 and 7. The removed cases corresponded to the category of "Functional affection of LV and RV" from the men's group. A sample view of both ES and ED with the available segmentation of one case from the dataset is presented in Figure 4.1.

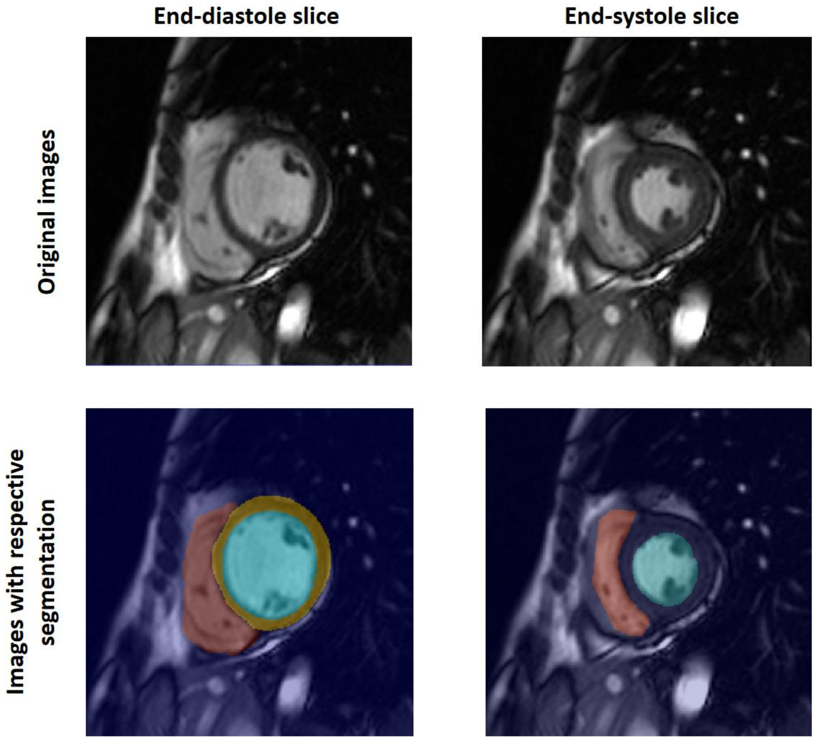


Figure 4.1 Case example of the used dataset. A single mid regional slice from both the end-diastole and end-systole is presented. Above, the original images, below the segmentations available overlapped with the images. For the End-systolic frames no myocardium label was available. Segmentation colors: red for right ventricle cavity, yellow for left ventricle myocardium and blue for left ventricle cavity.

4.2. Hardware

4.2.1. Context

Before describing the hardware employed, it is important to understand the industrial and technological context under which most ANN, and more specifically CNN fall. Specialized hardware is required to train convolutional neural networks. More concretely, they require one or several powerful GPUs to train them in reasonable time frames [101, 102]. The GPU is the key component to train an ANN, as they are specially designed to parallelize multiple basic computations, making them suitable to train these incredibly big models. The remaining components within the computer can have a significant impact on the general training performance too, creating bottlenecks to the GPU calculations, so a decent amount of RAM memory and a powerful enough CPU are recommended as well.

Additionally, nowadays most of DL libraries for development only function properly with the GPU brands from Nvidia Corporation (Santa Clara, California, U.S) so most developments are limited to using this brand. Nvidia offers different GPU series, mainly the GeForce series, the Quadro series and the Tesla series. The GeForce series are more focused on the gaming industry, with the best price/quality relation. The Quadro series was developed for the 3D design industry and are usually more stable than the GeForce, but are considerably more expensive, mainly due to a more dedicated support in their driver's software, although their technical hardware specifications do not differ significantly from the GeForce series. Finally, the Tesla series are the most powerful and expensive, these GPUs are designed for high-computing scientific problems, being deep learning one of the main fields they are used for. All these series can be efficiently employed for deep learning acceleration.

In recent years Google LLC (1600 Amphitheatre Parkway, Mountain View, California, U.S.) has also developed the so called TPUs (tensor processing units), which are a very specialized type of hardware similar to the GPUs but originally designed to work with deep leaning. Up to today these machines are normally only available for cloud computing.

With respect to DL software packages, the two currently dominant frameworks are Google's own development library TensorFlow (*www.tensorflow.org*) and Pytorch (*https://pytorch.org/*), the latter being developed and maintained by Meta's AI Research lab (Astor Place, New York City, New York, US). Both are mainly employed in Python programming language, although they are available in others like C++, JAVA or R.

4.2.2. Experimental equipment

For all the experiments the same computer was employed. The most important components of the computer are:

- CPU: Intel® Core™ i9-9900K (3.6 GHZ). Full specifications at: *https://www.intel.es/content/www/es/es/products/sku/186605/intel-core-i99900k-processor-16m-cache-up-to-5-00-ghz/specifications.html*.
- RAM: 64 GB of DDRM4

- Operating system: Windows 10 Pro
- GPU: Nvidia GeForce RTX 2080 Ti (11 GB of VRAM GDDR6). Full specifications at: *https://www.techpowerup.com/gpu-specs/geforce-rtx-2080-ti.c3305*.

As stated in the previous section the GPU is the most important component, in this case the model RTX 2080 Ti has both great computational power and a more than decent amount of memory. This GPU was one of the best models before the 3000 series was launched in 2020. However, it is still limited by the amount of memory offered, so very big models, or big images can still be problematic if one desired to employ moderate-to-big batch sizes. In the following chapters the training setting’s limitations described came specifically from this feature.

4.3. Software

4.3.1. General setting

The computer’s operating system installed was Windows 10 Pro. For all the experiments the programming language employed was Python 3.7.6. The package manager employed was conda (*https://anaconda.org/anaconda/conda*), installed through the anaconda distributions (https://www.anaconda.com/). The choice for anaconda was due to easier installation of some required software for GPU-acceleration. Employing conda avoids manual installation of several software components as the manager can install those for the user. For scripting the IDE PyCharm (*https://www.jetbrains.com/es-es/pycharm/*) was chosen due to its easy use, variety of useful tool and flexibility.

Regarding the medical image file formats, the original source was all in DICOM with .IMA extension. All the images and their respective segmentations were converted to the NIfTI format (*https://nifti.nimh.nih.gov/)* which employs the .nii extension. The conversion was done to make the whole processing and management of the images easier, as in DICOM every slice within the acquisition is saved in a different file, while in NIfTI all the data is saved in a single file.

4.3.2. Deep learning software

Design, implementation, training and inference processes with ANN were programmed using TensorFlow 2.1 (*www.tensorflow.org*, Google Brain, Mountain View, CA) using its Keras API. It should be mentioned that at the time of these experiments there existed two TensorFlow libraries, one to work with CPU and another for GPU-acceleration (TensorFlow-GPU). The latter one was used.

To enable the use of the TensorFlow library additional software requires installation. The software versions required depends on the library version and the GPU model. The package manager conda is capable of installing these packages automatically. The two most important ones are CUDA and cuDNN, both developed and maintained by Nvidia. CUDA (*https://developer.nvidia.com/cuda-zone*) is a software toolkit for GPU computations that allow to interface with the GPU hardware, on the other hand cuDNN (*https://developer.nvidia.com/cudnn*) is a library developed by Nvidia that includes several required functions for the use of GPU-accelerated ANN.

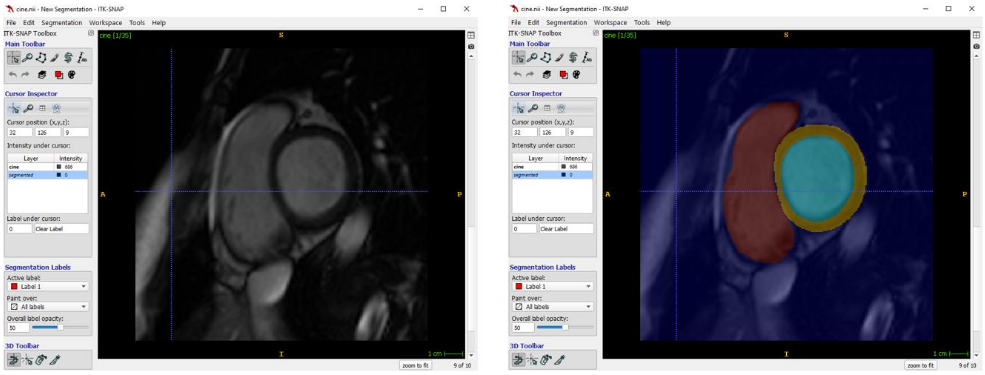
4.3.3. Miscellaneous

Besides all the software already described, other software and Python libraries were used during this thesis.

The main Python libraries used to preprocess, organize and visualize the data were:

- Numpy (*https://numpy.org/*): library used to manage and operate the image and data arrays.
- Sci-kit image (*https://scikit-image.org/*): library with several image processing functions. Used for several preprocessing and image transformation steps.
- Pydicom (*https://pydicom.github.io/*): library used to manage DICOM files, used for metadata analysis and to extract the overlayed segmentations.
- SimpleITK (*https://simpleitk.org/*): library used for general medical image managing and processing. Used for managing and converting the images from DICOM to NIfTI and for resampling operations.
- Matplotlib (*https://matplotlib.org/*): library employed for quick visualization of data within Python. Used for image inspection within the python environment.

Besides the Python related software, for visualization of the medical images the visualizer software ITK-snap (*http://www.itksnap.org/pmwiki/pmwiki.php*) was chosen. This software allows to use several medical image formats, including DICOM and NIfTI. This visualizer was used to explore the full 4D CMRI and their segmentations. A view example of its interface with a short-axis CMRI along its segmentation is presented in Figure 4.2.



Cursor position (x,y,z): 32 126 9

Intensity under cursor:

Layer	Intensity
cine	686
segmented	0

Label under cursor: 0 Clear Label

Active label: Label 1

Paint over: All labels

Overall label opacity: 50

Figure 4.2 Visualization of short-axis CMRI acquisition within ITK-snap visualizer. The visualizer allows to inspect voxel values, navigate through the different temporal frames, and overlap the segmentation data among other available functions.

Chapter 5. Automatic semantic segmentation

This chapter is based on a conference paper published in the context of this thesis [124]. The paper is available at: *https://doi.org/10.1109/BIBE50027.2020.00177*.

5.1. Introduction and motivation

Semantic segmentation is the task of classifying every pixel (or voxel) within an image into a specific category. This is a usual step in most medical image processing problems. Segmentation allows for the delimitation of specific regions of interest that can then be analyzed to generate radiological biomarkers. As such, it is a very important task and also the main bottleneck for a complete radiographic analysis due to the high time it can take to do manually.

In the context of short-axis cine CMRI it is necessary to segment various regions of the heart in order to characterize it. Additionally, the segmentation in these acquisitions must be obtained at least at two time frames: end-diastole (ED) and end-systole (ES). This is required in order to derive some of the major cardiac function biomarkers (see chapter 2, section 2.2.2) from which the ejection fraction is probably the most informative one. Obtaining the LV ejection fraction requires to measure the volume occupied by the LV blood pool chamber in both ED and ES.

Besides the Ejection fraction there are other parameters of interest, which in turn may require specific segmented regions at a specific contraction step or the full segmentation at both ES and ED. In general, in the clinical practice the main biomarkers only require the segmentation of the LV and RV inner blood pool at both ES and ED and the LV myocardium at ED. However, it should be mentioned that depending on the patient's condition the diagnosis may require additional segmentations. As a representative example, it is common to label the papillary muscles alongside the blood pool chamber as just one label, however in order to be more accurate the segmentation would require to separate these in two different labels. This is also the case if the diagnosis is focused in the papillary muscles as well. Figure 5.1 shows an example of both the usual segmentation for short-axis CMRI analysis and how it should be in order to obtain a full and more accurate analysis.

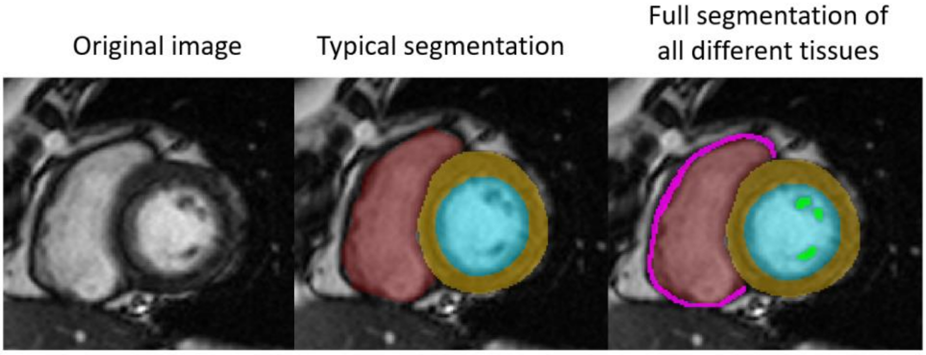


Figure 5.1 Example of typical segmentation and full segmentation on the main regions of interest in a short-axis CMRI slice. The typical approach used in research and on the clinical context is to only segment the ventricle's cavities (including the papillary muscles) and the left ventricle myocardium. In the full segmentation approach, the papillary muscles of the left ventricle are segmented as a different label and the right ventricle myocardium is also segmented [41].

This chapter covers the experiments done in this problem setting. Since both ES and ED are very similar in the image plane and only the ED had all the three main region's manual segmentations, the experiments mainly focused on the ED frames. The experiments involved designing different CNN architectures and see their performance at solving this task. More specifically the task was to check how well a new implementation design of the famous 2D U-net [103] could compare against a classical 3D version of it. Additionally, the same CNN that had been trained only on ED frames were tested against ES frames in order to check how well they captured the inner abstract information within the images.

5.2. Related work and state of the art

5.2.1. Convolutional neural networks for semantic segmentation

Semantic segmentation is one of the main tasks for which CNN are employed. As such there has been many works describing different architectures to tackle segmentation problems. Within them, probably the most influential one has been the U-net [103], especially in the biomedical imaging sector. Noteworthy at the moment of writing this thesis the original paper has more than 48000 citations (provided by google scholar metrics).

It is important to mention that there exist other architectures besides the U-net, some examples are the original FCN [62], Dilatednets [104], the different DeepLab versions (1, 2, 3 and 3+) [105] or the Mask-R-CNN [106] to mention some. All these different CNN are based on different paradigms but all of them have demonstrated to be capable of solving segmentation tasks. FCN for example was the first proposed CNN to tackle the problem of segmentation [62], being the first one, it has been largely surpassed by more advanced architectures. Dilatednet is a type of architecture that makes extensive use of dilated convolutions to improve context aggregation in each consecutive layer [104]. The Deeplab architectures are an improvement of the FCN and among their main features is the use of dilated convolutions and multiscale processing [105]. On the other hand, Mask-R-CNN is a special type of segmentation CNN where it employs a previous object detection architecture, the R-CNN [106] and then applies the segmentation layers on the detected objects. Mask-R-CNN actually tackles the problem of instance segmentation where the objective is not only to segment different objects, but also to differentiate between different examples of the same object category [107]. This allows to distinguish different objects that pertain to the same category that may overlap within the image. This type of problem is not that common within the medical image field compared to other image-related fields (i.e vehicle segmentation on traffic images or pedestrian instance segmentation).

As previously stated the original U-net has become the most popular fully convolutional neural network for medical imaging segmentation. The great majority of segmentation problems are tackled with this architecture or with variations of it, however, the architecture’s core remains in all these variations. All U-net-like architectures are based on an encoder-decoder architecture. In the encoder the image passes through different convolution layers and pooling layers to reduce the feature maps size and increase the field of view. After reaching the bottleneck, the feature maps are upsampled, normally by means of transposed convolution layers (also called up-convolutions) [108] followed by more convolutional layers. The U-net also includes skip-connections that passes the information from layers of the encoder to the layers in the decoder, allowing to recover information that could have been lost. Figure 5.2 shows a schematic of the basic U-net design.

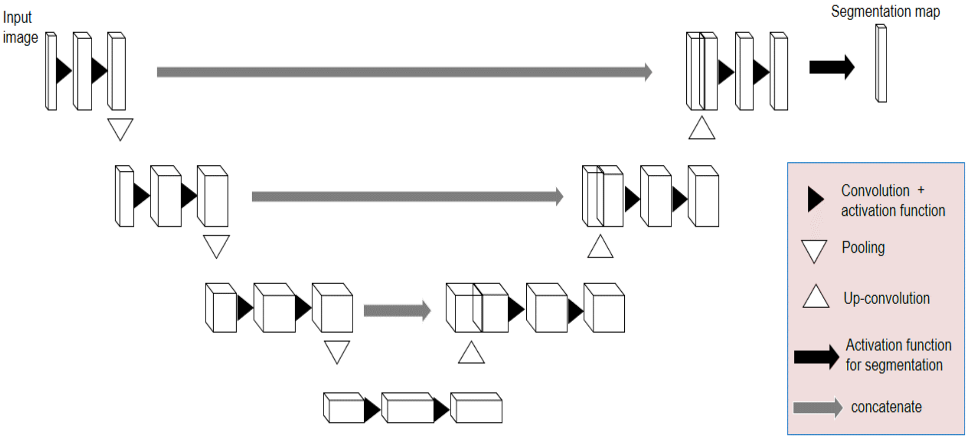

Input image
Segmentation map
Convolution + activation function
Pooling
Up-convolution
Activation function for segmentation
concatenate

Figure 5.2 Schematic of the general architecture of the U-net. In the first part the input is passed thorough different convolutional and pooling layers until reaching the lower bottleneck. Then the input’s size is recovered employing up-convolution operations, several convolutional layers are also applied during this process. Additionally, skip connections pass feature maps from the downsampling path to the upsampling path to recover spatial information that could have been lost. The skip connections are usually applied via concatenation, but other operation like summation can also be applied.

During the years many variations of the U-net that improved its performance have been described. Some notorious ones are: the 3D U-net [109] which uses the same idea but making use of 3D convolutional layers; the V-net [110] whose most important changes were introducing the use of residual functions [111] in each convolutional layer and the addition of strided convolutions instead of pooling operations; attention U-net [112] which makes use of attention modules that help the network focus on the regions of interest [113, 114]. These are only some of the many variations of the U-net that exist. However, a recent study [82] demonstrated that overall the original U-net architecture with some simple changes, which conforms the nn-Unet (no new Unet), can in fact outperform most complex versions of it and other segmentation architectures in a great number of medical imaging segmentation problems. This study also focused its attention in how a good preprocessing of the medical images is a key factor at determining the quality of the results and the authors designed an automated pipeline that applied this preprocessing based on general rules. They additionally gave importance to the use of 3D convolutions when processing 3D images and the use of patch-based analysis. This is in fact an important milestone in U-net research, as even though some of their proclaims were well known to improve performance (i.e., intensity normalization schemes, sufficient field of view, standardized resolution, etc.), they effectively demonstrated them employing a simple version of the U-net (3D or 2D) in very different medical image modalities and segmentation tasks.

5.2.2. Previous research in short-axis cine CMRI segmentation

There have been many works involving the use of CNN for short-axis cine CMRI segmentation in recent years. There are some works which specifically focused on a certain tissue to segment, while others target the usual three key regions: LV and RV cavities and LV myocardium. In general, the LV cavity has presented the higher quality scores in the majority of works, while the LV myocardium is usually the region with worst quality results. Additionally, ED frames seem to obtain better results compared to ES, except for the myocardium which in some cases has the quality tendency reversed. The explanation for these is easy, first the LV has the more stable shape, being it a round element, and second in the ED the LV and RV are bigger and present a less deformed shape, so they are probably easier to correctly segment by CNN, this also applies for the LV myocardium, whose space within the image plane is bigger in ES relative to the LV cavity compared to ED, thus making sense that better results are obtained in ES for this region.

In order to apply segmentation neural networks to this task the usual approach is to train a neural network for the ED segmentation and another one for the ES segmentation, however exceptions to this methodology exists. It should also be considered that different works have measured their algorithm’s performance with different metrics. These include derived volume estimations, relative volume errors and segmentation overlap quality scores. However, most of the works use the Dice coefficient (DC) [115] to measure the segmentation overlap quality, being it a typical quality measure in medical image segmentation problems.

In the work of [84] the authors tested the performance of a basic U-net architecture under different training configurations targeting only the LV in both systole and diastole in 2D images. They tried various pre-processing methods like intensity normalization on the images, applying an initial region of interest (ROI) and zero-padding. Data augmentation was also applied to test its benefits. They obtained very accurate results with DC around 0.95 for both systole and diastole. Their major conclusions were that the use of weighted loss functions was necessary to obtain high-quality results, that using data augmentation and applying intensity normalization allows for better segmentations and that an excessive zero-padding worsens the network performance.

Another work aimed to segment the LV and RV cavities and the LV endocardium using a novel neural network which they named Rianet [116]. This network was a typical U-net that incorporated specialized attention blocks. The attention blocks incorporated only information of the higher and lower resolution in the contracting path, and the result was then passed through the skip connection. This was a different approach to the original attention mechanism where the attention map was produced by incorporating both information from the contracting and expanding paths. This architecture consisted of two sub-network that followed the described structure. The first one was trained to detect the region of interest within the image and then crop it. Then, the cropped image was again used as input for the second network to obtain the final segmentation. They obtained average DC of 0.94 for the LV cavity, 0.92 for the RV cavity and 0.91 for the LV myocardium.

An uncommon and interesting approach was taken in [117]. In this work the authors implemented a U-net with recurrent layers to segment the LV cavity. Recurrent neural networks (RNN) are well known for time series analysis. The most employed layers in RNN are the LSTM [69] and the GRU [70]. These have been described and extensively employed to process time-related series. In the referenced work they use a GRU layer at the bottleneck of the U-net. In this context the function of the recurrent layer was not to find temporal relationships, but spatial relationships that were correlated between adjacent slices in the short axis volume, propagating the information throughout all the slices for segmenting the whole stack. They reported DC results of 0.90 and 0.93 for two different datasets.

The nn-Unet was also employed in the ACDC challenge dataset [118] (*https://acdc.creatis.insa-lyon.fr/description/index.html*), whose aim is to segment the main cardiac regions in short-axis cine CMRI. The dataset includes a total of 100 cases for training and 50 for the testing. The dataset, although relatively small is varied, with 2 different scanner acquisitions (of 1.5 and 3T both from Siemens brand), resolutions varying from 1.37 to 1.68 mm^2/pixel, slice thickness of either 5 or 8 mm, and a number of frames per acquisition ranging from 28 to 40. The full details are in the challenge's website. At the moment of writing this thesis the nnU-net has reported the highest score on the 50 evaluated cases, with DC of: LV of 0.967 (ED) and 0.928 (ES), RV of 0.946 (ED) and 0.904 (ES), myocardium of 0.896 (ED) and 0.919 (ES). These results had a considerable quality and it should be kept in mind that the nn-Unet uses an automatic pipeline to determine its best training configuration based on a relatively basic U-net architecture.

All the previous works described were done employing specific datasets, however more recent works have also been done in order to apply segmentation with CNN with different datasets coming from different centers and machine sources in order to solve the task in a more general setting that could be extended to any image acquisition source.

In the work described in [119] they employed a 2D U-net and trained it in three different settings: with images from the same center and same manufacturer scanner, with images from different centers but from the same manufacturer machine (multicenter setting) and with images from different centers and different manufacturers (multivendor and multicenter setting). To evaluate the performance, they tested each trained U-net on a dataset coming from a different vendor and center than those used for training. In this case the target was the endocardium and epicardium contours of the LV. They showed that when employing more variable data that came from different settings the segmentations improved. The network trained with a multivendor and multicenter setting achieved the best results with average DC of 0.88, 0.95 and 0.93 for the apical, mid and basal regions of the endocardium respectively, and 0.91, 0.96 and 0.94 for the same region in the epicardium. All the data underwent a preprocessing that include intensity normalization to set all intensity values within the same range, image cropping in order to ensure that only the central region remained, and resampling the images to a fixed resolution of 2×2 mm and a fixed image size of 128×128. This study furtherly demonstrated that including data from different sources could make a network for segmentation more generalizable to other acquisition sources.

