

Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research
University A. Mira of Bejaïa
Faculty of Exact Sciences
Computer science Department



MASTER THESIS

Option : Artificial Intelligence

Efficient Convolutional Neural Network for 2D Echocardiographic images segmentation: Application on CAMUS DataSet

Realised by

M. DJOUAD Mohand M. DADA Idriss

in front of the Jury composed of

President :	M ^s ALOUI Soraya	Dr	University of Bejaia
Examinater :	M. AITMATEN Zahir	Dr	University of Bejaia
Supervisor :	M. BELAID Ahror	Pr	University of Bejaia

Acknowledgements

First and foremost, we thank Allah Who helped us accomplish this work, and Who has been with us at all times of our lives.

We especially thank our mentor, Dr BELAID Ahror, who gave us an excellent opportunity to carry out this work, which allowed us to acquire a lot of knowledge and experience in the field of Machine Learning and Deep Learning. His patience, his motivation and his deep knowledge always guide us.

We would also like to thank all our professors and our comrades at the University of Bejaia.

A big thank you to our friends in particular "Lamine" and "Floki" and to the Sawtech team for the crazy times we shared together. Finally to our families for always being there with us every step of the way.

Dedications

To my dear parents, for all their sacrifices, their love, their tenderness, their support and their
prayers throughout my studies,
To all my family especially my grandmother, and my friends for their support throughout my
academic career,
Finally to my little flower that I will make her really mine someday, thanks for accepting the job
to be with me,
May this work be the fulfillment of your so-called wishes and escape your unwavering support.

M. DJOUAD Mohand

I want to thank my parents especially for their support as well as my friends for always being
there for me.

M. DADA Idriss

Contents

Contents	i
List of Figures	v
List of Tables	vi
List of Abbreviations	viii
Introduction	1
1 Image Processing	2
1.1 Introduction	2
1.2 Image	2
1.3 Image formats	3
1.4 Image Data Types	4
1.5 Semantic segmentation	6
1.6 Medical Images	6
1.6.1 Overview	6
1.6.2 Medical Ultrasound	8
1.6.3 Echocardiography	9
1.7 Conclusion	9
2 Machine learning and Deep Learning	10
2.1 Introduction	10
2.2 Machine Learning	10
2.2.1 Types of Machine Learning Systems	10
2.3 Supervised/Unsupervised Learning	11
2.3.1 Supervised learning	11
2.3.2 Unsupervised learning	11
2.3.3 Semi-supervised learning	12
2.3.4 Reinforcement Learning	13
2.4 Neural networks	13
2.4.1 Definition	14
2.4.2 Relation to Biology	15

2.4.3	Artificial Models	15
2.4.4	Activation function	16
2.4.5	Error Function	16
2.4.6	Backpropagation	17
2.4.7	Hyperparameters	18
2.4.8	Overfitting and Underfitting in Neural Networks	18
2.5	Deep Neural Network	18
2