In another work [120] the researchers proposed a data normalization pipeline that incorporated data augmentation to train a 2D U-net with a large dataset coming from a single vendor and center and then tested it with another dataset that came from different vendors and centers. The pipeline included resampling the image *xy* plane to a specific resolution of 1.25 × 1.25 mm without modifying the slice thickness. Intensity normalization was applied to that all the images so they had a mean intensity of 0 and a standard deviation of 1 (Z-score normalization). The data augmentation applied incorporated rotations, flipping operations and zooming effects to artificially increase the heart size. Last, all images were cropped to set the size to a constant range of 256 × 256 pixels. The cropping operation was randomly applied during training but for the test set they applied the crop only in the central region where the heart is usually present within the images. Employing this pipeline to standardize the training resulted in a network that reached average DC around 0.9 for the LV cavity, 0.82 for the LV myocardium and 0.82 for the RV cavity. This showed that standardization on a single center and source machine dataset could result in an efficient neural network for segmentation in datasets coming from different sources, although the quality still falls short when compared to the results obtained when the network is only applied to the same source machine it was trained on or when multicenter and multivendor training sets are employed.

It is clear that the majority of segmentation works focus their target in the LV and sometimes in the RV and LV myocardium, however there are other regions that might be of interest, mainly the RV myocardium and the papillary muscles. Even with the lack of works targeting these regions there exist some research addressing the importance of the papillary muscles in certain pathologies [121, 122]. Convolutional neural networks have been applied to segment these muscles as well, but to our knowledge only in one work [123]. In this work they achieved a mean DC of 0.72, 0,79 and 0.82 for different cardiac pathologies. Overall there is a considerable lack of work on this specific problem.

5.3. Material and Methods

5.3.1. Data

For the experiments the ED frames of the dataset were employed for training the models and both the ED and ES frames were used for testing. From the 399 available images, 99 (25%) were used for testing and the remaining 300 were used for training.

The 99 cases of the test set were both from the same acquisitions for the ED and ES evaluation, however only 98 cases were used for the ES due to missing labels in one case. Additionally, for training the 300 remaining cases were split in training (260 cases, 65%) and validation sets (40 cases, 10%). The splitting also considered the disease distribution, so every dataset had roughly the same proportion of categories. All the splitting was done randomly.

All the images employed were preprocessed before the experiments. The images were first resampled using bi-linear interpolation to an in-plane resolution to 1×1 mm and the image size was set to a constant of 176×176 pixels. For the image resizing cropping and zero-padding the borders was applied when necessary in order to get the desired size. The third axis was left untouched in both size and resolution. These preprocessing did not affect the presence of the heart within the images, as it was always present in the central region of the image plane. The same procedure was applied to the labeled images, with the exception of the interpolation technique, which was substituted with nearest-neighbor. Intensity normalization was also applied, in this case as all acquisitions came from the same scanner a min-max normalization scheme was applied to the entire volumetric image.

Since the original segmentations only included the tissue’s borders, they were modified to represent volumetric segmentations. Specifically, for the ED the integer values 1, 2 and 3 were assigned to the LV inner chamber, LV myocardium and RV inner chamber respectively. For the ES cases used, labels of 1 and 2 were assigned to the LV and RV inner chamber.

5.3.2. Models Architectures

Two different convolutional neural networks were designed and implemented. The two were based on the U-net architecture, one being a classical 3D type and the other having a new and novel design.

The first one was a 3D U-net that incorporated 3×3×3 convolutions with ReLU activation functions and batch normalization (BN) in each layer. The number of downsampling and upsampling steps was 4 using maxpooling for downsampling and transposed convolutions for the upsampling. Due to the lower resolution in the third axis, the downsampling was only applied in the image plane, leaving the third axis size constant through the different layers. This 3D U-net used as input image patches of size 176×176×3, meaning it could process three full slices each time. The final layer used a softmax activation function to produce a total of 4 channels, 1 for the background and the remaining three for the different regions. The model had a total of 87.51 million parameters, which occupied 1.03 GB of memory space. The architecture is visually presented in detail with the specific number of feature channels per layer in Figure 5.3.

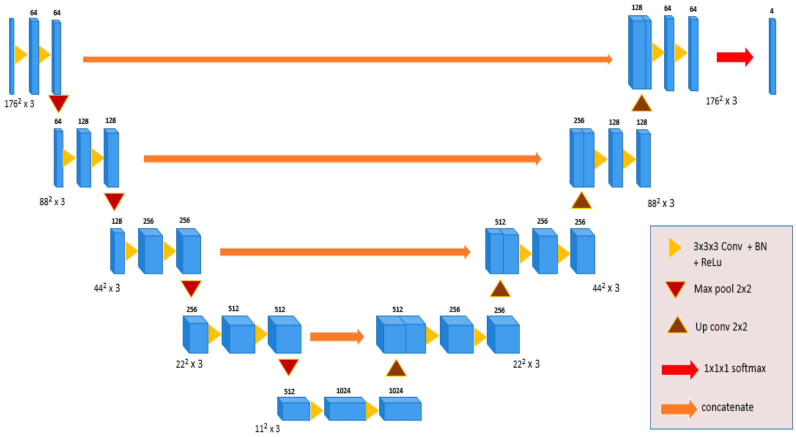


Figure 5.3 3D U-net architecture employed for the experiments. The architecture is equivalent to a vanilla U-net with minor modifications to adapt to the input's nature, like not applying the pooling operations in the 3rd dimension. The final layer uses the softmax activation to produce 4 different channels, each corresponding to the three different labels plus the background.

The second architecture is a combination of a 2D U-net with pyramid scene parsing modules (PSP) as in the PSPnet [125], which we called PSPU-net. These module's design allows the model to analyze inputs at different scales in parallel to better incorporate more contextual information. The PSPU-net was designed to work with patches of 3D slices, but all convolution layers were of size 3×3×1. In this way the design allows to process the same quantity of information as the 3D U-net but only processes it in 2D, effectively making this a 2D model. Every convolution layer is likewise followed by ReLU activation functions and batch normalization. The downsampling and upsampling layers are the same as in the 3D U-net, only applying the operations in the slice plane. The PSP modules are incorporated in the skip connections allowing to process each feature map stack at different scales in parallel on top of just downsampling them. The PSP module design is as depicted in Figure 5.4. The figure represents the modules at the highest level, lower levels have the same overall design but at every lower level the number of paths is reduced, eliminating the highest sampling rate path from the previous PSP module. Additionally, the number of filters is duplicated with respect to the previous PSP module. As an example, Fig 5.4 represent the 4-path PSP module (highest level), the 3-path PSP module of the next level will suppress the ×16 downsampling path and the remaining paths will have their number of channels duplicated. The same pattern is followed in successive modules.

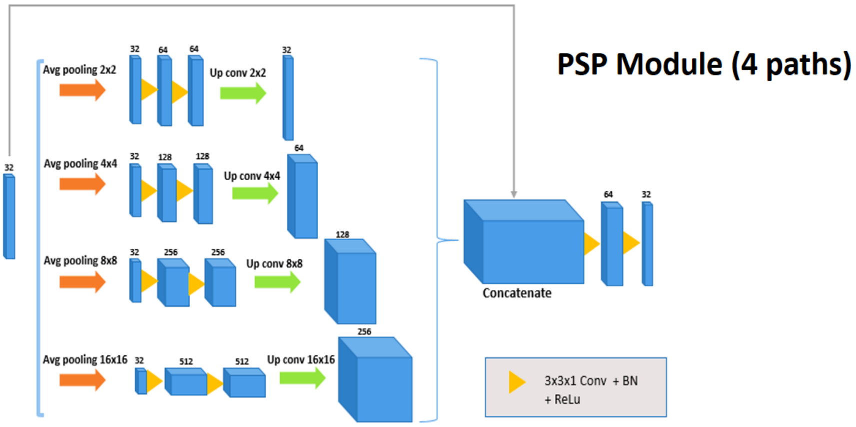


Figure 5.4 Overview of the PSP module with 4 paths employed in the PSPU-net used in the experiments. The inputs are processed in parallel at different sizes and the concatenated together to apply the last convolution layers. Following PSP modules (3, 2 and 1 paths) are equivalent but with the elimination of the higher downsampling path from the previous PSP block and doubling the number of channels of the retained paths.

An additional difference from the 3D U-net was that the number of feature maps throughout all the network is halved compared to the latter. The output is the same as in the 3D U-net. These implementation of the proposed PSPU-net had a total of 30.83 million parameters and occupied 362 MB. The whole PSPU-net architecture employed is presented in detail in Figure 5.5.

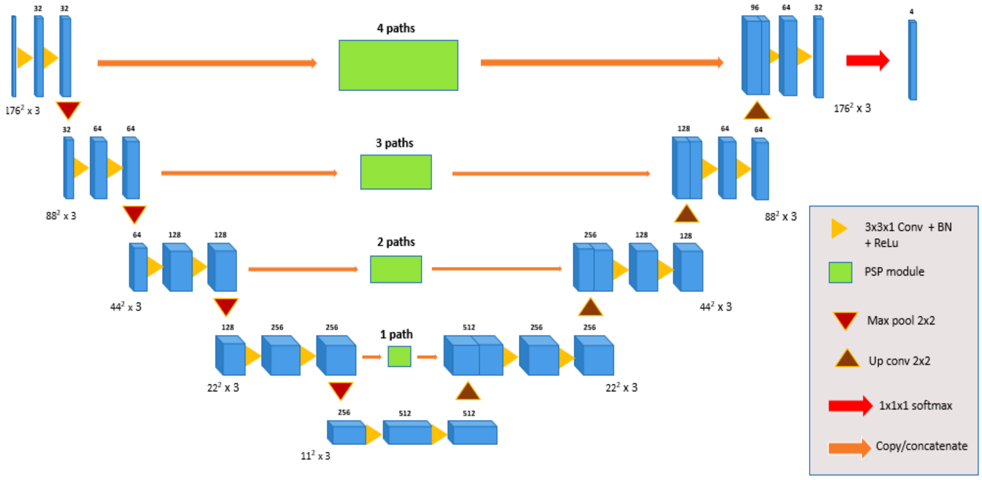


Figure 5.5 Overview of the PSPU-net architecture employed for the experiments. The 2D convolutions are implemented as 3D convolutions and the skip connections are characterized by applying the PSP modules before passing the feature maps to the upsampling layer. The final layer uses the softmax activation to produce 4 different channels, each corresponding to the three different labels plus the background.

5.3.3. Training schedule

Both, the 3D U-net and PSPU-net described were trained with the same configurations. They were trained for 50 epochs employing the training and validations datasets. Some testing was done in order to find the best values for the main hyperparameters to tune. More specifically the best configurations for both included the use of a learning rate of 0.001 with a batch size of 3. The optimizer employed was ADAM [126]. We note that these settings are similar to those used in other works for similar segmentation tasks [84, 116, 127].

The loss function used was the generalized Dice loss (GDL) [128]. This loss uses the generalized Dice score (GDS) [129] which assigns a weight based on the relative space occupied by each region. However, for this task we assigned specific values to the weights: 0.1 for the background and 0.3 for the different tissues. With this, it was intended for the networks to give the same importance to each region during the training process with the sole exception of the background (whose weight was set to a reasonable lower value).

Finally, the size of the training dataset was increased with data augmentation techniques to help avoid overfitting. The original size of the training dataset was of 2045 volumes of 3 slices extracted from the 260 full volume, while the validation consisted of 319 inputs extracted from the 40 full volumes. The extraction was performed by obtaining windows of three slices with overlap (step size of 1). The training dataset number was increased to a total of 8340 inputs. The additional 6295 inputs were generated by randomly selecting the inputs and applying a series of random transformations. In the random selection of the input to transform, the same input was only allowed to be selected up to 4 times. The transformation applied consisted of a random rotation between -30 and 30 degrees, a random zoom factor between 0 and 0.1 in the image center and a random shear between -20 and 20 degrees. These transformations were applied to all the 3 slices of each input in the slice plane, while the third axis was left untouched. The final images were automatically cropped to retain the original size if the transformation increased it (which could happen with the rotations).

5.3.4. Segmentation Evaluation

Two different types of measurement were done in order to evaluate the quality of the segmentations in the test set. This was done in both the ES and ED frames.

The first type of measurement was the level of overlap between the predicted segmentations and the manual segmentations. For this the Dice coefficient was employed for each different region. Additionally, the global Dice coefficient was also measured in order to get an average metric of the whole segmentation as well.

The second type of evaluation was the direct comparison of volume values derived from the automatic and manual segmentations. More specifically the relative absolute error was used. These measurements might be more precise in order to check the real performance that the models would have in the real world, as these are the final metrics that are used for the clinical assessment.

5.4. Results

5.4.1. Training performance

There were some differences in the training of both neural networks. The 3D U-net took 27 hours to complete the training, while the PSPU-net took 20 hours. Both the training and validation losses were tracked through each epoch, as depicted in Figure 5.6. It is noticeable that both neural networks seem to reach similar loss values throughout all epochs. Additionally, in both cases, the validation loss reaches a limit at a very early epoch (epoch 4 out of 50) and stays approximately still during the remaining ones. This indicates that even though the training loss keeps decreasing there does not seem to be any noticeable overfitting and the optimal generalization status of the models was reached very fast.

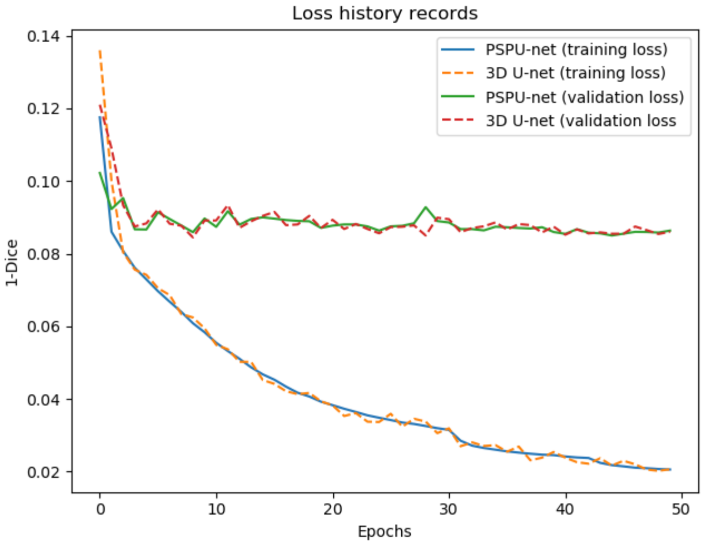


Figure 5.6 Graphic curves of the training and validation losses of the architectures over the entire training process (50 epochs). Overall both models had similar loss curves but the PSPU-net seemed to have less fluctuation in both the training and validation losses.

There is a noticeable pattern in the loss values. Even if both networks follow the same trend, it can be seen that the 3D U-net has bigger fluctuations than the PSPU-net, this is seen in both the training and validation losses, as the PSPU-net line tends to be in the middle of the 3D U-net fluctuations. This fluctuation difference is not big in absolute terms, but it may indicate that the training of the PSPU-net was slightly more stable.

5.4.2. Segmentation quality

Both neural networks were tested against the test set of ED (99 cases) and ES (98 cases). The ES testing was applied in order to check how well the neural networks could segment the same objects at different contraction steps. Basically, this also allows to check how well the models can work when working with images of similar nature but with notably different characteristics. The segmentation was evaluated on the entire image, meaning that the segmentation had to be reconstructed from the different segmented patches before calculating the quality (Figure 5.7 represent this process). The segmentation quality was measured using the Dice coefficient, table 5.1 show the results of the DC for each region and the average.

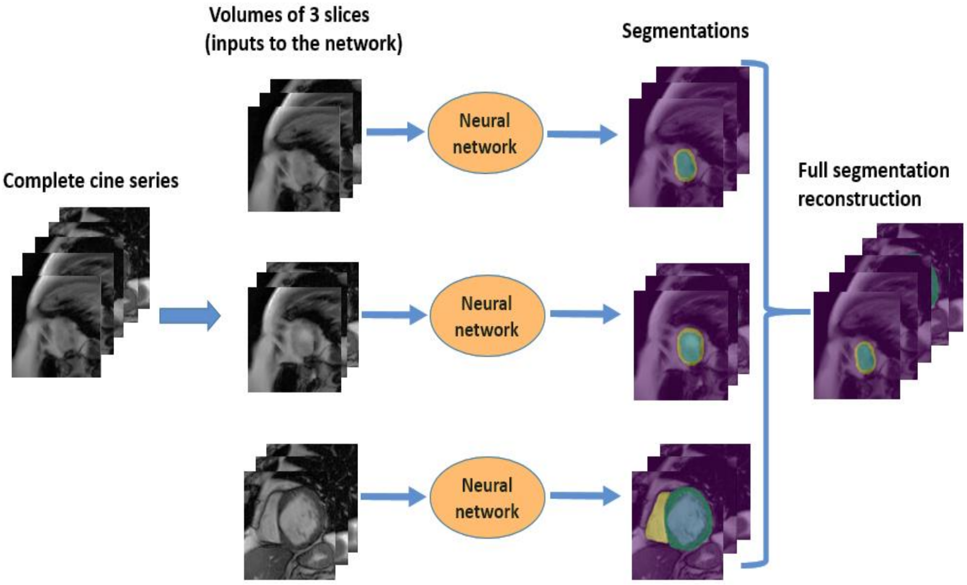


Figure 5.7 Reconstruction process from segmented patches. For each new image, the 3-slice stacks are extracted and passed independently over the neural networks to produce the segmentation on the stacks. The final segmented image is recovered by concatenating the stacks.

Table 5.1 Dice coefficient measurements on the test set for both ES and ED frames in the two trained models. Median and standard deviation of the DC distributions.

Model and evaluated frame	**Whole segmentation**	**LV cavity**	**RV cavity**	**LV myocardium**
3D U-net (ED)	*0.907±0.028*	*0.956±0.021*	*0.904±0.042*	*0.875±0.039*
PSPU-net (ED)	*0.910±0.026*	*0.955±0.021*	*0.905±0.036*	*0.875±0.037*
3D U-net (ES)	*0.826±0.060*	*0.881±0.073*	*0.781±0.080*	-
PSPU-net (ES)	*0.848±0.053*	*0.896±0.053*	*0.798±0.070*	-

At ED both networks showed similar high quality segmentations. Overall it seems that the PSPU-net is slightly superior, with slightly higher median DC and lower standard deviations. Still both achieved very satisfactory results with median DC of 0.95 for the LV cavity, 0.90 for the RV cavity and 0.87 for the myocardium. Figure 5.8 shows an example with some segmented slices for the ED.

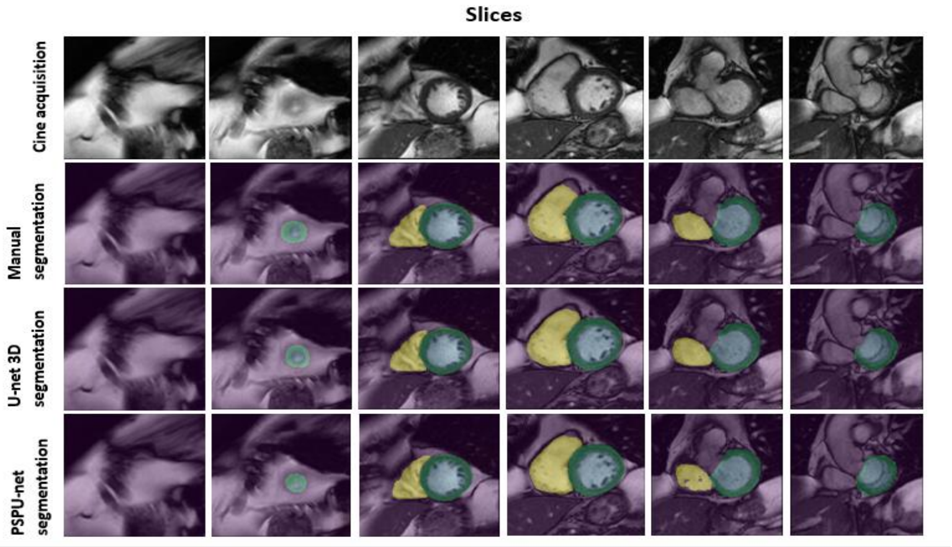


Figure 5.8 Example of obtained segmentation results. Visually, the results for both the 3D U-net and the PSPU-net show very similar results to that of the manual segmentations, with slight and mostly unappreciable differences.

For the ES test cases the results were more different between the two models. In this case the superiority of the PSPU-net is more clear in both the median and standard deviation of the DC. In this cases the LV chamber still has a considerably high DC value (0.896 and 0.881 for PSPU-net and 3D U-net respectively), but the decrease in the RV cavity is more notable (although close, neither reached a median DCS of 0.8).

The times required for automatically segment each case were on average 0.91 seconds and 1.11 seconds for the PSPU-net and the 3D U-net respectively employing the available hardware. These times include the segmentation of the different patches and the posterior reconstruction. These are the average times for segmenting one case each time (for each case the batch size at inference was equivalent to the number of patches that encompassed the whole volume).

5.4.3. Volumetric estimation quality

Besides the segmentation quality it is also important to check the derived volume values in order to get a better estimation on the quality of results. The volume values are the ones that are used in the end to compute the important biomarkers, so the final target is to get a good approximation to the real ones. Table 5.2 show the relative absolute error derived from the manual and the automatic segmentations for both the ED and ES volumes.

Table 5.2 Relative absolute error distributions obtained by the models in the test sets. The values indicate median and standard deviation

Model (evaluated frame)	LV cavity	RV cavity	LV myocardium
3D U-net (ED)	*0.025±0.032*	*0.058±0.070*	*0.048±0.049*
PSPU-net (ED)	*0.026±0.033*	*0.051±0.047*	*0.039±0.051*
3D U-net (ES)	*0.115±0.121*	*0.258±0.221*	-
PSPU-net (ES)	*0.084±0.118*	*0.234±0.196*	-

For the ED the tendency is similar as in the segmentation quality, with slightly better results for the PSPU-net, however the differences in this case are more notable. Still in both cases the median relative error is very low for all regions, with the highest being 0.051 (PSPU-net) and 0.058 (3D U-net) for the RV cavity, followed by the myocardium with 0.039 (PSPU-net) and 0.048 (3D U-net) and with the lower errors being those of the LV cavity with values of 0.026 (PSPU-net) and 0.025 (3D U-net).

In the case of the ES important error values were obtained. In the case of the 3D U-net the median absolute relative error resulted in 0.258 and 0.115 for the RV and LV cavities respectively. The PSPU-net obtained values of 0.234 and 0.084 for the same respective regions. As with the ED but to a greater degree, the PSPU-net resulted with notably better results in the derived volume values.

5.5. Discussion

The task of segmentation in short-axis CMRI is one of the most extended ones for this type of image, mainly because it is the most time-consuming task if performed manually or semi-automatically. There have been many works that achieved good quality results. By analyzing some of the major works and the results obtained with the implemented models some insights can be achieved for this problem.

We will first address the fact that the results obtained demonstrate that the PSPU-net has considerably better performance than a vanilla 3D U-net for short-axis cine CMRI segmentation. For images where both models were trained on ED both achieved very satisfactory and high-quality results, with slightly better results for the PSPU-net. However, when tested against images of similar nature to the training set but with different key features in the targeted regions (contracting state against relaxed state) the PSPU-net obtained considerably better results than the 3D U-net. This is an indicative that the PSP modules are capable of extracting additional abstract information that makes the model more robust to altered states of the targeted regions. Additionally, the PSPU-net's field of view is limited to 2D, compared to the 3D U-net that can make use of an additional axis with more information available. This further proves the previous statement, however this conclusion should be taken with caution, being this true for short-axis cine CMRI there could be other types of images where the 2D plane does not have enough information for the PSPU-net to outperform a 3D U-net. It must be considered that the type of images we are employing usually contain enough information in each slice to correctly segment it, so a different result could be obtained under different image types. Still, it seems clear that the PSP blocks improve the generalization capability of the model, so in a situation like the hypothesized one could probably be solved when extending the use of PSP modules to 3D. Last, it should be considered that in the case of the PSPU-net the errors obtained for the LV cavity in ES were low enough to be of practical use, although the quality would certainly improve if the model had been trained to specifically target the ES frames.

Besides the quality comparison, another important difference is the size of the models, the implemented 3D U-net is almost three times bigger than the PSPU-net used, which also results in faster training and inference speed and less memory consumption, making the PSPU-net described a more efficient one than the 3D U-net.

Comparing the results obtained to other works there are some important findings. A key one is the difficulty in segmenting the different regions. In all previous analyzed works [84, 116-120, 123] the best segmentation quality was consistently achieved for the LV inner cavity, followed by the RV inner cavity and the myocardium, additionally the results tend to be noticeably better in ED frames compared to ES. The only exception to this is that for the ES the myocardium tends to have better quality results than the RV cavity and at the same time the myocardium also tends to be better segmented at ES. We could not test this with our dataset as myocardium segmentations in ES were not available, but this tendency is found both in the works described and in all the competition results found in the leaderboard of the ACDC challenge (*https://acdc.creatis.insa-lyon.fr/description/results.html*). These findings suggest that in order to obtain better segmentations more focus should be given to harder regions. This could be achieved by giving bigger weights to harder regions in the loss functions employed. This, however, should be tested meticulously, as doing so could easily worsen the result of the better segmented regions as well. Training different models to only target one region each could also result in better segmentations, since the model will give all its attention to a single target. But this would inevitably come at the expense of requiring different models for different regions at inference time.

Another expected finding is that CNN that have been trained to segment a specific frame will not perform as well on other frames, we have tested this by comparing the results on ES to the ones of the ED where the models were trained on. This was in principle expected, as neural networks are known to obtain worse results in images that differ from the ones they have been trained on, however it is important to mention it, as it is intrinsic to the task. The typical approaches to this are to either use both ES and ED frames on the same model or use different models on different frames. Additionally, one could also apply data augmentation methods to deform one type of frame in order to make it look like another. There are some inconveniences to these methods. On one hand, unless we have a very large dataset, mixing the two types of image will probably result in worse results than only training against one type of image. On the other hand, requiring different models per frame is by itself a handicap. Consider that in the clinical setting one would also need to determine which are the ES and ED frames. Solving this via segmentation would require to train several models that can predict the segmentation at different time points, and then compare the volumes in order to select the correct ones. Although this is an approach that has been described [130] it is very impractical due to large number of labeled data required in any setting.

Finally, we would like to discuss the fact that not all works focus on segmenting the same regions, but some insights can also be extracted. Overall, the LV inner cavity is the most targeted region, which is expected, as the LV ejection fraction is the main diagnostic parameter used in to characterize the heart's contractility state. The RV and the myocardium seem to follow in the same degree of importance, however, we note that although the myocardium is also targeted at ES in some works, in our case it was only measured at ED, suggesting that this measurement is only required at more specific scenarios. Indeed, having the myocardium segmented at ED is sufficient to extract the ventricle mass and only the ES would be needed if wall thickening wants to be measured. We will additionally mention the papillary muscles, as there are works that have address this segmentation problem [123], but it is clear that this task is far less extended in the research community, probably due to the lower number of performed clinical analysis that may require this compared to segmenting other regions.

5.6. Conclusions

This chapter covered the introduction of a new type of CNN that combines the U-net and PSPnet architectures resulting in the PSPU-net. The results obtained in this study indicate that the use of PSP modules can result in better segmentations when using 2D models even against 3D models. This is true for short-axis CMRI, where the implemented PSPU-net have been demonstrated to obtain high quality results and outperform a classical 3D U-net for the ED frame, with which they were trained. On top of that, the training was slightly more stable for the PSPU-net.

Additionally, the PSPU-net further outperformed the 3D U-net against ES frames that the models had not been trained with. This determines that the incorporated PSP modules helped the model learn more global and generalizable features from the images that made it more robust against image outliers cases, such as very different heart contraction states.

Overall, it can be concluded that combining PSP modules with U-nets can result in more efficient models that can extract more robust features from the images in order to obtain high-quality segmentations. This has been proven on short-axis cine CMRI images, with results good enough to be employed in the clinical setting, offering minimal errors in the volumetric estimations.

Chapter 6. Automatic biomarker estimation and explainability

This chapter is based on a journal paper published in the context of this thesis [163]. The document is available at: *https://doi.org/10.1016/j.cmpb.2021.106275*.

6.1. Introduction and Motivation

Besides segmentation, classification and object detection, CNN can also be employed for regression problems. In our context, this is the automatic estimation of biomarker values from the images. This approach has been applied in the medical image field as well. The usual procedure to make use of CAD systems, as described in chapter 1, is to use an automatic system first to detect/segment the region from which the biomarkers will then be derived. In the case of short-axis cine CMRI these are mostly volume values and metrics derived from them. However, training a model to directly target the automatic computation of these biomarkers without segmenting a region can suppress some steps in the usual pipeline employed in most CAD systems. This allows for the CAD to directly offer the biomarkers without needing to obtain a segmentation nor applying post-processing for the biomarker estimation.

The regression task in the clinical context may have and additional benefit. Although manual segmentation is required to train a model for segmentation, in the case of regression one would only need the previously measured biomarkers. Medical images with segmentation dataset are not common within healthcare systems, however, there are plenty of recorded measured values available [131]. Furthermore, even if some kind of ROI was manually measured in order to obtain the desired value, it is not that usual to save these ROIs as usable information routinely, making this type of data scarcer. This is also true in CMRI in general, and makes directly targeting the biomarker values through regression an attractive approach to employ CNN for medical image analysis.

This, however has a very important negative point. A CAD system that only produces the final desired measurements is difficult to trust in a context as delicate as medical diagnosis. Even when a specialist only sees the biomarkers values measured by a radiologist, they are trustful because they were generated by another clinical expert that knows the actual procedure to make measurements. In the case AI in general (and more particularly in the deep learning field) the models are often viewed as “black boxes” that are complex enough to find patterns within the data in order to produce their estimations, even if we cannot understand the inner process by which a specific model predicted some result. Black boxes are certainly undesirable in the medical context, as important diagnostic and treatment decisions need justification. In order to address this situation there is a whole branch of research that focuses on trying to understand the mechanism by which these models reach their predictions. This is called “Explainable AI” (XAI), or alternatively “Interpretable AI” and it has produced some notable results, but it’s still a relatively new field and sill a lot of ongoing research is being done in the field.

More specifically XAI involves two different sub-problems: explainability and interpretability. These two concepts are often used interchangeably but are not exactly the same. Interpretability can be defined as the degree to which humans can understand the cause of a result or a decision, which directly involves understanding how all the model’s parameters affect the inputs [132]. Explainability, on the other hand, refers to the capacity to understand how a model operates in order to obtain its predictions (its overall inner mechanism of operation with the input features). These two concepts can overlap some times, but to put it with a simple example we can view that a decision tree or linear regression are both highly interpretable and explainable models, as we can easily describe how the algorithm operates and also know exactly how an input is affected and predict beforehand looking at the model the exact result an input will produce. Neural networks, with their great complexity have very low interpretability (the smaller CNN usually having in the order of millions of parameters), although explainability can still be achieved with the correct design and techniques.

Segmentation neural networks, although still having the same issues, generate a segmentation of the region used to compute the biomarkers, so for the experts, understanding that the final biomarker was obtained from a region they can check is sufficient to be sure that the results are reliable. In this sense, although a segmentation model might not be interpretable nor explainable, the final result by itself can explain from where the final computed biomarkers come from, which are the ones actually used for the diagnosis. With this in mind, this chapter covers the design and implementation of an explainable CNN that is trained only with LV volumes at ED to directly estimate them from the ED frames, but simultaneously, at inference time, its design offers explainability by producing an indirect segmentation of the region it based its calculations from. To produce the segmentation from a regression model, weak-supervised learning techniques were studied and employed. Weak-supervision is another important field within AI that tries to predict additional information from much more simple labels. The proposed approach mixes the concepts of weak-supervision and explainability, offering a way to train a CNN using only the biomarker value to obtain the segmentation of the target region. With this, two different objectives can be established regarding the LV in short-axis cine CMRI: producing high-quality explainable regression models, and training segmentation models without the need of manual segmentation labels, requiring only the final measurements for training.

6.2. Related work and state of the art

6.2.1. Regression convolutional neural networks

Convolutional neural networks employed for regression are not that common compared to the more widespread problem of classification. However, there have been some work on it, typically to make a model predict some spatial feature of the image, like rotations or geometric locations [133, 134]. One important consideration for these tasks is that usually the models have a lot more difficulties at converging at a good solution, so the scale of the dataset required can be considerably large for these problems. This is easily explained by the fact that a regression task within the image context is a very complex problem. In this case, the model needs first to analyze the image and find within it which features characterize the target value (if they are present within the image at all), then it will need to learn to map the located image features to the correct value. This is far more complex than any type of classification or localization task, where the second step is not applied.

The typical architecture for regression task is basically the same as for classification except for the final activation function: first several convolutional and pooling layer extract the spatial features from the images and then the resulting features are converted into a vector and passed to a fully connected neural network that at its final neuron uses an activation function that can map the result to the range of values of the regression task, usually a linear function, although others like ReLU (if only positive values are allowed) or sigmoid (if the range is between 0 and 1) could also be employed depending on the context. Additionally, these models are trained with different loss functions. Typical loss functions for regression tasks are the root mean squared error (RMSE), mean absolute error (MAE) or relative mean absolute error (RMAE)

In the medical image field, the regression task refers to the direct estimation of specific biomarkers from the images [131]. This is an approach that has been taken for different problems. To name some example we can mention age prediction from T1 MRI brain scans [135], estimation of bone mineral density and of lung percentage of emphysema [136], morphometric parameters of the corneal endothelium (cell density, cell size variation, and hexagonality) in corneal endothelium microscopy images [137], or Agatston score obtained from chest CT scans of the heart [138].

For the specific case of short-axis cine CMRI LV volume estimation via regression with neural networks there are very few works compared to the segmentation task. In [139] the authors presented a regression CNN for LV volume estimation that was employed in both ED and ES frames in a large image dataset of 1140 subjects (Data Science Bowl Cardiac Challenge Data). The network consisted of 5 convolutional layers followed with 3 fully connected layers. Another approach proposed in [140] using the same dataset added important pre-processing steps to the images in order to crop a ROI containing only the LV and then fed this data to a regression CNN with 13 convolutional layers followed by 3 fully connected layers.

6.2.2. Explainability in convolutional neural networks

Explainability has been one of the main focus in the deep learning research field. This has been even more important in image-related tasks. Many techniques have been developed to allow for explainable models, but the vast majority aims to produce some type of saliency map for the input image. A saliency map in the computer vision domain refers to an image heatmap that highlights the pixels of the image that the human vision first pays attention to. In the context of CNN these heat maps would give information regarding the spatial regions within an input image that the model is giving more importance to obtain its final outputs.

There are multiple techniques that allow the obtention of heatmaps intended for visual explainability. Some of the most famous ones are CAM (class activation mapping)

[141], Grad-CAM (gradient-weighted class activation mapping) [142], Guided Grad-CAM [142] or smoothGrad [143]. These are techniques that are applied at inference, after the model has been trained (sometimes they are also called post-hoc attention mechanisms). On the other hand, trainable attention mechanisms (or simply attention mechanisms) [111, 113] may provide heat maps as an intrinsic feature of the model. Besides these, there are other methods like Lime (local interpretable model-agnostic explanations) [144] or SHAP (SHapley Additive exPlanations) [145] which provide information on the input's features relative importance and can be used for different model types, including those related to imaging, although they are not as popular as CAM-based or attention-based approaches for visual tasks. Figure 6.1 shows an example of a heat map produced by Grad-CAM on a medical image task.

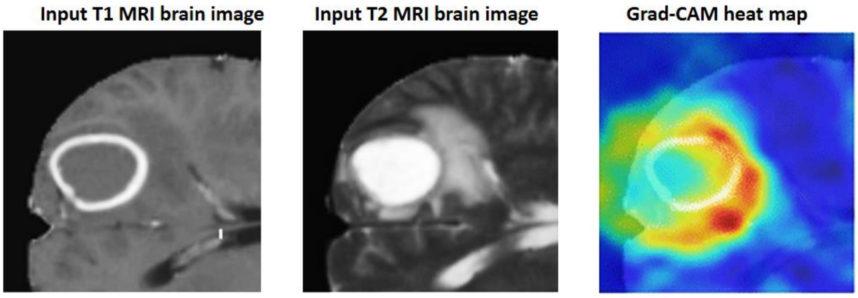


Figure 6.1 Example of visual explainability. Explainability methods often make use of heat-maps generated from the model's inner activations. In this example the Grad-CAM method is applied to a model for tumor classification on T1 and T2 brain MRI images to produce a heatmap that gives the region whose features were more important for the final output. Image modified [146].

As visual explainability can offer a way to help understand the model's focus on the image, these techniques have been applied in medical imaging as well [147]. Some topics where they have been applied are brain MRI imaging for Alzheimer disease [148] and brain tumor [146], breast MRI imaging for estrogen receptor classifying estrogen receptor status [149], or chest X-ray imaging for COVID-19 detection [150]. The overall application of these techniques was to help at finding the patterns within the images that the trained models employed to produce their predictions.

6.2.3. Weak-supervised learning

Weak-supervised learning refers to a branch of machine learning techniques that focus on using limited label information to train models that can produce complete labels [151, 152]. This allows to train a model to predict some type of label that is difficult to obtain employing weak but inexpensive label types that may include the target information in an indirect way. In computer vision this can arise in several ways. One typical example of the application of weak-supervision would be locating and/or segmenting objects within an image with only the information about the object's presence within it available for training (for example with some classification label).

Healthcare data-related tasks and medical imaging in particular are one of the principal fields where weak-supervision can prove to be especially useful, mainly due to the need of medical experts to label the data, which can be very expensive and a hard task to accomplish. As such, there have been some medical imaging applications that used these methodologies [153]. Some examples of applications where good results have been reported are: prediction of values of pectoralis muscle area (PMA), subcutaneous fat area (SFA) and liver mass area in single slice computed tomography (CT), and Agatston score estimated from non-contrast thoracic CT images (CAC) without training for the specific target [131]; covid-19 infected region segmentation using single points [154] or segmentation of different organs like spleen or pancreas using extreme points from the organ's contour [155].

For medical imaging the most important application of weak-supervised learning is to generate segmentations of a certain region of interest when only other indirect information is available. This type of task has been tested on different types of images using different characteristics to train with, like whole image classification labels [156, 157], seed points of the region to segment [154, 156], regions of interest as bounding boxes [158-160], or points around the contour of the region to segment [155, 161].

6.3. Material and Methods

6.3.1. Data

For the experiments described in this chapter, 397 cases of the available dataset were employed. The experiments were limited to the ED frame, so only the volumetric images corresponding to this time point were used. The volumetric values of the LV at

ED (in ml units) derived from the manual segmentations were employed for training and testing phases. The LV segmentation was only used for testing.

The dataset was randomly split in training, validation and test sets. All groups included the same percentage of cases for each different pathological description (as described in chapter 4, section 4.1). The training set included 259 cases (65%), the validation set 40 cases (10%) and the test set 98 cases. (25%).

All the images employed were preprocessed before the experiments. The images were first resampled using bi-linear interpolation to an in-plane resolution of 2×2 mm and the image size was set to 88×88 pixels. For the resizing, cropping and zero-padding was applied when necessary. The third axis was left untouched in both resolution and size. This downsampling was applied to reduce the number of features (represented by each voxel) for the network to process. These preprocessing did not affect the presence of the heart within the images, as it was always present in the central region of the image plane (see Figure 6.2). The same procedure was applied to the segmentation images, with the exception of the interpolation technique, which was substituted with nearest-neighbor. The z-axis was not modified in any manner. Finally, the images were normalized to make the pixel values range from 0 to 1 using min-max normalization.

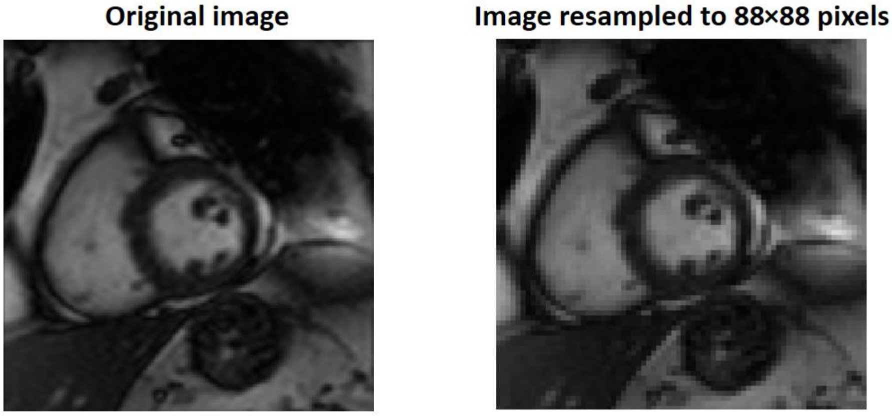


Figure 6.2 Visual differences between the original image and the downsampled one used for the experiments. Many high-frequency details are lost during the downsampling process but the main regions are still clearly differentiated.

6.3.2. Model architecture

The approach taken in the architecture design is dramatically different from typical regression CNN, in this case the overall scheme is that of a 3D U-net [109] with notable modifications followed by a scanning module that uses the 3D U-net output to extract the final volume estimation.

The 3D U-net design is a general 3D U-net that uses as inputs images of size 88×88×n. Here, n represent the number of slices of the image and can be variable, so the network can process images with different number of slices. This was done in order to avoid resampling the slice axis that had lower resolution and for which our dataset had sizes ranging between 8 to 14. All levels include two 3D convolutional layer of size 3×3×3 followed with a ReLU activation function. Additionally, all layers included batch normalization. The features maps sizes are reduced within the network with maxpooling operations of size 2×2×1 in order to halve the image plane while keeping the number of slices constant.

The up-sampling path is composed of up-convolutions of size 2×2×1 to which the resulting feature maps are concatenated to the downsampling path feature maps through the skip connections as in any typical U-net. Strong regularization is applied to the “bottleneck” layer output (that is the layer just before starting the up-sampling). This is done by means of L1 activity regularization in order to force the network to learn only a small subset of important features at the more abstract level, this will allow the network to highlight important spatial regions while setting to 0 those less significant. The last layer consists of a 1×1×1 convolutional layer with a sigmoid activation function. L1 activity regularization is also applied to the output feature map to force the network to generate a segmentation as close as possible to the LV region (by means of setting to 0 non-relevant regions and to 1 the relevant ones). This last output is the class activation map (CAM) that gives the probability that a certain voxel is part of the target region [162], which in this case is the LV. This CAM output, although related in its conceptual idea and terminology, is not to be confused with the CAM technique (class activation mapping) referenced in section 6.2.2. See Figure 6.3 for a representative example of a CAM output of the trained model.

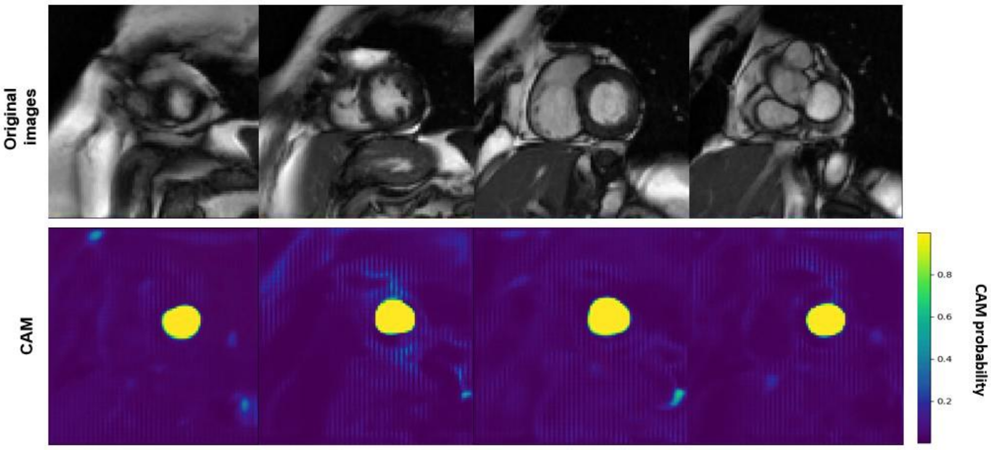


Figure 6.3 Example of the CAM produced by the trained model on different slices. It is clearly visible how the left ventricle region matches that of the high activity of the CAM. Additionally, the model's design allows for the generation of CAMs whose left ventricle region have near 1 probability values, while maintaining the background to near 0 values.

The output CAM of this 3D U-net should represent the LV region. In order to achieve this an additional module is coupled at the end. This module scans the CAM in order to extract two key characteristics: a diameter feature of the object and the volume within it. Both of this will be used as targets to define the loss functions. The volume captures the actual volume to estimate, and the diameter will be used to compute an estimation of the π value. The reason for estimating π is to force the learning process to base its volume prediction from a circular object, as the LV has an overall round shape.

In order to compute the volume, the scanning module first applies a non-trainable convolution with a single kernel filled with ones of size 25×25×1. This size is big enough to ensure the LV is always captured within it. After this an additional 88×88×1 maxpooling operation is applied. This pooling will capture the biggest size of any object that could fit within the previous convolutional layer kernel for each slice and subsequently produce a single value per slice that will correspond to the area occupied at each slice (in number of pixels). These areas are then summed together and the result is multiplied by the image resolution per voxel converted to ml, obtaining an estimation of the volume.

The diameter is computed using two different paths. The first applies an 88×1×1 non-trainable convolution with a fixed kernel fixed of ones. The other applies the same operation but with a kernel size of 1×88×1. These convolutions will estimate the diameter of the object within the image along vertical and horizontal directions. The results of both paths are then averaged at the slice level. With this, the model will be forced to compute the largest diameter of the objects present within the CAM. The estimated diameters are then used along the estimated areas to get a ratio that is used as an additional estimator of the π value (as round objects will get a close value to it). This section basically encompasses a circularity feature extractor, which is what ultimately allows the net to detect circular objects whose volume match that of the target, corresponding to the LV. Figure 6.4 and 6.5 shows the detailed architecture and scanning module designs. The whole model contained a total of 21.87 million parameters.

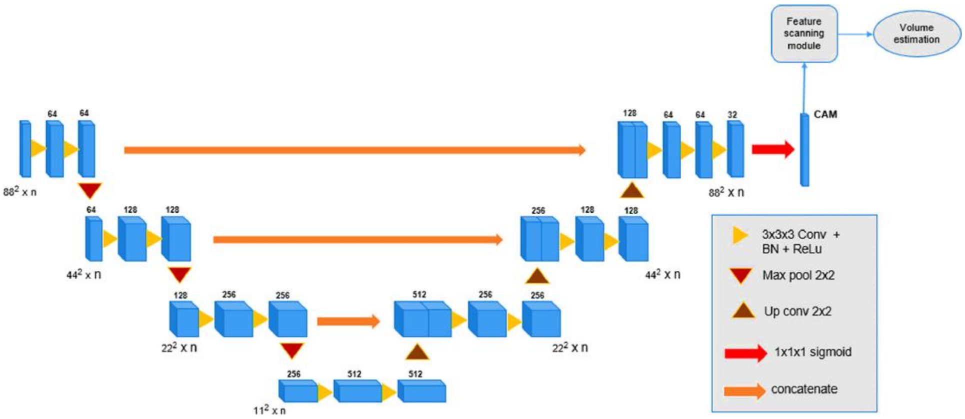


Figure 6.4 Architecture of the model employed for the experiments. The general design is that of a 3D U-net that can process volumes of 88×88×n sizes, where n can be any integer number. At the end of the model a CAM is produced with a sigmoid activation function, which is then passed on the feature scanning module.

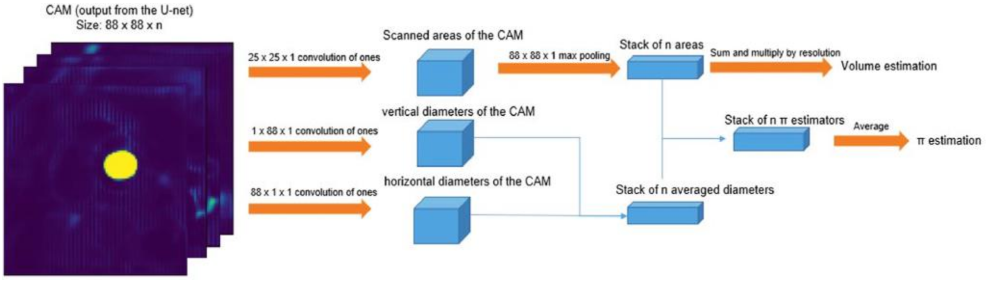

CAM (output from the U-net)
Size: 88 x 88 x n
25 x 25 x 1 convolution of ones
1 x 88 x 1 convolution of ones
88 x 1 x 1 convolution of ones
Scanned areas of the CAM
vertical diameters of the CAM
horizontal diameters of the CAM
88 x 88 x 1 max pooling
Stack of n areas
Sum and multiply by resolution
Volume estimation
Stack of n π estimators
Average
π estimation
Stack of n averaged diameters

Figure 6.5 Scanning module design. The modules apply different sweeps to the CAM in order to derive two estimations: the volume of the biggest object present within the CAM and the π value estimated from it assuming it had a round shape.

6.3.3. Training schedule

The training comprised a total of 50 epochs using both the training and validation dataset. After some testing, optimum hyperparameter configuration was determined with a learning rate of 0.001 with ADAM optimizer and a batch size of 5. As each batch needs to contain tensors of the same shape but the network allows for a variable number of slices, the training dataset was organized accordingly so for each training iteration the batch contained samples with the same size. This was done by picking random samples for each specific size and creating a new batch in every iteration, after which the used samples were discarded for the following batches. Additionally, the training dataset was increase with data augmentation. Specifically, for each generated batch, a new one was created using the same procedure taking random samples from the training pool and applying the transformations to the images. The transformation scheme was defined by the application of a random rotation (between -30° and +30°), a random shear (between -20° and +20°), a random translation (between -15 and +15 pixels) and adding Gaussian noise (mean of 0.035 and standard deviation of 0.01) to the image. As the number of training samples could not exactly be divided by the batch size, at each epoch one case was always left out in both the raw batches and the batches with transformations. This was addressed by forcing that at the beginning of every epoch, the cases left out in the previous one were automatically selected to fill the first raw and transformation batches, while keeping the selection of remaining samples random.

For training, a custom loss function was defined. This loss took into account the L1 activity regularization applied at the bottleneck and CAM output and the mean absolute errors of the estimation obtained for the volumes and π. Each of these parameters contribute to the loss in a different manner due to the difference in scale. L1 applies a penalty that is the summed absolute value of all the features present in the output feature map, so the scale can be very large considering the number of features in both the CAM and the bottleneck (easily in the order of thousands or more). The volume error, being calculated in ml can be in the order of hundreds in the cases with high errors. In contrast, the π estimation error is normally in the order of units. To compensate for this the final loss was the weighted sum of all these components: the π error was given a weight of 100, the L1 activity regularization components were applied a weight of 10^{-3} and the volume error was applied a unitary weight value. The specific values for the components contributions were chosen based on experimentation and observation of the training evolution. Only with these setting a good training evolution was accomplished, which indicates that in this setting an equilibrated contribution of the different parameters was required. The loss function equation is presented in Equation 6.1, where MAE (vol) and MAE (π) represent the mean absolute error for the estimated volume (in milliliters) and for the estimated π value respectively, L1(bn) and L1(CAM) represent the L1 activity regularization factors for the bottleneck feature map and CAM respectively

$$Loss = MAE\ (vol) + 100 \times MAE(\pi) + 10^{-3} \times [L1(bn) + L1(CAM)] \qquad \textit{Equation 6.1}$$

6.3.4. Evaluation method

The trained model was evaluated using the remaining 98 cases of the test set. Two different conditions were evaluated from the results. One was the error associated to the volume estimation, employing both the relative and absolute error (in ml), additionally correlation between the predicted and real values was computed. The second measurement involved measuring the degree of correctness of the CAM generated. For this the CAM was converted to a binary segmentation employing a minimum threshold of 0.9 and then selecting the biggest object within the image (which corresponds to the object whose volume was obtained by the scanning module), this second condition was necessary in some outlier cases, in which we found out that the CAM produced residual regions with high associated probabilities. One example of this situations is presented in Figure 6.6. The derived binary segmentation was compared against the manual segmentations using the Dice coefficient as quality measurement.

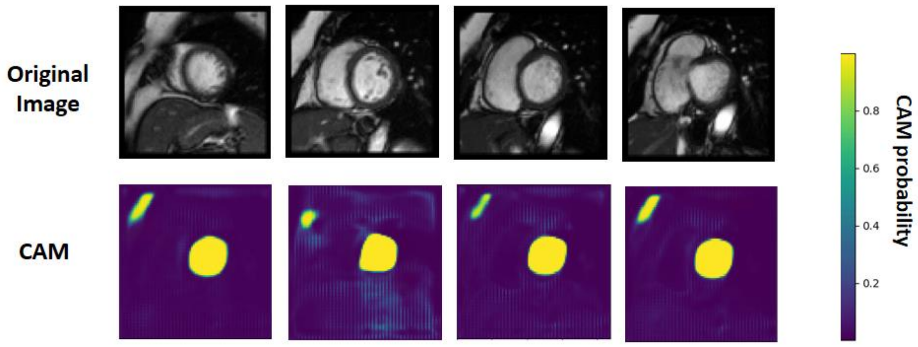


Figure 6.6 Example of a generated CAM that included a residual region with abnormally high probabilities. In rare occasions the model produced this additional small residual objects. These objects where not used in the final predicted volume thanks to the scanning module design, which only used the areas of the biggest object present.

6.4. Results

6.4.1. Training performance

The model was trained for 50 epochs, which required 12 hours to complete. Figure 6.7 shows the training and validation loss evolution across epochs. The training loss continuously decreased, first with high speed and then with dramatically speed reduction starting around epoch 10. The validation loss showed a very erratic behavior in the first epochs, with very pronounced spikes and fluctuations, which indicates some inner difficulty in finding a correct generalizable solution during those epochs. However, after epoch 15 the trend changes dramatically: from epoch 15 to 20 it continuously decreases after a previous high spike, from that point onwards, the validation loss stays approximately still with minor fluctuations. This complex loss evolution might be an indicator that only around epoch 20 the model could find meaningful features relevant enough to avoid overfitting problems.

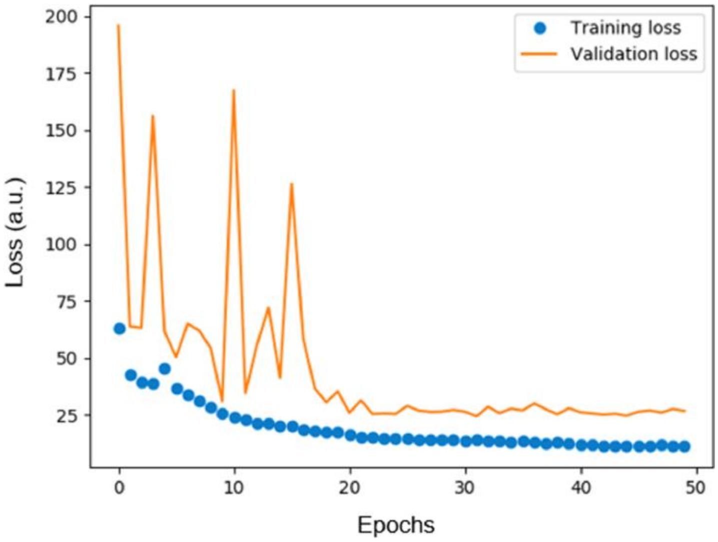


Figure 6.7 Training and validation loss curves registered during the model's training. The model had a difficult time at finding a good and stable generalization setting as seen by the big fluctuations on the validation loss. The model was able to stabilize itself around epoch 20, from which no noticeable changes were observed in the validation loss.

6.4.2. Volume estimation

The absolute error of the predicted volumes by the model followed a distribution of 9.127 ± 18.888 ml (mean and standard deviation). The associated relative absolute error followed a mean and standard deviation of 8.50 ± 6.60 %. The correlation between the predicted and real volumes was R=0.95. The predicted volumes showed some tendency for underestimation of the real LV volumes. This trend appears more evident the bigger the LV. Fitting the results with a regression line obtained a model with a slope of 0.81 and bias of 19.41. These parameters are consistent with the high correlation and the slight underestimation observed. Figure 6.8 shows the scatterplot of the results along the regression line, where the underestimated volumes start to be more clear for LV volumes of 250 ml or more.

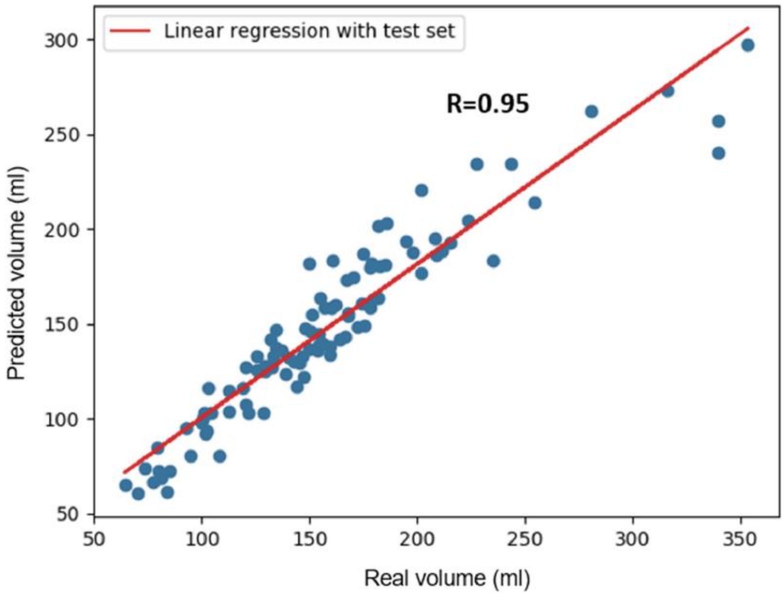


Figure 6.8 Cloud of point of the predicted volume values by the model against the real volumes of the left ventricle. The correlation obtained was very high (R=0.95), the regression curve had a slope of 0.81 and a bias of 19.41 respectively. There is a clear tendency for the model to underestimate the left ventricle volumes to some degree. This tendency becomes more apparent for volumes higher than 250 ml, where the error difference is notably bigger.

The time required for the model for each prediction was on average 1.07 seconds. The average was obtained by predicting the entire test set (98 samples) using a batch size of 1, which took 104.85 seconds to complete.

6.4.3. Derived segmentation

In order to offer a degree of both the explainability power of the model and the weak-supervised method employed to train the model, the masks derived from the CAM were compared against the manual segmentations using the Dice coefficient. Furthermore, a visual exploration was done in order to find any meaningful patterns.

After exploring the images three important features were found. First, the segmentation mask tended to leave a portion of the most external LV region outside of the mask in the majority of slices. Second, specifically for the more basal slices the mask tended to capture small regions outside of the LV. And third, overall all slices were segmented with a region area of similar shape and size and with its location correctly centered at the LV slice center. Figure 6.9 shows a representative example of the resulting masks compared to the manual ones and Fig 6.10 shows a 3D rendering of the same case done with ITK-snap comparing the 3D rendered mask of the prediction and the manually segmented one where these patterns are visually clear.

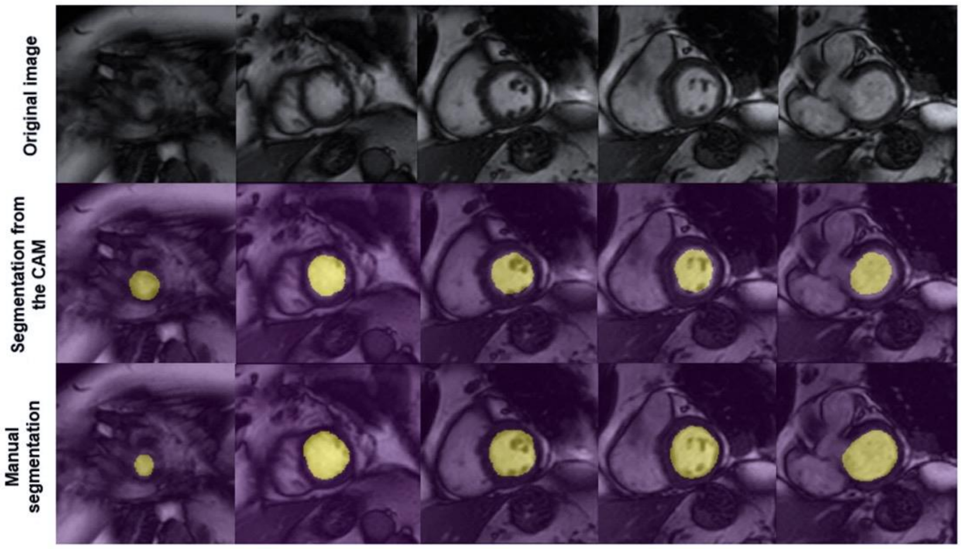


Figure 6.9 Example of the final segmentation derived from the CAM generated by the model. The segmentation was obtained by applying a threshold of 0.9 and selecting only the biggest object present. The derived segmentation is a good match to the manual one, albeit with a slight suppression on the borders except for the apical slice, where the mask’s area is overestimated. In all cases the derived segmentation matched the left ventricle location.

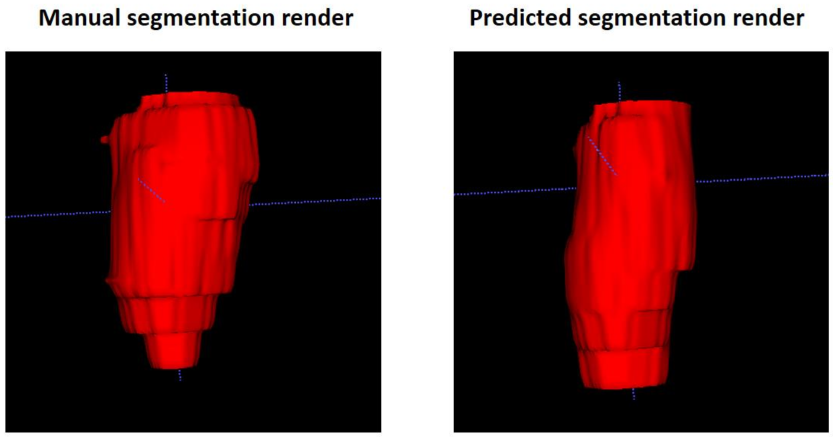


Figure 6.10 Example of a 3D rendering of the manual and derived segmentations on ITK-snap. The manual segmentation is overall thicker than the prediction except for the lower region corresponding to the apical zone. Additionally, the predicted segmentation produced smoother renders compared to the manual segmentation due to the lower difference between the masks at adjacent slices.

In general, the obtained segmentations tended to produce more “tube-like” shapes for the LV compared to the real shape, where it could be seen more as a deformed cone, with smaller slice areas towards the apical regions.

Based on this visual exploration, a quick post-processing step was designed in order to check if these finding were truly consistent. We applied a dilation of size 5×5×1 to all slices in order to increase the degree of overlap. This step significantly improved the results. The original segmentation obtained a Dice coefficient distribution of 0.720± 0.053 and the post-processed masks of 0.791 ± 0.042 (mean and standard deviation). These high improvement matches the visual findings described and demonstrates that these are consistent throughout the predicted masks.

6.5. Discussion

This chapter describes an implementation of a neural network capable of estimating the LV volumes from the ED frames from short-axis cine CMRI and at the same time it can produce a segmentation of the region it based its final prediction from, thus offering a way to explain where the results came from.

If one checks the current literature regarding image-related task for the LV analysis in CMRI, the vast majority focuses on the segmentation problem. This approach usually achieves state of the art results, normally ranging from 0.9 to 0.96 Dice coefficient values (see chapter 5). More specifically the PSPU-net presented in chapter 5 reaches a Dice score of 0.955 with a relative volumetric error of 0.026, which far surpasses the regression U-net results obtained. However, this method requires the previous task of manually segmenting the images and save this segmentation in a fitting way for use in deep learning applications, which is a hard, costly and time-consuming problem. In this sense, directly estimating the target volumes can be a more treatable problem in term of label data, as it is common practice in most clinical settings to save only the final output in the form of patient's reports [131] (in our context, these would be the LV mass, volume, ejection fraction, etc.). This is furthered proved if we check one of the biggest public datasets available for LV volume estimation with more than 1000 patients, where only the volumes are available to train the models (Data Science Bowl Cardiac Challenge Data, *https://www.kaggle.com/c/second-annual-data-science-bowl/data*). Still, this methodology comes with the counterpart of requiring larger dataset, as regression models for image processing are much harder to train than their segmentation counterparts.

There have been very few works directly trying to use CNN to estimate the LV volumes from the images. At the time of writing this thesis only two works were found, both employing the same dataset from the Data Science Bowl Cardiac Challenge containing a total of 1140 subjects. In the work presented in [139] the authors reported correlation values of 0.95 and 0.92 for the ED and ES respectively using 337 cases as a test set. Additionally, the mean error reported was of 5.1 ml and 3.6 ml respectively [139]. In the other work described in [140] the published results were worse with a mean error of 15.83 ml (ED) and 9.82 ml (ES) using as test set a total of 440 cases. These works successfully trained CNN models to directly estimate the LV volumes from the images employing regression architectures, however, these models lacked any explainability capabilities in their design and no report regarding the use of explainable techniques to demonstrate that the models were actually using LV features was provided. Although the results offered were quite good, in a clinical context the predictions would have low trust by the expert clinicians, something that does not happen in segmentation models, were one could always check the segmentation quality.

In contrast to these works, we focused on both problems, the direct estimation of the LV volume and a way to offer explainability of the results by producing a segmentation. The methodology proposed for both problems is notably different than the usual procedure in much of the explainable AI techniques, where after training, some specialized techniques are applied to infer an explainable heat map that can then be converted to a segmentation. In this case the implemented model directly coupled both objectives as the training objective, as the final prediction comes directly from a previous generated CAM that serves as the heat map. This approach is overall the same as weak-supervision. In weak-supervision the objective is to produce complex labels (like segmentations) from less informative ones. In this case the less informative labels would be the LV volumes from which a LV segmentation is desired. It can be seen that there is an important link between weak-supervised learning and explainable neural networks and that both worlds are very intertwined. The proposed model directly uses this link to solve two different possible problems: offering explainability from a regression model and offering a way to produce segmentation without the need of this type of labeling. As another important feature we additionally incorporated into the training schedule prior information in the form of the shape the object should have (circular). This actually helped the model considerably and is what ultimately led the model to target the circularity of the LV within the image.

Even with the satisfactory results obtained by the network there are some important limitations. The predicted volumes are in general good, with small relative error, however there was a clear tendency for underestimation as seen in both the segmentation and predicted values, however the apical slices had a slight overestimation in the CAM. Probably these compensated to some extent the missing target regions. One possible solution to this could be to train the model to predict areas in each slice and separate the volume prediction in different area components to target. A simpler approach could also be to add some post-processing after the predicted CAM just before the scanning module. Based on the fact that applying dilations notably improved the Dice coefficient of the segmentation results. More analysis on the specific parameters that could define the morphological operations to optimize the results would be required, but it seems to be an appropriate approach based on our findings.

Another difficulty was the training itself; section 6.4.1 clearly shows that the model initially had a hard time optimizing itself to a level that could generalize properly, indicating that training this model is hard, this could also be caused by the limited amount of images employed. As previously stated, the regression problem is a hard one that may require datasets in the order of thousands. Still, we could reach a 0.791 average Dice coefficient and a mean error value of 9.127 ml with excellent Pearson correlation (0.95) in the 98 cases of the test set. These results are comparable to the ones reported by the other works employing regression networks for LV volume estimation and show a good match with the real values. This are promising results, however they still fall behind the state of the art results, specially from the neural networks that are trained to directly offer segmentations.

Another limitation is that the model was only tested to target the volumes in ED frame, however, given the nature of the images it is expected that the same scheme could function as well for ES frames after some minor modifications. Probably, the main modification that would be required is the replacement of the area scanning convolutional layer within the scanning module. In this case the size employed was 25×25×1, but considering the smaller size of the LV chamber at ES it could be argued that reducing its size would probably work better in that context. We also want to make note that the overall approach taken could probably be extended to other medical (and non-medical) imaging problems as well after applying the necessary changes depending on the context.

Explainability itself is major concern within the deep learning field for those task that may require justification on the results. Neural networks in general work very much as black boxes were both interpretability and explainability is difficult if not impossible. Giving solutions to this view is important in a field such as radiology, where regression networks may be able to accurately predict some biomarker, but if they do not provide a way for the clinical expert to understand the reason for that prediction, they might as well be very difficult to trust [164, 165]. There are some works discussing how the way most of the explainability techniques work can´t actually give response to the demand of transparency for understanding these models [166]. This is based on the fact that methods that rely on heat maps do not actually give a reasoning on how the features are employed afterwards. There is some debate regarding these topics, but with the methodology described in this work even those suspicions are out of the question, as the scanning module introduced is a simple and deterministic set of operations that work on the CAM and thus this makes for both an interpretable (after the CAM prediction layer) and explainable model. We believe the approach and methodology taken, even with its possible limitations, will be adopted in the near future as the demand for transparent neural networks increases. This will be crucial in the clinical realm, (where delicate decision-making is done affecting people’s health) and is staring to be considered as a serious topic in the legal setting [167].

6.6. Conclusions

This chapter has covered the model design and training approach to directly estimate LV volumes from short-axis cine CMRI at ED and at the same time generate a segmentation that helps explain from where does this model compute the predicted volumes within the image. The model obtained good volume predictions close to the real one in the test set composed of 98 images. The derived mask obtained also showed a good match to that of the LV location within the image, which ensured that the neural network was targeting the correct region to derive its results. We believe the methodology described is important and can impact the way deep learning can be employed from a clinical perspective, since it helps to understand how these complex models offer their results and avoid the black box view of them to an important degree. The described method also helps broadening the options in research directions for weak-supervised learning in order to expand the possibility to use more datasets where, even with less informative labeled data, could still be employed for the LV volume estimation and segmentation.

Chapter 7.
Automatic end-systole and end-diastole detection

This chapter is based on a journal paper published in the context of this thesis [180]. The full document is available at: *https://doi.org/10.1016/j.compmedimag.2022.102085*.

7.1. Introduction and Motivation

The main information that permits the correct characterization of the heart state comes from analyzing the short-axis cine CMRI images at both End-systole and End-diastole. When a segmentation of the regions of interests at both frames is available it is possible to extract the main functional biomarkers. However, before being able to segment these regions it is necessary to determine and select the frames of the ES and ED, as only those will be used. This is also true in the clinical setting, where normally this step requires manual or semiautomatic intervention to select the right frames to use.

As discussed in chapter 5, the problem of segmentation in short-axis cine CMRI is well-studied and with many CNN-based reported solutions that can achieve state of the art results. However, the prior requirement of determining which are the frames to use is often overlooked in these studies. From a practical point of view this is understandable since segmentation is the main bottleneck, but in order to develop fully automatic CAD systems to help in the assessment of the heart the automatic detection of the ES and ED is a necessary step as well, which is required to speed up the clinical workflow.

Detection of the ED and ES frames within an image sequence combines two types of data: image information (spatial domain) and the dynamic relationship of the images information (temporal domain). Within DL, temporal analysis has been traditionally tackled with recurrent neural networks (RNN), which were specially designed to treat time-series data and has been employed for several task including forecasting problems or natural language processing (NLP). Additionally, transformers have been employed in recent years for the same type of tasks and have nowadays relegated RNN to a secondary stage in most time-related tasks due to the superiority of these new architectures.

The usual approach would be to combine CNN with some of these specialized neural networks. However, instead of using these types of architectures to process the temporal information of the acquisition, in this chapter we cover the design and implementation of a fully convolutional neural network with the additional motivation to prove that convolutions are also well suited to treat this type of information in the context of event detection in dynamic medical imaging. All the designs described in this chapters were specifically developed for the detection of ES and ED in short-axis cine CMRI, but the overall design described could be used for any type of event detection problem in a series of dynamic images (video in general), so the application scope remains wide outside the cardiac imaging one.

The overall work described in this chapter covers the design of a fully convolutional neural network capable of detecting the ED and ES frames in a short-axis stack of cine CMRI sequences with both an arbitrary number of frames and of slices per frame. The model makes use of convolutions with dilation rates (dilated convolutions) which have been used for different deep learning problems, most notably for segmentation [104, 105, 168]. Dilated convolutions are as any convolution but with the addition of having an enlarged field of view while keeping the same number of parameters. This is accomplished by introducing zeros between the weights, the size increase of the kernel is determined by a dilation rate which indicates the number of zeros introduced between the convolution weights (the rate is defined as the distance between adjacent weights). Figure 7.1 shows an example of the concept of dilation in convolutional layers.

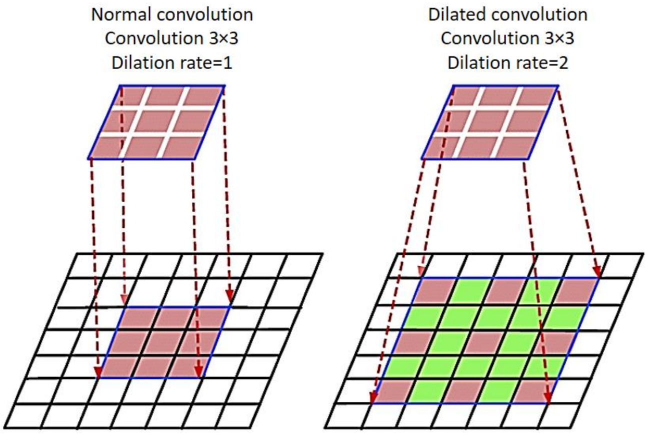


Figure 7.1 Illustration comparing a normal convolution (dilation rate=1) with a kernel of size 3×3 with a dilated convolution with the same kernel (using a dilation rate of 2). Increasing the dilation rate is equivalent to introduce zeros (green squares) between adjacent elements of the kernel. The result is a convolution operation with the same number of parameters but with a bigger field of view [169].

The motivation of using dilated convolutions comes from the paradigm shift that the wavenet introduced when obtaining a superior quality with 1D dilated convolutions over recurrent layers [71]. Although the wavenet was described for a completely different problem (audio generation), it still demonstrated that the use of dilated convolutions could be used to encode temporal information efficiently and surpass the different recurrent neural network layers usually employed to tackle temporal-related data.

Finally, we introduce a novel way of training models for this type of task by employing a different loss function to the classical one used for classification. To do this we train the model under two different configurations, employing a weighted cross-entropy loss and a weighted Dice loss, and compare the results of both. Cross-entropy loss is the usual choice for classification problems [170], and in this case the objective is to classify each frame in a sequence, with the additional limitation that only one frame can be ES and only one can be ED, the remaining being “background frames”. The idea behind using the Dice loss is the use of an overlap measurement that uses the entirety of frame predictions as a whole, which contrast with how cross entropy works where the loss is simply the average of the loss computed per frame.

7.2. Related work and state of the art

There are some works that have addresses the problem of the automatic detection of the ED and ES problem in the context of echocardiography imaging [171-175] and angiographic imaging [176]. However, for short-axis cine CMRI there is a considerable lack of work compared to the segmentation problem. At the time of this thesis writing, to the best of our knowledge only three works have been published addressing this task in some manner. In the work described in [177] the authors used the usual approach of first using convolutional layers to extract spatial features and then these were fed to LSTM layers to process the temporal information. In this case the acquisitions only included one slice per frame and the number of frames was constant with 25. In the remaining 2 works very different approaches were taken. In [130] they simply applied a segmentation CNN to segment the LV in all frames and then selected the ones with biggest and lowest volumes. In the case of [178] the authors used a similar approach by employing a CNN to first segment the LV and then use its center as a relative position parameter whose change was used to determine the ES and ED.

Besides these, there have been some works making use of the temporal information of short-axis cine CMRI but not with the objective of detecting the ES and ED frames. For example, in the work described in [179] the authors developed a generative convolutional neural networks to model a probabilistic motion field that could be used for several tasks, like registration or motion modification on source images. In this work first spatial features were extracted from the different frames and then these were processed with a temporal convolutional layer, which basically consisted on several 1D convolutions with different dilation rates that processes the temporal relation between the features. Although for a different image modality, in [176] a very similar approach for the detection of the ES and ED on coronary angiographies was applied, employing likewise convolution layer to process the temporal information. These works, although not focused in the same task, are still relevant, as they also made use of convolutions to process temporal information in similar medical imaging contexts.

7.3. Material and Methods

7.3.1. Data

For the work described in the current chapter a total of 397 were employed. As the available segmentations had been applied to the previously manually selected frames, the labeling of each frame as ED or ES was done by selecting the volumes that contained myocardium segmentations (present within the ED ones) and the volumes that contained all but the myocardium segmentation (not present within the ES ones).

The dataset was randomly split in training, validation and test sets. All groups included the same percentage of cases for each different pathological description (as described in chapter 4, section 4.1). The training set included 259 cases (65%), the validation set 40 cases (10%) and the test set 98 cases. (25%).

All the images employed were preprocessed before the experiments. The images were first resampled using bi-linear interpolation to an in-plane resolution to 1×1 mm and the image size was set to a constant of 176×176 pixels. For the image resizing cropping and zero-padding was applied when necessary in order to get the desired size. The third spatial axis and the temporal axis was left untouched in both size and resolution. These preprocessing did not affect the presence of the heart within the images, as it was always present in the central region of the image plane. The intensity values for every frame were also normalized to a range of 0 and 1 using min-max normalization. The labels employed consisted of hot-encoded vector for each case with the same length as the number of frames in the acquisition. Each element in the vector contained three values representing the labels ES, ED and background.

The experiments performed had an inherent limitation, since the entire dynamic images were too big to be directly fed to a neural network with the available hardware due to the excessive memory consumption. To solve this, each 3D frame was transformed to a single 2D image, thus eliminating the third axis from the spatial dimensions. The original dataset consisted of 4D stack (3D+time), after the transformation the dataset consisted of 3D stack (2D+time). The transformation applied aimed at generating a single representative image of the contraction state at the frame. For this, a median projection was applied along the third spatial axis between the second and penultimate slice. The use of the median helped avoid possible outliers that could introduce some distortions in the final projection, and not using the first and last slices comes from the fact that some short-axis acquisitions can include regions outside the LV, and in the cases where it still includes the LV these slices will correspond to very extreme region in the apical and basal zones. Overall, the aim was to extract more information from the mid region, where the LV contraction is more predominantly visible. An example of the final median projection employed is presented in Figure 7.2.

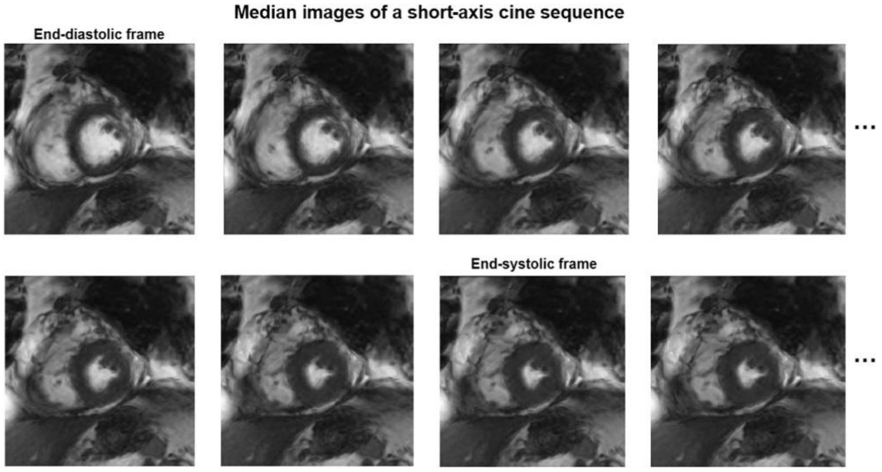


Figure 7.2 Example of a median projection of short-axis CMRI acquisition. The end-systolic and end-diastolic frames are indicated above the corresponding frame. Overall, the images are less clean and the different organs and tissues appear more deformed compared to real slices, but the contraction of both the LV and RV are easily visible, being it more clear in the LV.

7.3.2. Model architecture

The model consists of two different blocks: first a pure 2D convolutional block extracts the spatial features from each image frame, and second a spatio-temporal dilated-convolutional block process the temporal relationships between spatial regions of the output from the previous spatial convolutional block.

The first block takes as inputs arrays of size 176 × 176 × n (n being the number of frames, which is a non-fixed value) and then applies two consecutive 2D convolutions with ReLU activation functions and max-pooling operation to halve the size of feature channels. Each convolution is always followed by a batch normalization layer. The 2D convolutions were implemented as 3D convolutions of size 3×3×1, which are equivalent to a 2D convolution. This scheme is repeated 4 times where at each level the number of channels is doubled and the size of them is halved. At the end of this block the result is a stack of channels of size 11×11 which are then collapsed to a single channel (using a 1 ×1 ×1 convolution) resulting in an array of size 11×11×n×1 obtaining a single, heavily compressed feature map per frame. This output is then passed to the spatio-temporal block. There were two major causes for the decision of using 4 downsampling steps: the first is due to constrains in the image dimensions, as they could only be halved a total of 4 times (additional downsampling would have required padding or cropping operations within the model that we preferred to avoid) and the second and most important is that the relevant features are those corresponding to the heart, which occupies a great portion within the images, by reducing the size as much as possible we allow the next temporal block to have a bigger spatial field of view as well.

The second section takes the array generated by the first block and passes it to different parallel paths that apply the temporal dilated convolutions. Each path applies a dilated convolution of different size in the temporal axis, specifically three kernels were employed (3×3×3, 3×3×5 and 3×3 ×7) in combination with three different dilation rates of 1, 2 and 4 applied in the temporal dimension. This results in a total of 9 paths where each sees 3×3 spatial features along different temporal fields of view. Specifically, the temporal field of view under these conditions spans from 3 to 25 frames, which allows the analysis of short, mid and long-term ranges. Each one of these paths applies a single convolution with ReLU activation function and a batch normalization layer and outputs a single channel. At the end all the paths' outputs are concatenated along the original input of the spatio-temporal block (via a skip connection) and a final last $11 \times 11 \times 1$ convolution followed by a softmax activation function is applied to produce the final predictions. The softmax activation function outputs 3 values for each frame, each corresponding to the probability of said frame of being ES, ED or a background frame, with these probabilities always adding up to 1. The whole model design is presented in Fig 7.3. This architecture had a total of 4.7 million parameters and occupied 54 MB of space.

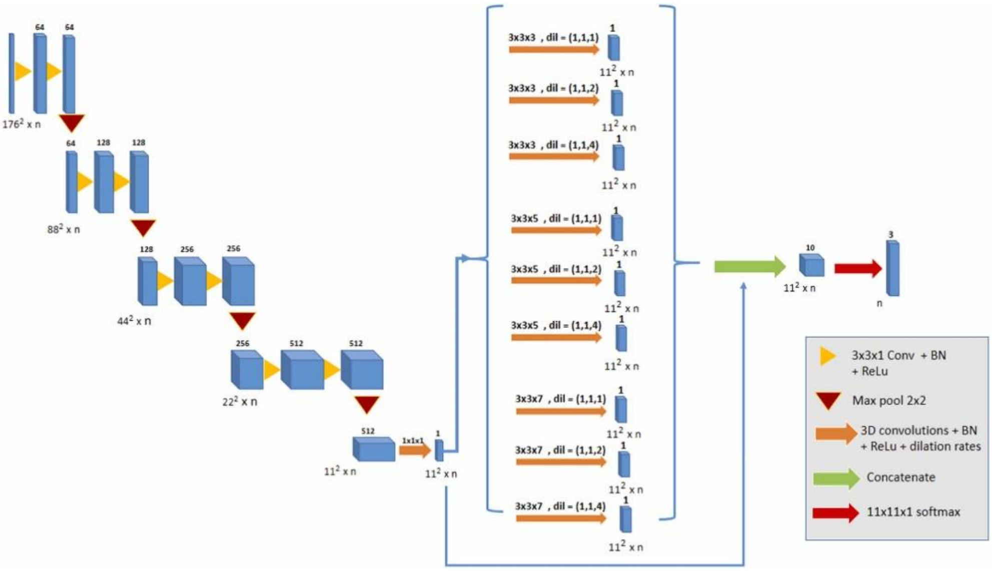


Figure 7.3 Architecture of the model employed. The first section applies 2D convolutional and pooling operations to extract spatial features from each frame. The next section consists of different dilated convolutional layers applied in parallel. The dilated convolutions are 3D convolutions where the third axis is modified with different sizes and dilation rates to enable a different temporal fields of view.

7.3.3. Training schedule

The model training lasted 100 epochs employing both the training and validations sets. The chosen optimizer was ADAM with a learning rate of 10^{-5}. The batch size was limited to 2 for each iteration due to memory constraints. Since processing a batch of data requires all inputs to have the same dimensions, the batch generation was implemented so that no acquisitions with a different number of frames were grouped together, for these we simply divided the training set in different groups depending on its time dimension, then each batch was randomly generated by taking two samples from only a single group each time and making the used samples unavailable for the rest of the epoch. Since the number of samples used was not even, at the end of each epoch a random case was left out and in the next epoch it was automatically selected in the first batch generated. The remaining cases were always selected randomly.

Data augmentation was applied in the training set. The entire dataset size was increased 7-fold, going from the original 259 to 1813 cases. Each case was transformed 7 times with a random combination of the following: random rotations around the image center (between +20 and −20 degrees), random shear (between 10 and −10 degrees) and random translations in both x and y axis (between 44 and −44 pixels). The same geometric transformation was applied in all image frames of each sequence to ensure spatial consistency between frames. Additionally, a random temporal delay was added to each case. Since all acquisition encompassed a single cardiac cycle and these are repeated sequentially in real time, the delay was applied so that the frames that surpassed the dimension length were translated to the beginning of the sequence accordingly. With this each additional delayed sequence had a different location for both ES and ED and the first frame corresponded to a different point in the cardiac cycle. This temporal delay was applied by randomly displacing the entire sequence by a factor between 0% and 40% of the sequence length.

7.3.4. Loss function

The model was equally trained with the described schedule under two different settings regarding the loss function employed: in one case the weighted cross-entropy (WCE) was used and in the other a variation of the generalized Dice loss (GDL) was employed.

The cross-entropy is a classical and popular loss function for classification tasks. It has been extensively employed for image classification, but also for dense predictions like segmentation [128]. In this case we trained the model using the weighted cross-entropy assigning different weights for each frame prediction. Some testing was required to find a satisfactory setting to get acceptable results. In the end we found out that the ES and ED frame required extremely big weights compared to the other frames. Specifically, both the systolic and diastolic frames were assigned a weight of 100, while keeping a unitary weight to the remaining frame predictions. Equation 7.1 shows the formula for the weighted cross-entropy employed. In the formula *NF* is the number of frames, *w* is the weight applied, being 100 for the ES and ED and 1 form the remaining frames, *NC* is the number of classes (3 in our case: ES, ED and background frame), *y* is the hot encoded categorical vector for the frame (i.e. the target), with a value of 1 in the associated category and 0 for the other two, and *p* is the predicted probability vector for the frame.

$$WCE = -\sum_{i=1}^{NF} w_i \sum_{C=1}^{NC} y_{ic} \log(p_{ic}) \qquad \text{Equation 7.1}$$

The second loss function was a modified version of the generalized Dice loss [128]. The generalized Dice coefficient is employed for measuring a Dice coefficient value that weights the predicted segmentation in the cases of class imbalances [129] and has been used as a loss function for segmentation neural networks. The Dice loss has been speculated to be useful for time-related tasks due its capacity to target both sensibility and specificity [181] and has been successfully employed in natural language processing tasks [182]. Considering the problem of event detection, the Dice loss seems like a very good candidate, as we may consider that our problem is equivalent to the classification of vector elements, which in turn is the same as segmentation with the specific characteristic that we only have 1 dimension (the temporal dimension) as opposed to the usual 2 or 3 dimensions in image segmentation. A such, we may consider that the task to solve is equivalent to a segmentation problem in the 1D domain. The loss function employed is a specific weighted Dice loss (WDL) with predefined weights. The assigned weights were 0.45 for both the ES and ED categories and 0.1 for the background category. The formula for this weighted Dice loss is presented in Equation 7.2. In the formula all variables have an equivalent meaning as in Equation 7.1, with the exception of *w*, which represent the weight associated for each class (0.45 for systolic and diastolic classes and 0.1 for background).

$$WDL = 1 - 2\sum_{C=1}^{NC} w_c \frac{\sum_{i=1}^{NF} y_{ic} p_{ic}}{\sum_{i=1}^{NF} y_{ic+} p_{ic}} \qquad \text{Equation 7.2}$$

7.3.5. Prediction post-processing methods

The described neural network produces a probability vector for each frame, where each of the three probabilities must add up to 1. However, between frames there is not a limitation on the output probabilities, meaning that in the end, the model could produce probability vectors where different frames could have similar high probabilities, so an additional post-processing was required to select only 2 frames for the ES and ED class.

Two different post-processing methods were applied and compared. In the first one we simply chose the frame which had the highest probability associated for ES and for ED (naïve method). The second one assumed that there could be adjacent frames with very high probabilities when around either ES or ED location (due to close frames being in very contracted or very relaxed states that could be similar to real ES and ED). To address this, this second method first applies a probability threshold to select only the frames with very high probabilities, and then the frame located at the center position of these frames is selected as either ES or ED. We refer to this method as "central method". The chosen probability threshold was set to 90% (0.9 in the output vector). Figure 7.4 shows a schematic on how the central method operates.

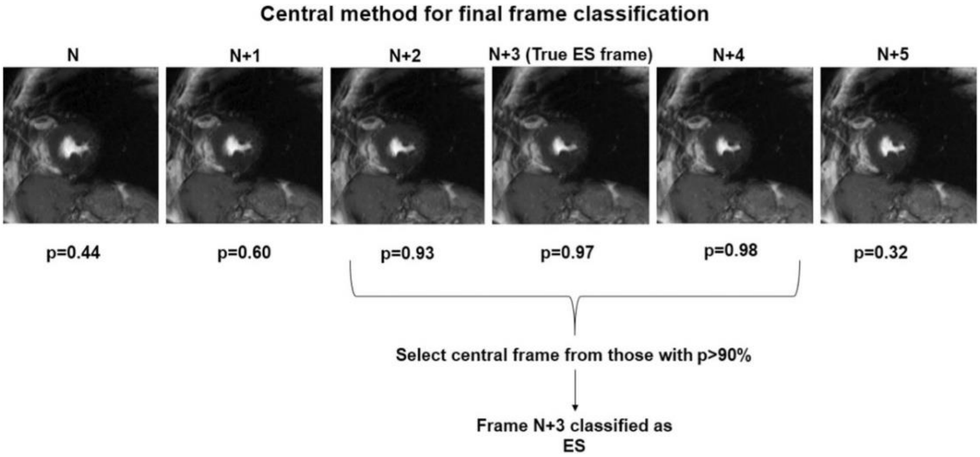

Central method for final frame classification
N
N+1
N+2
N+3 (True ES frame)
N+4
N+5
p=0.44
p=0.60
p=0.93
p=0.97
p=0.98
p=0.32
Select central frame from those with p>90%
Frame N+3 classified as ES

Figure 7.4 Schematic of the central method for final frame classification. First, only the frames with a probability higher than 0.9 obtained by the model are selected, the frame is then chosen as the one located at the central location.

7.3.6. Evaluation method

To evaluate the quality of the final predictions we employed the frame difference error for both the ES and ED, which is the distance in number of frames from the selected frame to the real one. This quality metric has been used previously for the same task [177] and it is an intuitive measure of the result's quality. Additionally, we introduce the relative frame difference error, which is the frame difference error divided by the length of the sequence. This error might be more appropriate for comparison of algorithms performance between different datasets where the number of frames can be very different, allowing for a normalized value that can make experiments mores comparable.

Both quality measurements were calculated for the predictions obtained with the 98 cases of the test set in the four different setting: the combination of the two models (one trained with WCE and the other with WDL) with the two final frame classification methods (naïve and central method).

7.4. Results

7.4.1. Training performance

Figure 7.5 shows the recorded training and validation losses for the two models trained with the different loss functions described, the WCE and WDL.

For the model trained with the WCE loss, the training loss continually decreases in all epochs, reaching a plateau around epoch 60. The validation loss, in contrast, seems to be slightly unstable during the first 20 epochs with a lot of great fluctuations. From epoch 20 the validation loss stabilizes and remains approximately constant until the end, with no signs of overfitting.

The model trained with the WDL shows a similar tendency in the training loss, but reaching the plateau much more quickly, around epoch 40 there does not seem to be any improvements. The validation loss follows a similar trend but with notably lower values than the training loss, and additionally around epoch 60 it starts a slight tendency to increase, which could indicate that the model is starting to slightly overfit.

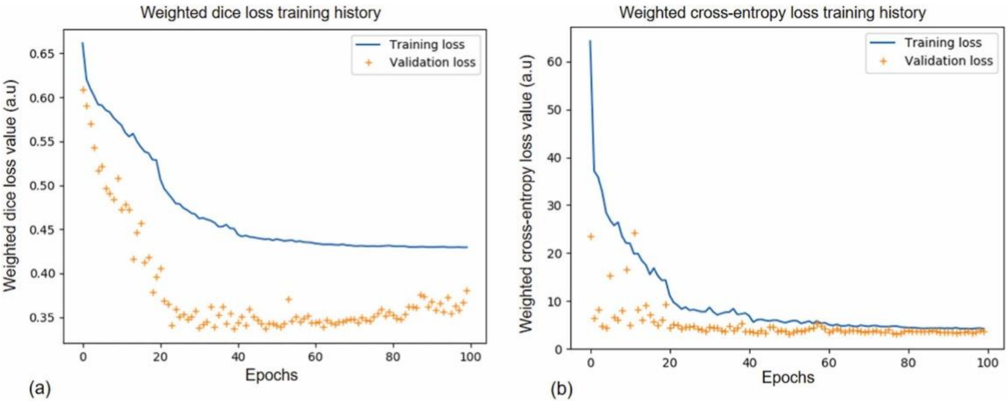


Figure 7.5 History records of the training and validation loss of the model trained with the weighted Dice loss and weighted cross-entropy loss. In both cases the plateau was reached during mid-training at early epochs. The weighted Dice loss (a) reached its training loss plateau around epoch 40, and from epoch 60 onwards the validation loss started to increase, indicating a slight overfitting. Similarly, the weighted cross-entropy loss (b) reached its training loss plateau around epoch 40, but not sign of overfitting is seen in the validation loss, additionally, early epochs show some big fluctuations in the validation loss indicating some initial difficulty at finding a good generalization trend.

An interesting feature of the history training records is that in both cases the validation loss is lower than the training loss (more notably for the WDL case) and that the model reached their optimum status pretty early. The first situation could be due to the fact that the loss values are calculated as the average across batches at the end of each epoch. During training the model has many batches to try different evolution directions that sometimes make it get worse results but in the end it always ends with a similar set of weights to that of the optimum, which is indicated by the lower validation loss. This additionally could mean that the model has very little room to try enough modifications to improve itself without hurting performance. This links directly with the second tendency of reaching the plateau very early, probably due to the model being incapable of finding better weight modifications. All this is easily explained by the model’s inner design. As there are very few layers in the spatio-temporal analysis section, which was an inner limitation due to the limited amount of VRAM at our disposal and the large size of the input arrays.

The whole training process took 22 h to complete with both loss functions, meaning that the choice of loss does not impact the calculations speed. The models selected for the final evaluation were chosen based on the validation loss, specifically the selected model trained with the WCE was the one obtained after epoch 100 (due to lack of appreciable overfitting) and for the model trained with the WDL we selected the optimized one just after epoch 60, since from that point the network seem to lose quality.

7.4.2. Frame detection

The evaluated test included 95 cases with 35 frames, 2 cases with 25 frames and 1 case with 22 frames. The results with the distribution of the frame difference error and its relative version are presented on Table 7.1. The best results are those offered by the model trained with the WDL and with the application of the central method for the final ED and ES selection. This setting achieves a perfect result for the ED detection (error of 0) and very good result for the ES detection with an average frame difference of 1.24 frames (relative error of 0.03). Additionally, it is worth mentioning that the great majority of cases (65%) had an error of 0 or 1, with the remaining cases having errors between 2 and 4 with a decreased number of cases for the bigger errors. There were not any noticeable differences between the cases with 35 frames and the other 3 cases, with errors of 1 frame in 2 cases and one case with a perfect result (error for the ES, as for the ED all cases were correctly detected).

Table 7.1 Frame difference error for the model trained with different loss functions and with the different classification methods employed. Values correspond to mean and standard deviation. The relative error values are presented below in parenthesis. Bold letters indicate the best results achieved.

	ES **(naïve method)**	**ES** **(central method)**	**ED** **(naïve method)**	**ED** **(central method)**
WCE	*3.121 ± 3.500*	*2.505 ± 2.249*	*0.141 ± 0.619*	*0.141 ± 0.619*
	(0.090 ± 0.101)	*(0.072 ± 0.065)*	*(0.005 ± 0.018)*	*(0.005 ± 0.018)*
WDCL	*1.747 ± 1.849*	***1.242 ± 1.45***	*0 ± 0*	***0 ± 0***
	(0.051 ± 0.053)	***(0.036 ± 0.042)***	*(0 ± 0)*	***(0 ± 0)***

The different setting combination did yield expected results. In all the 4 settings, the use of WDL during training resulted in a better model, and the same can be said for the central method, which achieved better results than the naïve method in all setting with the exception of the ED frame, where only the choice of loss function made a difference in the results.

With respect to the inference speed, the average time required for processing a single case was 0.1 seconds using both the naïve and central method (the difference between using one or the other was negligible while employing vectorized programming). This inference refers to the average time required for processing batches of size 1 and includes the final prediction method step.

7.5. Discussion

We have described a methodology employing a fully convolutional neural network capable of detecting the ES and ED frames within a short-axis cine CMRI. The neural network is characterized by the use of dilated convolutions with different dilation rates in order to process temporal information and by being trained with the weighted Dice loss.

Detection of both ES and ED points is a prior necessary step before any analysis can be made, however this problem has not been treated as extensively as the segmentation of the different regions on these frames (a lot of work has been done for left ventricular segmentation). This difference in focus within the research community can be easily explained because segmentation is a clearly more time-consuming problem compare to the ES and ED frame selection. Still, for any automatic CAD system that is to be developed for assessing the heart in CMRI this is still an important and hard task from the algorithm's design point of view. Additionally, even if it does not take that much time compared to manual segmentation, the selection of the correct frames can still take some time depending on the user's experience. In a radiological setting when lots of patients require fast diagnosis, saving this time can further improve the workflow of the clinical experts.

To the best of our knowledge, we can only compare our work against three other previous works done in the matter. The model used in [177] employed the more widespread approach of combining a convolutional block with recurrent layers (LSTM modules) to process the temporal information. In this work the authors reached an average frame difference error of 0.38 and 0.44 for ED and ES respectively using a 4-fold cross-validation methodology on a dataset comprising of single-slice sequences with a constant of 20 frames. The results are better for ES and worse for ED than our method, however it is noteworthy that the datasets employed are considerably different, with our acquisitions consisting of 3D volume frames with both a variable number of slices and of frames, with the vast majority having 35 frames, which in addition had to be transformed to lower quality image representations of the volume frames via the median projection. One approach simply made use of a previous segmentation on the LV throughout all the frames, but no report in any type of quality measurement is provided [130]. In this case we can't know how well their detection went, however it is clear that if the possibility of a high accurate segmentation is available, selecting the frames in this way is the most obvious method. However, we think this is a very impractical methodology, as this would require having lots of different segmented frames in order to properly train a model for full segmentation on the entire cine acquisition, and additionally the predictions should be applied in all the frames, making the automatic process slower. Lastly, in another work the authors employed a CNN to first segment the LV in much longer sequences of 1 slice [178]. The sequences in this case had a constant duration of 84 frames. The authors used the relative location of the center of the LV to determine the ES and ED. They use 10 cases with 10 different sequences, each covering different regions of the heart, to validate their method. They reported an accuracy of 75%, which increased to 95% when only the mid regions were analyzed. In this case the way they analyzed the classification and the sequence's nature is very different.

The proposed model has more similarities with the model described in [177], where we use a similar overall design but instead of using LSTM layers we use different dilated convolutional paths to process the temporal information. LSTM and GRU layers are the most successful type of recurrent layers for time-series analysis [183], however they can be harder to train and more unstable compared to convolutional layers [184, 185]. Besides, the use of dilated convolutions has already proven to be an efficient method for time-series analysis [71, 176, 179], being capable of retaining long-term information better than recurrent layers [71].

The described model used the weighted Dice loss in order to obtain better results than the more widespread weighted cross-entropy loss. The experiments clearly demonstrated that the use of the Dice coefficient is beneficial for training the model, and seeing how other works have obtained superior results as well with its use in the context of NLP [182], we speculate that it could also be used for any temporal classification task with improved results. Probably the main reason of the superiority of the Dice loss is due to it being computed at the label level, employing all temporal point at the same time, while other functions like cross-entropy computes the value for each frame and then applies an average. This is actually similar to segmentation problems, where the Dice loss has proven to be superior to cross-entropy as well [110] and our problem could very much be viewed as a segmentation task in 1 dimension.

There are important limitations to consider in the model design. Mainly in the temporal convolutional blocks, where only 1 channel was obtained per convolution, resulting in a very shallow network. This limitation came from the memory constraint of using large tensors (size of 176 × 176 × 35 in the vast majority). An alternative approach could have been reducing the size of inputs in the spatial domain, for example with images of size 88 × 88, but considering that the images were already derived from the original images and were not a perfect representation of the spatial features of the real volume (median projected images), we preferred to maintain as much spatial information as possible. Additionally, considering the disentangled number of spatial and temporal features we can see how a lower number of kernels could extract enough information from the temporal dimension (the spatial domain has a total 30976 features represented by the number of pixels, whereas the temporal had at most 35 represented by the number of frames). This proved to be true considering the obtained results, but at the same time the loss history demonstrated that the model had some inner limitation that could not be surpassed at early epochs. Probably enlarging the number of kernels per convolution layer in the dilated convolution block could remove this limitation and reach even better results.

With all things considered, the errors obtained demonstrate that the model is very accurate. We introduced the relative frame difference error besides the standard frame difference as a better way to estimate the quality of the results and believe it could be used to better compare methods in future works. In our case, with our test set of 98 cases we could achieve a relative frame difference error of 0.03 for the ES (and with 65% of cases having a zero error results), meaning that on average the distance from the selected frame to the real one was 3% of the entire sequence length. For the ED the results were perfect in all cases. The error obtained for ES is small enough to not have a relevant clinical impact on the final estimated biomarkers, as very close frames yield very similar contraction states, which results in small differences in the final volume measurements in the worst case.

There is another important finding in that for the ES the error was higher, which is also consistent with the results of others works [177]. The explanation for this is probably because of the shape the LV (which is the major contracting element) takes at ES and ED. At ED the LV is usually round, while at ES its shape is, while still round, more irregular and smaller. Additionally, the adjacent frames are notably more similar between them around the ES than at ED (check Figure 7.2 for visualize this clearly) which also contribute at its higher detection difficulty.

Another important point is that the real ES and ED varies depending on the region. These differences are not very big along the heart but for a perfect determination one would need to detect the real ES and ED for each available slice. We did not test this, as our data did not allow for it and because it is not usually done in the clinical context. Still, the described model could be used in such a context without problems, as it can take inputs of any slice, and for the extreme case of one slice the inputs would not be modified by the median operation applied at preprocessing.

In conclusion, the results obtained are very promising. With higher computation resources an improved version could be easily designed and implemented. Finally, we want to further highlight that the proposed pipeline is suitable not only in the case of short-axis cine CMRI, but for any type of dynamic imaging in general, as the whole design was made with the vision of processing any type of input shape (variable number of slices and of number of frames).

7.6. Conclusions

This chapter has covered the design and implementation of a fully convolutional neural network capable of detecting the end-systolic and end-diastolic frames in short-axis cine CMRI with high accuracy. The main elements that characterized the model were the use of dilated convolutions to process the temporal relationships between the frames and the use of the weighted Dice coefficient as a loss function to improve the final trained model in contrast to the classical cross-entropy loss.

The design taken additionally demonstrated its potential considering the low number of parameters of the neural network, especially in its spatio-temporal section. Overall, the proposed model shows promising results that are good enough to be used in a clinical scenario, with perfect estimations for ED frames and very accurate matches for the ES frames. Additionally, the inner model designs and the proposed preprocessing pipeline allows it to work with sequences with a variable number of frames composed of volumes of any size, including single-slice sequences.

There were important limitations for the design of this model, mainly due to the large size of inputs which limited its potential with a minimum depth implementation of the layers in the temporal analysis. However, even with these restrictions the neural network still proved to be able to get excellent result in the task at hand. With this in mind we believe that the overall design of this architecture and training profile could be employed for any type of time-series problems, widening its potential applications.

Chapter 8.
Final conclusions

8.1. General overview

In this project, the applications of convolutional neural networks for the design of automatic Computer-Aided-Diagnosis in cardiac MRI has been extensively studied and different novel implementations have been presented.

A fully automatic CAD system is intended to process the medical images and provide different biomarkers of interest to the clinical expert without intervention. As such, several parts may conform the CAD system. The most usual way is to segment the region of interest from which to extract the different biomarkers. In this sense, CNN are very efficient for segmentation tasks. The PSPU-net described was capable of producing high quality segmentations of the short-axis cine CMRI images and proved to be superior to a heavier 3D U-net.

Another approach that is sometimes employed in deep learning applications is the use of automatic classification or regression CNN models. CAD systems that make use of these models can benefit of an easier training scheme design if only the biomarker values are available for training, but it comes with the important cost of eliminating outputs that can help the clinical experts to correctly interpret the results. This does not happen when segmentations are available, as they serve this purpose. Is in this setting where explainable-AI is a must. The employment of weak-supervision methods can help provide direct full segmentations using the biomarkers available, and then use this segmentation to derive the biomarker. This approach is taken in this project, as it provides a better way to offer explainability compared to other explainable AI techniques that can only provide heat maps highlighting the most important regions but fail at providing a way to interpret how these highlighted features are then employed by the model to reach their prediction. The regression U-net proposed here allows to produce a segmentation that is guaranteed to be the one used to obtain the biomarkers with an easy-to-interpret scanning module. The model was not perfect due to the difficulty of training with these conditions with a limited dataset, but still could achieve very low errors in its predictions, and the derived segmentation had a good overall quality, always matching the region of the left ventricle which indicated a very good quality on the explainability power of the model.

Depending on the problem at hand, a full CAD system may require additional image processing steps besides segmentation and biomarker computation. This is also true for the case of short-axis cine CMRI where the segmentations are required for two frames (end-systole and end-diastole) out of several available frames. An automatic CAD for cardiac assessment necessarily requires to detect these frames as a first step. Noteworthy, this is something that seems to not have caught the attention of the research community very much. In this work and additional model was designed with the exclusive use of convolutional layers that employed dilation rates for the temporal analysis and additionally a very effective way to train it with the use of the Dice loss is described. The model was heavily limited in its temporal processing capabilities due to the large memory consumption of the inputs, but still managed to get excellent results with the best settings, indicating that the overall design is the core of its functional performance and with more resources further improvement could be made by simply increasing the layer's depth.

The general overview of this work presents the main elements to consider to apply convolutional neural networks for developing a fully automatic CAD system for cardiac MRI assessment. The work focuses on the case of short-axis cine CMRI acquisitions analysis due to them being the more general and informative images employed in the clinical context, but additional biomarkers could be computed for other cardiac acquisition protocols using the same or slightly modified versions of the methods described.

8.2. Limitations

8.2.1. Dataset

The dataset employed was limited but sufficiently large to study all the different deep learning methodologies applied. The cohort was specifically taken from patients that had been clinically assessed and whose images had been analyzed and labeled within the clinical diagnosis workflow.

Still, as has been appointed multiple times, deep learning models shine the most the more data one can feed them. The number of required samples varies from problem to problem, overall harder problems will require larger datasets. As seen in the obtained results, the number of samples was enough to generate high quality results, with the notable exception of the regression U-net model which could achieve good results, but not as excellent as for segmentation or frame detection. This was expected since that task was in nature considerably harder than the others.

Finally, it must be mentioned that the employed dataset was limited to a single machine acquisition, so increasing the dataset with images coming from different sources could also be determinant at being able to generalize the models defined for different scanners.

8.2.2. Hardware

All the described methods employed were mainly limited by the GPU employed. The GPU model RTX 2080 Ti that was used corresponded to one of the best from the GeForce series at the moment this thesis began. However, since then, newer and more powerful models have been launched. Mainly the 3000 series, with the RTX 3090 Ti (24 GB of VRAM) staying at the top of the GeForce series with more than doubling the VRAM of the RTX 2080 Ti. At the moment of writing this thesis the 4000 series were announced and rapidly launched to the market. These come with even higher computational capabilities but are still equivalent to the 3000 series in terms of the available memory.

Training the described models with a more powerful GPU like some of the 3000 or 4000 series could also help in the results. Another option could be to add additional GPUs to work in parallel. The main bottleneck in this context is the available VRAM, as it will determine how big a model can be trained and the reachable batch size. Both factors can have a very important impact on the final predictive power of the trained model. This was especially true for the frame detection model, whose size had to be strongly limited due to the large size of the inputs, which also limited the batch size considerably. We believe this model could be further improved just by increasing the GPU capabilities.

We only mentioned the Nvidia Geforce alternatives for GPU, since although they are expensive, they are still affordable for what they offer. However, if one is not restricted by monetary constrains, even more powerful models can be acquired like those of the Nvidia high-end Tesla series, or use cloud computing with TPUs, albeit this will most probably incur in exponential increase in costs.

8.3. Future lines of work

8.3.1. Using additional datasets

The approaches taken could be furtherly tested with additional dataset apart from the one we had access to. In this sense, we have recently found two relatively big public datasets that can be used as additional data sources.

The first one is the ACDC challenge dataset (*https://acdc.creatis.insa-lyon.fr/description/index.html*) which includes 100 cases for training and 50 for testing. The second is the Multi-Centre, Multi-Vendor and Multi-Disease Cardiac Segmentation (M&Ms) Challenge (*https://www.ub.edu/mnms/*) which includes 175 cases for training and 200 cases for testing.

Both of these datasets include segmentation labels for the end-diastole and end-systole and includes acquisition for different machine brands. Adding these to our own dataset could notably increase the number of cases and provided a more varied image pool with respect to scanners employed.

8.3.2. New architectures and training designs

One of the future lines of work for further development of this thesis results is to test variations of architectures and training designs. The reader should have noted that in all described models there were some common features. We briefly describe some possible alternatives as an initial start point for anyone interested in trying the models with some modifications.

Regarding the training designs, all the described models employed ADAM optimizer for training, the use of batch normalization in each convolutional layer and the use of ReLU activation function. There are many different options that could have been chosen, but we stick with these mainly due to them being the most extensively employed ones in deep learning designs and we believed that to prove the quality of the presented architectures it made more sense to use the most used design features. One future line of work would to be to test the same models under different configuration conditions. We will mostly refer to options described in the convnext design [75] where as indicated by the authors can obtain state of the art results, surpassing some of the current trends employing vision transformers and more classical CNN implementations. We will mention some interesting ones that could be tested to replace the ones employed in this thesis:

- **Batch normalization alternatives**: there are some works describing the use of other layer standardization procedures. Some examples could be instance normalization [186] or layer normalization [187]. These basically function similarly to batch normalization but calculate the inner statistics from each sample's own features, thus having the advantage of being applicable for mini batches with just one sample. Additionally, there is evidence that proves that these methods could be superior to batch normalization in some contexts [73, 75, 82].
- **ADAM optimizer alternatives:** in general ADAM (adaptative moment estimation) is a very strong optimizer that usually leads to faster convergences compared to other optimizers, however for specific cases a classic optimizer like SGD (stochastic gradient descent) with momentum might be preferable as there is also evidence that SGD, even if slower, can result in better generalization [188]. To mention the one proposed in the convnext, we also have ADAMW which adds a weight decay factor in the form of L2 regularization [189], this optimizer is starting to get notable attention.
- **ReLU alternatives:** there are many activation functions at disposal as described in chapter 3. However, we consider interesting the GELU (Gaussian Error Linear Unit) [190] which has been used in some notorious transformer architectures like GPT-3 [191] or BERT [192]. Additionally, the very recent GCU (growing cosine unit) is proclaimed to be superior to any variant of ReLU and capable of solving the XOR limitation in single artificial neurons [49].

With respect to the model's architectures themselves there could be many different approaches. We describe some simple examples for the model presented in chapter 6 and 7.

The U-net regression model presented could perhaps be modified in its scanning block to change the way the circular features are calculated, introducing some morphological operations. Additionally, other features from the object could be computed beforehand and used as additional targets (i.e: the average intensity within the

LV). Another interesting approach could be testing the same architecture to process only 2D slices, considering the nature of the results, individual slice processing could be an interesting option in comparison to averaging the roundness properties through the slices.

The ES/ED classification model could be improved with deeper layers. Alternatively, the entire temporal analysis block could be replaced by a transformer architecture, which are based on completely different paradigms from that of convolutional neural networks and are the current standard for time series analysis, having proven its power in very complex problems like NLP [89, 191] or protein structure prediction (one of the most complex challenges within biological science nowadays) [192]. Another interesting thing to try in the future is seeing if adding and additional block could suppress the necessity of the central method described for the final classification. Such a block could, for example, take the value prediction of each frame at the ES and ED classes, apply some 1D convolution on the vectors and again apply a softmax activation function. This would give for the entirety of the sequence a set of probabilities that would sum to 1 for all the frames, and if the model was able to learn some inner characteristics of the prediction distributions maybe simply applying the naïve method afterwards could obtain comparable results.

All the described options are just examples and hypothesis that we intend to try in the future, but in any case, considering the wide variety of option within the deep learning field many others could be thought as well.

8.3.3. Beginning-to-end CAD model training

Another future line of work would be designing a model that directly couples the frame detection with the segmentation of the regions of interest (via either a segmentation CNN or a regression CNN with explainability capabilities). A straightforward approach would be to couple both models after training them separately but it would be more interesting to train a full beginning-to-end system that would represent the final CAD system itself. This would require feeding the entire image acquisition and probably use multiple loss functions to get optimum results, as the gradient would need to flow from the segmentation or predicted volumes obtained for the whole sequence passing through the initial frame classifier.

8.3.4. Generative models for synthetic image generation

One of the most interesting lines to follow in the future is that of synthetic image generation, as it is one of the current main applications to computer vision but was left out of this work due to the very specific nature of its application and for time limitations. The main interest of trying generative models is for data augmentation purposes. Having a good enough model of this kind could allow to generate new instances along with labels on the fly, and additionally other generative techniques can alter an original image, for example to introduce a disease within an image of a healthy subject. All these characteristics could be very helpful in the radiology context, where bigger and more balanced dataset could be created. Furthermore, these additional images would not have any use or access constrains, as they would not come from real patients.

There are many available generative deep learning models. To name some important ones we have variational autoencoders (VAE) which try to capture a distribution representation from which to sample new instances; generative adversarial networks (GAN) that uses a generator against a discriminator in order to produce high quality images until the discriminator cannot tell them apart from real samples; or diffusion models which use a Markov chain approach to iteratively reverse the addition of noise to images that have been extensively corrupted.

8.3.5. Few-shot learning

Another interesting line of work is the investigation of few-shot learning approaches. These are machine learning methods that aim to design training methodologies to enable models to learn from very few samples [193], reaching even the limit of using just one sample (one-shot learning). This could also be interesting in the medical imaging field, as data is usually scarce and hard to obtain, especially the label information.

8.4. Conclusions

This work describes several key applications of convolutional neural networks to Computer-Aided-Diagnosis systems in the context of cardiac MRI assessment. The project covers in several chapters the full image processing with convolutional neural networks pipeline that should be considered for this type of system. Several novel designs and implementations of deep learning methodologies and models have been described, including the topics of segmentation, explainable-AI, weak-supervised learning and event-detection. The models described obtained promising results that allow for the implementation of a fully convolutional neural network system for the assessment of the heart in cardiac MRI with high quality, enabling its use in real clinical context

Chapter 9.
References

[1] L. L. Ankile, M. F. Heggland, and K. Krange, "Deep Convolutional Neural Networks: A survey of the foundations, selected improvements, and some current applications," Nov. 2020, doi: 10.48550/arxiv.2011.12960.

[2] J. Baric-Parker and E. E. Anderson, "Patient data-sharing for AI: ethical challenges, catholic solutions," *The Linacre Quarterly*, vol. 87, no. 4, pp. 471–481, May 2020, doi: 10.1177/0024363920922690.

[3] L. Nordling, "A fairer way forward for AI in health care," *Nature*, vol. 573, no. 7775, pp. S103–S105, Sep. 2019, doi: 10.1038/D41586-019-02872-2.

[4] C. W. Tsao *et al.*, "Heart disease and stroke statistics-2022 update: a report from the American Heart Association," *Circulation*, vol. 145, no. 8. Lippincott Williams & Wilkins Hagerstown, MD, pp. E153–E639, Feb. 22, 2022. doi: 10.1161/CIR.0000000000001052.

[5] N. Townsend, L. Wilson, P. Bhatnagar, K. Wickramasinghe, M. Rayner, and M. Nichols, "Cardiovascular disease in Europe: epidemiological update 2016," *European Heart Journal*, vol. 37, no. 42. Oxford University Press, pp. 3232–3245, Nov. 07, 2016. doi: 10.1093/eurheartj/ehw334.

[6] K. M. Namara, H. Alzubaidi, and J. K. Jackson, "Cardiovascular disease as a leading cause of death: how are pharmacists getting involved?," *Integr. Pharm. Res. Pract.*, vol. 8, p. 11, Feb. 2019, doi: 10.2147/IPRP.S133088.

[7] F. Von Knobelsdorff-Brenkenhoff, G. Pilz, and J. Schulz-Menger, "Representation of cardiovascular magnetic resonance in the AHA / ACC guidelines," *J. Cardiovasc. Magn. Reson.*, vol. 19, no. 1, pp. 1–21, Sep. 2017, doi: 10.1186/S12968-017-0385-Z/TABLES/17.

[8] S. E. Petersen *et al.*, "Reference ranges for cardiac structure and function using cardiovascular magnetic resonance (CMR) in Caucasians from the UK Biobank population cohort," *J. Cardiovasc. Magn. Reson.*, vol. 19, no. 1, pp. 1–19, Feb. 2017, doi: 10.1186/S12968-017-0327-9/FIGURES/5.

[9] S. V. Babu-Narayan, G. Giannakoulas, A. M. Valente, W. Li, and M. A. Gatzoulis, "Imaging of congenital heart disease in adults," *Eur. Heart J.*, vol. 37, no. 15, pp. 1182–1195, Apr. 2016, doi: 10.1093/EURHEARTJ/EHV519.

[10] P. J. Kilner, T. Geva, H. Kaemmerer, P. T. Trindade, J. Schwitter, and G. D. Webb, “Recommendations for cardiovascular magnetic resonance in adults with congenital heart disease from the respective working groups of the European Society of Cardiology,” *Eur. Heart J.*, vol. 31, no. 7, pp. 794–805, Apr. 2010, doi: 10.1093/EURHEARTJ/EHP586.

[11] P. DeSaix *et al.*, *Anatomy & Physiology* . OpenStax, 2013. Accessed: Oct. 12, 2022. [Online]. Available: https://openstax.org/details/books/anatomy-and-physiology

[12] E. N. Marieb and K. Hoehn, *Human Anatomy & Physiology*, 9th ed. Boston: Pearson Education, 2013. Accessed: Oct. 12, 2022. [Online].

[13] S. E. Petersen, M. Y. Khanji, S. Plein, P. Lancellotti, and C. Bucciarelli-Ducci, “European Association of Cardiovascular Imaging expert consensus paper: a comprehensive review of cardiovascular magnetic resonance normal values of cardiac chamber size and aortic root in adults and recommendations for grading severity,” *Eur. Hear. J. - Cardiovasc. Imaging*, vol. 20, no. 12, pp. 1321–1331, Dec. 2019, doi: 10.1093/EHJCI/JEZ232.

[14] J. E. Hall and M. E. Hall, *Guyton and Hall Textbook of Medical Physiology* , 14th ed. Elsevier , 2020. Accessed: Oct. 12, 2022. [Online]. Available: https://www.elsevier.com/books/guyton-and-hall-textbook-of-medical-physiology/hall/978-0-323-59712-8

[15] A. C. Adler, B. H. Nathanson, K. Raghunathan, and W. T. McGee, “Effects of body surface area-indexed calculations in the morbidly obese: a mathematical analysis,” *J. Cardiothorac. Vasc. Anesth.*, vol. 27, no. 6, pp. 1140–1144, Dec. 2013, doi: 10.1053/J.JVCA.2013.06.011.

[16] M. RD, “Simplified calculation of body-surface area,” *N. Engl. J. Med.*, vol. 317, no. 17, pp. 1098–1098, Oct. 1987, doi: 10.1056/NEJM198710223171717.

[17] V. Bodí *et al.*, “Usefulness of a comprehensive cardiovascular magnetic resonance imaging assessment for predicting recovery of left ventricular wall motion in the setting of myocardial stunning,” *J. Am. Coll. Cardiol.*, vol. 46, no. 9, pp. 1747–1752, Nov. 2005, doi: 10.1016/J.JACC.2005.07.039.

[18] J. Sandstede *et al.*, “Age- and gender-specific differences in left and right ventricular cardiac function and mass determined by cine magnetic resonance imaging,” *Eur. Radiol.* , vol. 10, no. 3, pp. 438–442, 2000, doi: 10.1007/S003300050072.

[19] A. Fuchs *et al.*, “Automated assessment of heart chamber volumes and function in patients with previous myocardial infarction using multidetector computed tomography,” *J. Cardiovasc. Comput. Tomogr.*, vol. 6, no. 5, pp. 325–334, Sep.

2012, doi: 10.1016/J.JCCT.2012.01.006.

[20] A. Fuchs *et al.*, “Normal values of left ventricular mass and cardiac chamber volumes assessed by 320-detector computed tomography angiography in the Copenhagen General Population Study,” *Eur. Hear. J. - Cardiovasc. Imaging*, vol. 17, no. 9, pp. 1009–1017, Sep. 2016, doi: 10.1093/EHJCI/JEV337.

[21] A. G. Gheorghe *et al.*, “Cardiac left ventricular myocardial tissue density, evaluated by computed tomography and autopsy,” *BMC Med. Imaging*, vol. 19, no. 1, pp. 1–9, Apr. 2019, doi: 10.1186/S12880-019-0326-4/TABLES/5.

[22] D. J. Griffiths and D. F. Schroeter, *Introduction to quantum mechanics*, 3rd ed. Cambridge university press, 2018. Accessed: Oct. 12, 2022. [Online]. Available: https://books.google.es/books?hl=es&lr=&id=LWRnDwAAQBAJ&oi=fnd&pg=PA3&ots=l1iy8ICeP2&sig=Wi5Ep9x-Qyvitww0j43SF9xgtNo&redir_esc=y#v=onepage&q&f=false

[23] E. Merzbacher, *Quantum mechanics*, 3rd ed. Wiley, 1998. Accessed: Oct. 12, 2022. [Online]. Available: https://www.wiley.com/en-dk/Quantum+Mechanics%2C+3rd+Edition-p-9780471887027

[24] D. Moratal, M. E. Brummer, L. Martí-Bonmatí, and A. Vallés-Lluch, “NMR Imaging,” *Wiley Encyclopedia of Biomedical Engineering*. John Wiley & Sons, Ltd, Hoboken, NJ, USA, pp. 2590–2606, Apr. 14, 2006. doi: 10.1002/9780471740360.EBS0843.

[25] D. W. McRobbie, E. A. Moore, M. J. Graves, and M. R. Prince, *MRI from picture to proton*, 2nd ed. Cambridge University Press, 2006. doi: 10.1017/CBO9780511545405.

[26] J. P. Ridgway, “Cardiovascular magnetic resonance physics for clinicians: Part I,” *J. Cardiovasc. Magn. Reson.*, vol. 12, no. 71, pp. 1–28, Nov. 2010, doi: 10.1186/1532-429X-12-71/TABLES/2.

[27] J. M. Wardlaw *et al.*, “A systematic review of the utility of 1.5 versus 3 Tesla magnetic resonance brain imaging in clinical practice and research,” *Eur. Radiol.*, vol. 22, no. 11, pp. 2295–2303, Jun. 2012, doi: 10.1007/S00330-012-2500-8/TABLES/3.

[28] A. Laader *et al.*, “1.5 versus 3 versus 7 Tesla in abdominal MRI: a comparative study,” *PLoS One*, vol. 12, no. 11, p. e0187528, Nov. 2017, doi: 10.1371/JOURNAL.PONE.0187528.

[29] O. Kraff, A. Fischer, A. M. Nagel, C. Mönninghoff, and M. E. Ladd, “MRI at 7 tesla and above: demonstrated and potential capabilities,” *J. Magn. Reson. Imaging*, vol. 41, no. 1, pp. 13–33, Jan. 2015, doi: 10.1002/JMRI.24573.

[30] A. Sadeghi-Tarakameh *et al.*, “In vivo human head MRI at 10.5T: a radiofrequency safety study and preliminary imaging results,” *Magn. Reson. Med.*, vol. 84, no. 1, pp. 484–496, Jul. 2020, doi: 10.1002/MRM.28093.

[31] T. F. Budinger *et al.*, “Toward 20 T magnetic resonance for human brain studies: opportunities for discovery and neuroscience rationale,” *Magn. Reson. Mater. Physics, Biol. Med. 2016 293*, vol. 29, no. 3, pp. 617–639, May 2016, doi: 10.1007/S10334-016-0561-4.

[32] D. Saloner, J. Liu, and H. Haraldsson, “MR physics in practice. how to optimize acquisition quality and time for cardiac MR imaging.,” *Magn. Reson. Imaging Clin. N. Am.*, vol. 23, no. 1, pp. 1–6, Feb. 2015, doi: 10.1016/j.mric.2014.08.004.

[33] A. Sodickson, “Breaking the magnetic resonance imaging acquisition speed barrier: Clinical implications of parallel imaging,” *Applied Radiology*, pp. 6–18, Feb. 02, 2004. Accessed: Oct. 12, 2022. [Online]. Available: https://www.appliedradiology.com/articles/breaking-the-magnetic-resonance-imaging-acquisition-speed-barrier-clinical-implications-of-parallel-imaging

[34] R. M. Menchón-Lara, F. Simmross-Wattenberg, P. Casaseca-de-la-Higuera, M. Martín-Fernández, and C. Alberola-López, “Reconstruction techniques for cardiac cine MRI,” *Insights Imaging*, vol. 10, no. 1, pp. 1–16, Dec. 2019, doi: 10.1186/s13244-019-0754-2.

[35] M. S. Nacif, A. Zavodni, N. Kawel, E. Y. Choi, J. A. C. Lima, and D. A. Bluemke, “Cardiac magnetic resonance imaging and its electrocardiographs (ECG): tips and tricks,” *Int. J. Cardiovasc. Imaging*, vol. 28, no. 6, pp. 1465–1475, Aug. 2012, doi: 10.1007/S10554-011-9957-4/FIGURES/11.

[36] R. D. Kacere, M. Pereyra, M. A. Nemeth, R. Muthupillai, and S. D. Flamm, “Quantitative assessment of left ventricular function: steady-state free precession mr imaging with or without sensitivity encoding,” *Radiology*, vol. 235, no. 3, pp. 1031–1035, Jun. 2005, doi: 10.1148/RADIOL.2353030995.

[37] G. B. Chavhan, P. S. Babyn, B. G. Jankharia, H. L. M. Cheng, and M. M. Shroff, “Steady-state MR imaging sequences: physics, classification, and clinical applications,” *Radiographics*, vol. 28, no. 4, pp. 1147–1160, Jul. 2008, doi: 10.1148/RG.284075031.

[38] D. T. Ginat *et al.*, “Cardiac Imaging: Part 1, MR pulse sequences, imaging planes, and basic anatomy,” *Am. J. Roentgenol.*, vol. 197, no. 4, pp. 808–815, 2011, doi: 10.2214/AJR.10.7231.

[39] L. R. R. Durán *et al.*, “Aplicaciones cardiovasculares empleando sistema convencional de Resonancia Magnética,” *An. Radiol. México*, vol. 2, no. 3, pp. 147–155, 2003.

[40] S. M. Forbat, M. A. Sakrana, K. H. Darasz, F. El-Demerdash, and S. R. Underwood, "Rapid assessment of left ventricular volume by short axis cine MRI," *Br. J. Radiol.*, vol. 69, no. 819, pp. 221–225, Feb. 2014, doi: 10.1259/0007-1285-69-819-221.

[41] M. Pérez-Pelegrí, J. V. Monmeneu, M. P. López-Lereu, and D. Moratal, "Convolutional neural networks for segmentation in short-axis cine cardiac magnetic resonance imaging: review and considerations," in *Convolutional Neural Networks For Medical Image Processing Applications.*, Ş. Öztürk, Ed. ROUTLEDGE, 2022, p. 274. Accessed: Sep. 12, 2022. [Online]. Available: https://www.routledge.com/Convolutional-Neural-Networks-for-Medical-Image-Processing-Applications/Ozturk/p/book/9781032104003.

[42] M. Z. Alom *et al.*, "A state-of-the-art survey on deep learning theory and architectures," *Electron.* , vol. 8, no. 3, p. 292, Mar. 2019, doi: 10.3390/ELECTRONICS8030292.

[43] F. Castro, "We're (data) hungry: the importance of Big Data for the subsea industry," *Abyssal*, Jun. 17, 2020. https://abyssal.eu/were-data-hungry/ (accessed Oct. 12, 2022).

[44] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, Dec. 1943, doi: 10.1007/BF02478259.

[45] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, Nov. 1958, doi: 10.1037/H0042519.

[46] S. Sharma, "What the hell is perceptron? ," *Towards Data Science*, Sep. 09, 2017. https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53 (accessed Oct. 12, 2022).

[47] S. Jadon, "Introduction to different activation functions for deep learning," *Medium*, Mar. 16, 2018. https://medium.com/@shrutijadon/survey-on-activation-functions-for-deep-learning-9689331ba092 (accessed Oct. 13, 2022).

[48] M. Minsky and S. A. Papert, *Perceptrons, reissue of the 1988 expanded edition with a new foreword by Léon Bottou*. The MIT Press, 2017. Accessed: Oct. 13, 2022. [Online]. Available: https://mitpress.mit.edu/9780262534772/perceptrons/

[49] M. M. Noel, A. L, A. Trivedi, and P. Dutta, "Growing cosine unit: a novel oscillatory activation function that can speedup training and reduce parameters in convolutional neural networks," Aug. 2021, doi: 10.48550/arxiv.2108.12943.

[50] T. Hastie, J. Friedman, and R. Tibshirani, *The Elements of Statistical Learning*.

New York, NY: Springer New York, 2001. doi: 10.1007/978-0-387-21606-5.

[51] M. Hosseinzadeh *et al.*, “A multiple multilayer perceptron neural network with an adaptive learning algorithm for thyroid disease diagnosis in the internet of medical things,” *J. Supercomput.*, vol. 77, no. 4, pp. 3616–3637, Apr. 2021, doi: 10.1007/S11227-020-03404-W/TABLES/5.

[52] D. E. Rumelhart and J. L. McClelland, “Learning internal representations by error propagation,” in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, MIT Press, 1987, pp. 318–362. Accessed: Oct. 13, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/6302929

[53] P. J. (Paul J. Werbos, *The roots of backpropagation : from ordered derivatives to neural networks and political forecasting*, 1st ed. Wiley-Interscience, 1994. Accessed: Oct. 13, 2022. [Online]. Available: https://www.wiley.com/en-us/The+Roots+of+Backpropagation%3A+From+Ordered+Derivatives+to+Neural+Networks+and+Political+Forecasting+-p-9780471598978

[54] Y. Le Cun and F. Fogelman-Soulié, “Modèles connexionnistes de l’apprentissage,” *Intellectica*, vol. 2, no. 1, pp. 114–143, 1987, doi: 10.3406/INTEL.1987.1804.

[55] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, Jun. 2010, pp. 807–814. Accessed: Oct. 13, 2022. [Online]. Available: https://icml.cc/Conferences/2010/papers/432.pdf

[56] D. H. Hubel and T. N. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex,” *J. Physiol.*, vol. 160, no. 1, p. 154, Jan. 1962, doi: 10.1113/JPHYSIOL.1962.SP006837.

[57] D. H. Hubel and T. N. Wiesel, “Receptive fields and functional architecture of monkey striate cortex,” *J. Physiol.*, vol. 195, no. 1, pp. 215–243, Mar. 1968, doi: 10.1113/JPHYSIOL.1968.SP008455.

[58] R. N. Bracewell, *The Fourier transform and its applications*, 3rd ed. Science, Engineering, & Math - McGraw Hill, 1999.

[59] M. Yani, B. Irawan, and C. Setiningsih, “Application of transfer learning using convolutional neural network method for early detection of Terry’s nail,” *J. Phys. Conf. Ser.*, vol. 1201, no. 1, p. 012052, May 2019, doi: 10.1088/1742-6596/1201/1/012052.

[60] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, Jun. 2017, doi: 10.1145/3065386.

[61] M. Z. Alom *et al.*, “The history began from AlexNet: a comprehensive survey on deep learning approaches,” Mar. 2018, doi: 10.48550/arxiv.1803.01164.

[62] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440, doi: 10.1109/CVPR.2015.7298965

[63] Y. Pang, J. Lin, T. Qin, and Z. Chen, “Image-to-Image translation: methods and applications,” *IEEE Trans. Multimed.*, vol. 24, pp. 3859–3881, 2022, doi: 10.1109/TMM.2021.3109419.

[64] K. Armanious *et al.*, “MedGAN: Medical image translation using GANs,” *Comput. Med. Imaging Graph.*, vol. 79, p. 101684, Jan. 2020, doi: 10.1016/J.COMPMEDIMAG.2019.101684.

[65] R. Alaguselvi and K. Murugan, “Image enhancement using convolutional neural networks,” in *2019 International Conference on Clean Energy and Energy Efficient Electronics Circuit for Sustainable Development, INCCES 2019*, Dec. 2019, pp. 1–5. doi: 10.1109/INCCES47820.2019.9167741.

[66] T. Qiu, C. Wen, K. Xie, F. Q. Wen, G. Q. Sheng, and X. G. Tang, “Efficient medical image enhancement based on CNN-FBB model,” *IET Image Process.*, vol. 13, no. 10, pp. 1736–1744, Aug. 2019, doi: 10.1049/IET-IPR.2018.6380.

[67] R. J. Chen, M. Y. Lu, T. Y. Chen, D. F. K. Williamson, and F. Mahmood, “Synthetic data in machine learning for medicine and healthcare,” *Nat. Biomed. Eng.* , vol. 5, no. 6, pp. 493–497, Jun. 2021, doi: 10.1038/s41551-021-00751-8.

[68] V. Thambawita *et al.*, “SinGAN-Seg: Synthetic training data generation for medical image segmentation,” *PLoS One*, vol. 17, no. 5, p. e0267976, May 2022, doi: 10.1371/JOURNAL.PONE.0267976.

[69] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[70] K. Cho *et al.*, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, Jun. 2014, pp. 1724–1734. doi: 10.3115/v1/d14-1179.

[71] A. van den Oord *et al.*, “WaveNet: a generative model for raw audio,” Sep. 2016, Accessed: Dec. 14, 2020. [Online]. Available: http://arxiv.org/abs/1609.03499

[72] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, “1D convolutional neural networks and applications: A survey,” *Mech. Syst. Signal*

Process., vol. 151, p. 107398, Apr. 2021, doi: 10.1016/J.YMSSP.2020.107398.

[73] A. Vaswani *et al.*, "Attention is all you need," in *31st International Conference on Neural Information Processing Systems (NIPS'17)*, Jun. 2017, vol. 2017-Decem, pp. 6000–6010. Accessed: Sep. 10, 2021. [Online]. Available: https://arxiv.org/abs/1706.03762v5

[74] S. Khan *et al.*, "Transformers in vision: a survey," *ACM Comput. Surv.*, vol. 54, no. 10s, pp. 1–41, Jan. 2022, doi: 10.1145/3505244.

[75] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2022, pp. 11966–11976. doi: 10.1109/CVPR52688.2022.01167.

[76] H. Wu *et al.*, "CvT: Introducing convolutions to vision transformers," in *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 22–31. doi: 10.1109/ICCV48922.2021.00009.

[77] M. B. Afifi, A. Abdelrazek, N. A. Deiab, A. I. Abd El-Hafez, and A. H. El-Farrash, "The effects of CT x-ray tube voltage and current variations on the relative electron density (RED) and CT number conversion curves," *J. Radiat. Res. Appl. Sci.*, vol. 13, no. 1, pp. 1–11, Jan. 2020, doi: 10.1080/16878507.2019.1693176.

[78] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, May 2010, pp. 249–256. Accessed: Oct. 13, 2022. [Online]. Available: https://proceedings.mlr.press/v9/glorot10a.html

[79] J. C. Reinhold, B. E. Dewey, A. Carass, and J. L. Prince, "Evaluating the impact of intensity normalization on MR image synthesis," in *Proc. SPIE 10949, Medical Imaging 2019: Image Processing*, Mar. 2019, vol. 10949, p. 126. doi: 10.1117/12.2513089.

[80] N. Jacobsen, A. Deistung, D. Timmann, S. L. Goericke, J. R. Reichenbach, and D. Güllmar, "Analysis of intensity normalization for optimal segmentation performance of a fully convolutional neural network," *Z. Med. Phys.*, vol. 29, no. 2, pp. 128–138, May 2019, doi: 10.1016/J.ZEMEDI.2018.11.004.

[81] "DICOM PS3.3 2022d - Information Object Definitions," *NEMA*. https://dicom.nema.org/medical/dicom/current/output/html/part03.html (accessed Oct. 13, 2022).

[82] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnU-Net: a self-configuring method for deep learning-based biomedical image

segmentation," *Nat. Methods* , vol. 18, no. 2, pp. 203–211, Dec. 2020, doi: 10.1038/s41592-020-01008-z.

[83] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015, doi: 10.1109/TPAMI.2015.2389824.

[84] E. Abdelmaguid *et al.*, "Left ventricle segmentation and volume estimation on cardiac MRI using deep learning," *arXiv Comput. Vis. Pattern Recognit.*, Sep. 2018, Accessed: Sep. 23, 2020. [Online]. Available: http://arxiv.org/abs/1809.06247

[85] S. Bubeck and M. Sellke, "A universal law of robustness via isoperimetry," in *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, Dec. 2021, vol. 34, pp. 28811–28822.

[86] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Express*, vol. 6, no. 4, pp. 312–315, Dec. 2020, doi: 10.1016/j.icte.2020.04.010.

[87] E. Hoffer, I. Hubara, and D. Soudry, "Train longer, generalize better: closing the generalization gap in large batch training of neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 2017-December, pp. 1732–1742, May 2017, Accessed: May 31, 2021. [Online]. Available: http://arxiv.org/abs/1705.08741

[88] D. M. Hawkins, "The problem of overfitting," *J. Chem. Inf. Comput. Sci.*, vol. 44, no. 1, pp. 1–12, Jan. 2004, doi: 10.1021/ci0342472.

[89] T. B. Brown *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020, vol. 33, pp. 1877-1901, doi: https://doi.org/10.48550/arXiv.2005.14165. Avialable: https://arxiv.org/abs/2005.14165

[90] W. Sarle, "Stopped training and other remedies for overfitting ," in *Proceedings of the twenty-seventh symposium on the interface of computing science and statistics*, 1995, pp. 352–360.

[91] X. Ying, "An overview of overfitting and its solutions," *J. Phys. Conf. Ser.*, vol. 1168, no. 2, pp. 022022, Mar. 2019, doi: 10.1088/1742-6596/1168/2/022022.

[92] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *32nd International Conference on Machine Learning, ICML 2015*, Feb. 2015, vol. 1, pp. 448–456. Accessed: Sep. 23, 2020. [Online]. Available: https://arxiv.org/abs/1502.03167v3

[93] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: representing

model uncertainty in deep learning," in *International conference on machine learning*, 2016, pp. 1050–1059. Accessed: May 31, 2021. [Online]. Available: http://proceedings.mlr.press/v48/gal16.pdf

[94] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "DropBlock: a regularization method for convolutional networks," in *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, Oct. 2018, vol. 31. Accessed: May 31, 2021. [Online]. Available: http://arxiv.org/abs/1810.12890

[95] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using Convolutional Networks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Oct. 2015, vol. 07-12-June-2015, pp. 648–656. doi: 10.1109/CVPR.2015.7298664.

[96] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," Dec. 2014. Accessed: May 31, 2021. [Online]. Available: https://arxiv.org/abs/1312.6114v10

[97] Y. Pu *et al.*, "Variational autoencoder for deep learning of images, labels and captions," in *Advances in Neural Information Processing Systems*, Sep. 2016, pp. 2360–2368. Accessed: May 31, 2021. [Online]. Available: http://arxiv.org/abs/1609.08976

[98] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems*, Aug. 2014, pp. 2672–2680. Accessed: May 31, 2021. [Online]. Available: http://arxiv.org/abs/1308.4214

[99] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020, vol. 33, pp. 6840–6851. Accessed: Oct. 13, 2022. [Online]. Available: https://github.com/hojonathanho/diffusion.

[100] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification," *Neurocomputing*, vol. 321, pp. 321–331, Dec. 2018, doi: 10.1016/j.neucom.2018.09.013.

[101] Z. Chen, J. Wang, H. He, and X. Huang, "A fast deep learning system using GPU," in *Proceedings - IEEE International Symposium on Circuits and Systems*, 2014, pp. 1552–1555. doi: 10.1109/ISCAS.2014.6865444.

[102] A. Kayid and Y. Khaled, "Performance of CPUs/GPUs for deep learning workloads,", Media Engineering and Technology Faculty German University in Cairo, May 2018. doi: 10.13140/RG.2.2.22603.54563.

[103] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for

biomedical image segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9351, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[104] F. Yu and V. Koltun, "Multi-Scale context aggregation by dilated convolutions," *4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc.*, Nov. 2015, Accessed: Dec. 14, 2020. [Online]. Available: http://arxiv.org/abs/1511.07122

[105] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv*, Jun. 2017, Accessed: Dec. 14, 2020. [Online]. Available: http://arxiv.org/abs/1706.05587

[106] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448. doi: 10.1109/ICCV.2015.169.

[107] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, Dec. 2017, vol. 2017-October, pp. 2980–2988. doi: 10.1109/ICCV.2017.322.

[108] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," Mar. 2016, Accessed: May 30, 2021. [Online]. Available: http://arxiv.org/abs/1603.07285

[109] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-net: learning dense volumetric segmentation from sparse annotation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Oct. 2016, vol. 9901 LNCS, pp. 424–432. doi: 10.1007/978-3-319-46723-8_49.

[110] F. Milletari, N. Navab, and S. A. Ahmadi, "V-Net: fully convolutional neural networks for volumetric medical image segmentation," in *Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016*, Dec. 2016, pp. 565–571. doi: 10.1109/3DV.2016.79.

[111] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. Accessed: May 30, 2021. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[112] O. Oktay *et al.*, "Attention U-Net: learning where to look for the pancreas," Apr. 2018. Accessed: May 30, 2021. [Online]. Available: http://arxiv.org/abs/1804.03999

[113] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," Sep. 2015. Accessed: May 30, 2021. [Online].

Available: https://arxiv.org/abs/1409.0473v7

[114] K. Xu *et al.*, “Show, attend and tell: neural image caption generation with visual attention,” in *International conference on machine learning*, Jun. 2015, pp. 2048–2057. Accessed: May 30, 2021. [Online]. Available: http://proceedings.mlr.press/v37/xuc15.html

[115] K. H. Zou *et al.*, “Statistical validation of image segmentation quality based on a spatial overlap index,” *Acad. Radiol.*, vol. 11, no. 2, pp. 178–189, Feb. 2004, doi: 10.1016/S1076-6332(03)00671-8.

[116] Q. Tong *et al.*, “RIANet: Recurrent interleaved attention network for cardiac MRI segmentation,” *Comput. Biol. Med.*, vol. 109, pp. 290–302, Jun. 2019, doi: 10.1016/j.compbiomed.2019.04.042.

[117] R. P. K. Poudel, P. Lamata, and G. Montana, “Recurrent fully convolutional neural networks for multi-slice MRI cardiac segmentation,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Oct. 2017, vol. 10129 LNCS, pp. 83–94. doi: 10.1007/978-3-319-52280-7_8.

[118] O. Bernard *et al.*, “Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: is the problem solved?,” *IEEE Trans. Med. Imaging*, vol. 37, no. 11, pp. 2514–2525, Nov. 2018, doi: 10.1109/TMI.2018.2837502.

[119] Q. Tao *et al.*, “Deep Learning–based method for fully automatic quantification of left ventricle function from cine mr images: a multivendor, multicenter study,” *Radiology*, vol. 290, no. 1, pp. 81–88, Jan. 2019, doi: 10.1148/radiol.2018180513.

[120] C. Chen *et al.*, “Improving the generalizability of convolutional neural network-based segmentation on CMR images,” *Front. Cardiovasc. Med.*, vol. 7, p. 105, Jun. 2020, doi: 10.3389/fcvm.2020.00105.

[121] P. Rajiah, N. L. Fulton, and M. Bolen, “Magnetic resonance imaging of the papillary muscles of the left ventricle: normal anatomy, variants, and abnormalities,” *Insights Imaging*, vol. 10, no. 1, pp. 1–17, Dec. 2019, doi: 10.1186/s13244-019-0761-3.

[122] A. Scatteia *et al.*, “Abnormal papillary muscle signal on cine MRI as a typical feature of mitral valve prolapse,” *Sci. Rep.*, vol. 10, no. 1, pp. 1–7, Dec. 2020, doi: 10.1038/s41598-020-65983-1.

[123] A. Bartoli *et al.*, “Deep learning–based automated segmentation of left ventricular trabeculations and myocardium on cardiac MR images: A Feasibility Study,” *Radiol. Artif. Intell.*, vol. 3, no. 1, pp. e200021, Nov. 2021, doi:

10.1148/ryai.2020200021.

[124] M. Perez-Pelegri *et al.*, "PSPU-Net for automatic short axis cine mri segmentation of left and right ventricles," in *2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE)*, Dec. 2020, pp. 1048–1053. doi: 10.1109/bibe50027.2020.00177.

[125] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2881–2890. Accessed: May 30, 2021. [Online]. Available: https://github.com/hszhao/PSPNet

[126] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," Jul. 2015. doi: 10.48550/arxiv.1412.6980.

[127] A. H. Curiale, F. D. Colavecchia, and G. Mato, "Automatic quantification of the LV function and mass: a deep learning approach for cardiovascular MRI," *Comput. Methods Programs Biomed.*, vol. 169, pp. 37–50, Feb. 2019, doi: 10.1016/J.CMPB.2018.12.002.

[128] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, "Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017, vol. 10553 LNCS, pp. 240–248. doi: 10.1007/978-3-319-67558-9_28.

[129] W. R. Crum, O. Camara, and D. L. G. Hill, "Generalized overlap measures for evaluation and validation in medical image analysis," *IEEE Trans. Med. Imaging*, vol. 25, no. 11, pp. 1451–1461, Nov. 2006, doi: 10.1109/TMI.2006.880587.

[130] C. Hsin and C. Danner, "Convolutional neural networks for left ventricle volume estimation," Stanford University, 2016. Accessed: Dec. 14, 2020. [Online]. Available: http://cs231n.stanford.edu/reports/2016/pdfs/305_Report.pdf

[131] C. Cano-Espinosa, G. Gonzalez, G. R. Washko, M. Cazorla, and R. S. J. Estepar, "Biomarker localization from deep learning regression networks," *IEEE Trans. Med. Imaging*, vol. 39, no. 6, pp. 2121–2132, Jun. 2020, doi: 10.1109/TMI.2020.2965486.

[132] T. Miller, "Explanation in artificial intelligence: insights from the social sciences," *Artif. Intell.*, vol. 267, pp. 1–38, Feb. 2019, doi: 10.1016/J.ARTINT.2018.07.007.

[133] P. Fischer, A. Dosovitskiy, and T. Brox, "Image orientation estimation with convolutional networks," in *Pattern Recognition*, Springer , 2015, pp. 368–378. doi: 10.1007/978-3-319-24947-6_30/FIGURES/7.

[134] S. Mahendran, H. Ali, and R. Vidal, “3D pose regression using convolutional neural networks,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Aug. 2017, vol. 2017-July, pp. 494–495. doi: 10.1109/CVPRW.2017.73.

[135] J. H. Cole *et al.*, “Predicting brain age with deep learning from raw imaging data results in a reliable and heritable biomarker,” *Neuroimage*, vol. 163, pp. 115–124, 2017, doi: 10.1016/j.neuroimage.2017.07.059.

[136] G. Gonzalez Serrano, G. R. Washko, and R. San José Estépar, “Deep learning for biomarker regression: application to osteoporosis and emphysema on chest CT scans,” in *Medical Imaging 2018: Image Processing*, Mar. 2018, vol. 10574, p. 52. doi: 10.1117/12.2293455.

[137] J. P. Vigueras-Guillen, J. Van Rooij, H. G. Lemij, K. A. Vermeer, and L. J. Van Vliet, “Convolutional neural network-based regression for biomarker estimation in corneal endothelium microscopy images,” in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, Jul. 2019, pp. 876–881. doi: 10.1109/EMBC.2019.8857201.

[138] G. González, G. R. Washko, R. S. J. Estépar, M. Cazorla, and C. Cano Espinosa, “Automated Agatston score computation in non-ECG gated CT scans using deep learning,” in *Medical Imaging 2018: Image Processing*, Mar. 2018, vol. 10574, p. 91. doi: 10.1117/12.2293681.

[139] G. Luo, G. Sun, K. Wang, S. Dong, and H. Zhang, “A novel left ventricular volumes prediction method based on deep learning network in cardiac MRI,” in *2016 Computing in Cardiology Conference (CinC)*, 2016, pp. 89–92. Accessed: Nov. 06, 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7868686

[140] F. Zhu, “Estimating left ventricular volume with ROI-based convolutional neural network,” *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 26, no. 1, pp. 23–34, Jan. 2018, doi: 10.3906/elk-1704-335.

[141] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2921–2929. Accessed: Sep. 23, 2020. [Online]. Available: http://cnnlocalization.csail.mit.edu

[142] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: visual explanations from deep networks via gradient-based localization,” in *Proceedings of the IEEE International Conference on Computer Vision*, Dec. 2017, vol. 2017-October, pp. 618–626. doi: 10.1109/ICCV.2017.74.

[143] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, “SmoothGrad: removing noise by adding noise,” Jun. 2017, doi: 10.48550/arxiv.1706.03825.

[144] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘Why should I trust you?’ explaining the predictions of any classifier,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, vol. 13-17-August-2016, pp. 1135–1144. doi: 10.1145/2939672.2939778.

[145] S. M. Lundberg, P. G. Allen, and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017. Accessed: Oct. 13, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

[146] S. Pereira, R. Meier, V. Alves, M. Reyes, and C. A. Silva, “Automatic brain tumor grading from MRI data using convolutional neural networks and quality assessment,” in *Lecture Notes in Computer Science*, vol. 11038, Springer Verlag, 2018, pp. 106–114. doi: 10.1007/978-3-030-02628-8_12/FIGURES/3.

[147] A. Singh, S. Sengupta, and V. Lakshminarayanan, “Explainable deep learning models in medical image analysis,” *J. Imaging* , vol. 6, no. 6, p. 52, Jun. 2020, doi: 10.3390/JIMAGING6060052.

[148] F. Eitel and K. Ritter, “Testing the robustness of attribution methods for convolutional neural networks in MRI-based Alzheimer’s disease classification,” in *Lecture Notes in Computer Science*, vol. 11797, Springer, 2019, pp. 3–11. doi: 10.1007/978-3-030-33850-3_1/FIGURES/3.

[149] Z. Papanastasopoulos *et al.*, “Explainable AI for medical imaging: deep-learning CNN ensemble for classification of estrogen receptor status from breast MRI,” in *Medical Imaging 2020: Computer-Aided Diagnosis*, Mar. 2020, vol. 11314, pp. 228–235. doi: 10.1117/12.2549298.

[150] L. Wang, Z. Q. Lin, and A. Wong, “COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images,” *Sci. Rep.*, vol. 10, no. 1, pp. 1–12, Nov. 2020, doi: 10.1038/s41598-020-76550-z.

[151] Z. H. Zhou, “A brief introduction to weakly supervised learning,” *Natl. Sci. Rev.*, vol. 5, no. 1, pp. 44–53, Jan. 2018, doi: 10.1093/NSR/NWX106.

[152] L. Chan, M. S. Hosseini, and K. N. Plataniotis, “A comprehensive analysis of weakly-supervised semantic segmentation in different image domains,” *Int. J. Comput. Vis.*, pp. 1–24, Dec. 2019, doi: 10.1007/s11263-020-01373-4.

[153] J. Peng and Y. Wang, “Medical image segmentation with limited supervision: a review of deep network models,” *IEEE Access*, vol. 9, pp. 36827–36851, 2021,

doi: 10.1109/ACCESS.2021.3062380.

[154] I. Laradji *et al.*, “A weakly supervised consistency-based learning method for COVID-19 segmentation in CT images,” in *Proceedings - 2021 IEEE Winter Conference on Applications of Computer Vision, WACV 2021*, Jan. 2021, pp. 2452–2461. doi: 10.1109/WACV48630.2021.00250.

[155] H. R. Roth *et al.*, “Going to extremes: weakly supervised medical image segmentation,” *Mach. Learn. Knowl. Extr.* , vol. 3, no. 2, pp. 507–524, Jun. 2021, doi: 10.3390/MAKE3020026.

[156] S. Wang, W. Chen, S. M. Xie, G. Azzari, and D. B. Lobell, “Weakly supervised deep learning for segmentation of remote sensing imagery,” *Remote Sens.*, vol. 12, no. 2, p. 207, Jan. 2020, doi: 10.3390/rs12020207.

[157] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, “Weakly-supervised semantic segmentation network with deep seeded region growing,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7014–7023.

[158] G. Yang *et al.*, “Weakly-supervised convolutional neural networks of renal tumor segmentation in abdominal CTA images,” *BMC Med. Imaging*, vol. 20, no. 1, p. 37, Apr. 2020, doi: 10.1186/S12880-020-00435-W.

[159] C.-C. Hsu, K.-J. Hsu, C.-C. Tsai, Y.-Y. Lin, and Y.-Y. Chuang, “Weakly supervised instance segmentation using the bounding box tightness prior,” in *Advances in Neural Information Processing Systems 32*, 2019, pp. 6586–6597. Accessed: Sep. 23, 2020. [Online]. Available: https://github.com/chengchunhsu/WSIS_BBTP.

[160] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele, “Simple does it: weakly supervised instance and semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 876–885, doi: 10.1109/CVPR.2017.181.

[161] K. B. Girum, G. Créhange, R. Hussain, and A. Lalande, “Fast interactive medical image segmentation with weakly supervised deep learning method,” *Int. J. Comput. Assist. Radiol. Surg.*, vol. 15, no. 9, pp. 1437–1444, Sep. 2020, doi: 10.1007/s11548-020-02223-x.

[162] P. Zhang, Y. Zhong, Y. Deng, X. Tang, and X. Li, “A Survey on deep learning of small sample in biomedical image analysis,” *arXiv Prepr. arXiv1908.00473*, Aug. 2019, Accessed: Sep. 23, 2020. [Online]. Available: http://arxiv.org/abs/1908.00473

[163] M. Pérez-Pelegrí *et al.*, “Automatic left ventricle volume calculation with explainability through a deep learning weak-supervision methodology,” *Comput.*

Methods Programs Biomed., vol. 208, pp. 106275 Jul. 2021, doi: 10.1016/J.CMPB.2021.106275.

[164] C. Moreira, R. Sindhgatta, C. Ouyang, P. Bruza, and A. Wichert, "An Investigation of interpretability techniques for deep learning in predictive process analytics," Feb. 2020, Accessed: Sep. 23, 2020. [Online]. Available: http://arxiv.org/abs/2002.09192

[165] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Müller, "Toward interpretable machine learning: transparent deep neural networks and beyond," Mar. 2020, Accessed: Sep. 23, 2020. [Online]. Available: http://arxiv.org/abs/2003.07631

[166] M. Ghassemi, L. Oakden-Rayner, and A. L. Beam, "The false hope of current approaches to explainable artificial intelligence in health care," *Lancet Digit. Heal.*, vol. 3, no. 11, pp. e745–e750, Nov. 2021, doi: 10.1016/S2589-7500(21)00208-9.

[167] "Artifical Intelligence in EU medical device legislation," 2020. Accessed: Nov. 09, 2020. [Online]. Available: https://www.cocir.org/fileadmin/Position_Papers_2020/COCIR_Analysis_on_AI_in_medical_Device_Legislation_-_Sept._2020_-_Final_2.pdf

[168] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018, doi: 10.1109/TPAMI.2017.2699184.

[169] J. Du, L. Wang, Y. Liu, Z. Zhou, Z. He, and Y. Jia, "Brain MRI super-resolution using 3D dilated convolutional encoder-decoder network," *IEEE Access*, vol. 8, pp. 18938–18950, 2020, doi: 10.1109/ACCESS.2020.2968395.

[170] J. Brownlee, *Probability for Machine Learning: Discover How To Harness Uncertainty With Python*. Machine Learning Mastery, 2019. Accessed: Oct. 14, 2022. [Online]. Available: https://books.google.es/books?id=uU2xDwAAQBAJ

[171] C. R. Dominguez *et al.*, "Classification of segmental wall motion in echocardiography using quantified parametric images," in *International Workshop on Functional Imaging and Modeling of the Heart*, 2005, vol. 3504, pp. 477–486. doi: 10.1007/11494621_47.

[172] P. Gifani, H. Behnam, A. Shalbaf, and Z. A. Sani, "Automatic detection of end-diastole and end-systole from echocardiography images using manifold learning," *Physiol. Meas.*, vol. 31, no. 9, pp. 1091–1103, Sep. 2010, doi: 10.1088/0967-3334/31/9/002.

[173] A. Shalbaf, H. Behnam, P. Gifani, and Z. Alizadeh-Sani, "Automatic detection of end systole and end diastole within a sequence of 2-D echocardiographic images using modified Isomap algorithm," in *2011 1st Middle East Conference on Biomedical Engineering, MECBME 2011*, 2011, pp. 217–220. doi: 10.1109/MECBME.2011.5752104.

[174] M. Zolgharni *et al.*, "Automatic detection of end-diastolic and end-systolic frames in 2D echocardiography," *Echocardiography*, vol. 34, no. 7, pp. 956–967, Jul. 2017, doi: 10.1111/echo.13587.

[175] A. Meidellfiorito, A. Ostvik, E. Smistad, S. Leclerc, O. Bernard, and L. Lovstakken, "Detection of cardiac events in echocardiography using 3D convolutional recurrent neural networks," in *IEEE International Ultrasonics Symposium, IUS*, 2018, pp. 1–4. doi: 10.1109/ULTSYM.2018.8580137.

[176] C. Ciusdel *et al.*, "Deep neural networks for ECG-free cardiac phase and end-diastolic frame detection on coronary angiographies," *Comput. Med. Imaging Graph.*, vol. 84, p. 101749, Sep. 2020, doi: 10.1016/J.COMPMEDIMAG.2020.101749.

[177] B. Kong, Y. Zhan, M. Shin, T. Denny, and S. Zhang, "Recognizing end-diastole and end-systole frames via deep temporal regression network," in *International Conference on Medical Image Computing and Computer-Assisted Intervention-MICCAI 2016*, Oct. 2016, vol. 9902 LNCS, pp. 264–272. doi: 10.1007/978-3-319-46726-9_31.

[178] F. Yang, Y. He, M. Hussain, H. Xie, and P. Lei, "Convolutional neural network for the detection of end-diastole and end-systole frames in free-breathing cardiac magnetic resonance imaging," *Comput. Math. Methods Med.*, vol. 2017, 2017, doi: 10.1155/2017/1640835.

[179] J. Krebs, H. Delingette, N. Ayache, and T. Mansi, "Learning a generative motion model from image sequences based on a latent motion matrix," *IEEE Trans. Med. Imaging*, vol. 40, no. 5, pp. 1405–1416, May 2021, doi: 10.1109/TMI.2021.3056531.

[180] M. Pérez-Pelegrí, J. V. Monmeneu, M. P. López-Lereu, A. M. Maceira, V. Bodi, and D. Moratal, "End-systole and end-diastole detection in short axis cine MRI using a fully convolutional neural network with dilated convolutions," *Comput. Med. Imaging Graph.*, vol. 99, p. 102085, Jul. 2022, doi: 10.1016/J.COMPMEDIMAG.2022.102085.

[181] M. Roald, "Detecting valvular event times from echocardiograms using deep neural networks," University of Oslo, 2018. Accessed: Sep. 10, 2021. [Online]. Available: https://www.duo.uio.no/handle/10852/61922

[182] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, and J. Li, “Dice loss for data-imbalanced NLP Tasks,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 465–476, Nov. 2019, Accessed: Jan. 26, 2021. [Online]. Available: http://arxiv.org/abs/1911.02855

[183] Y. Yu, X. Si, C. Hu, and J. Zhang, “A review of recurrent neural networks: lstm cells and network architectures,” *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019, doi: 10.1162/NECO_A_01199.

[184] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *International conference on machine learning* , 2013, pp. 1310–1318. Accessed: Feb. 18, 2021. [Online]. Available: http://arxiv.org/abs/1211.5063

[185] L. Hou, J. Zhu, J. T. Kwok, F. Gao, T. Qin, and T.-Y. Liu, “Normalization helps training of quantized LSTM,” in *Neural Information Processing Systems (NeurIPS 2019)*, 2019, pp. 7346–7356.

[186] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Instance normalization: the missing ingredient for fast stylization,” Jul. 2016, doi: 10.48550/arxiv.1607.08022.

[187] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” Jul. 2016, doi: 10.48550/arxiv.1607.06450.

[188] P. Zhou, J. Feng, C. Ma, C. Xiong, S. C. H. Hoi, and W. E, “Towards theoretically understanding why sgd generalizes better than adam in deep learning,” in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020, vol. 33, pp. 21285–21296. Accessed: Oct. 14, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/f3f27a324736617f20abbf2ffd806f6d-Abstract.html

[189] I. Loshchilov and F. Hutter, “Fixing weight decay regularization in adam,” Feb. 2019. Accessed: Oct. 14, 2022. [Online]. Available: https://openreview.net/forum?id=rk6qdGgCZ

[190] D. Hendrycks and K. Gimpel, “Gaussian Error Linear Units (GELUs),” Jun. 2016, doi: 10.48550/arxiv.1606.08415.

[191] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” May 2018, doi: 10.48550/arxiv.1810.04805.

[192] J. Jumper *et al.*, “Highly accurate protein structure prediction with AlphaFold,” *Nat.* , vol. 596, pp. 583–589, Jul. 2021, doi: 10.1038/s41586-021-03819-2.

[193] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, “Generalizing from a few examples: a survey on few-shot learning,” *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, Jun. 2020, doi: 10.1145/3386252

Chapter 10.
Publications

10.1. Publications derived from this PhD Thesis

This section lists all the publications directly related to the context and results of this thesis.

Chapter 5. Automatic semantic segmentation

International conferences:

1) M. Perez-Pelegri, J. V Monmeneu, M. P. López-Lereu, S. Ruiz-España, I. Del-Canto, V. Bodí and D. Moratal, "PSPU-Net for automatic short axis cine mri segmentation of left and right ventricles," in *2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE)*, Dec. 2020, pp. 1048–1053. doi: https://doi.org/10.1109/BIBE50027.2020.00177.

Book chapters:

1) M. Pérez-Pelegrí, J. V. Monmeneu, M. P. López-Lereu, and D. Moratal, "Convolutional neural networks for segmentation in short-axis cine cardiac magnetic resonance imaging: review and considerations," in *Convolutional Neural Networks For Medical Image Processing Applications.*, Ş. Öztürk, ed. CRC Press, 2022. doi: https://doi.org/10.1201/9781003215141.

Chapter 6. Automatic biomarker estimation and explainability

Journal papers:

1) M. Pérez-Pelegrí, J. V. Monmeneu, M. P. López-Lereu, L. Pérez-Pelegrí, A. M. Maceira, V. Bodí and D. Moratal, “Automatic left ventricle volume calculation with explainability through a deep learning weak-supervision methodology,” *Comput. Methods Programs Biomed.*, vol. 208, pp. 106275 Jul. 2021, doi: https://doi.org/10.1016/j.cmpb.2021.106275.

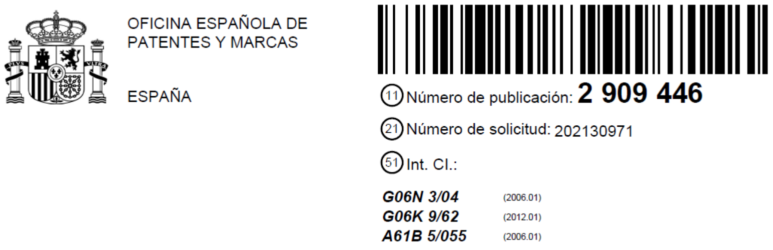
Chapter 7. Automatic End-Systole and End-Diastole detection

Journal papers:

1) M. Pérez-Pelegrí, J. V. Monmeneu, M. P. López-Lereu, A. M. Maceira, V. Bodi, and D. Moratal, “End-systole and end-diastole detection in short axis cine MRI using a fully convolutional neural network with dilated convolutions,” *Comput. Med. Imaging Graph.*, vol. 99, p. 102085, Jul. 2022, doi: https://doi.org/10.1016/j.compmedimag.2022.102085.

Patents:

1) Moratal Pérez, D; Pérez Pelegrí, M.; Monmeneu Menadas, JV; López Lereu, MP; Santabárbara Gómez, JM; Maceira Gonzalez, A M. (2022). Método de detección automática de sístole y diástole, Oficinal Española de Patentes y Marcas*.



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 909 446**

(21) Número de solicitud: 202130971

(51) Int. Cl.:

G06N 3/04	(2006.01)
G06K 9/62	(2012.01)
A61B 5/055	(2006.01)

*Accepted but pending for final approval and publication

10.2. Other publications

In this section, additional contributions published in the context of other projects related to the topics involved in this thesis but not directly applied are presented:

International conferences:

1) J. M. Jaén-Lorites, M. Pérez-Pelegrí, V. Laparra, and D. Moratal, "Understanding the style transfer deep-learning technique through a web dashboard," in *INTED 2022 Proceedings*, Mar. 2022, pp. 8914–8919. IATED. doi: https://doi.org/10.21125/INTED.2022.2324.

2) J. M. Jaen-Lorites, M. Pérez-Pelegrí, V. Laparra, and D. Moratal, "Synthetic generation of cardiac MR images combining convolutional variational autoencoders and style transfer," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, Jul. 2022, vol. 2022, pp. 2084–2087. IEEE. doi: https://doi.org/10.1109/EMBC48229.2022.9871135.

Supervised Bachelor's Degree final projects:

1) Miró Mezquita, L. (2021). (Directores: Moratal Pérez D., Del Canto I., Pérez Pelegrí M.). *Desarrollo de un modelo predictivo basado en redes neuronales convolucionales para el diagnóstico automático de la miocardiopatía hipertrófica y de la amiloidosis a partir del análisis de imágenes de resonancia magnética cardíaca* (Bachelor's Degree dissertation, Universitat Politècnica de València).
2) Andión García, C. D. L. O. (2021). (Directores: Moratal Pérez D., Del Canto I., Pérez Pelegrí M.). *Desarrollo de un método de diagnóstico automático de la miocardiopatía hipertensiva y la amiloidosis a partir del análisis de imágenes de resonancia magnética cardíaca mediante redes neuronales convolucionales* (Bachelor's Degree dissertation, Universitat Politècnica de València).
3) Pérez Herrero, S. (2021). (Directores: Moratal Pérez D., Pérez Pelegrí M.). *Desarrollo de un modelo para la predicción de la evolución de la extensión del infarto cerebral mediante técnicas de inteligencia artificial y el análisis de*

imágenes médicas (Bachelor's Degree dissertation, Universitat Politècnica de València).

4) Pérez Martínez, S. (2022). (Directores: Moratal Pérez D., Fernández Cisnal A., Pérez Pelegrí M.). *Desarrollo de un modelo predictivo de revascularización de oclusiones totales crónicas basado en imágenes de angiografía y métodos de aprendizaje profundo* (Bachelor's Degree dissertation, Universitat Politècnica de València).
5) Sastre García, B. (2022). (Directores: Moratal Pérez D., Romero Martín J.A., Pérez Pelegrí M.). *Diseño y validación de un flujo de trabajo para la segmentación automática de telerradiografías de extremidades inferiores mediante un modelo de arquitectura U-net y posterior automatización de sus mediciones* (Bachelor's Degree dissertation, Universitat Politècnica de València).

Supervised Master's thesis:

1) Jaén Lorites, JM. (2022) (Laparra Pérez Muelas V., Moratal Pérez D., Pérez Pelegrí M.). Generación de imágenes sintéticas de resonancia magnética cardíaca mediante técnicas de aprendizaje profundo (Master's dissertarion, Universitat de València).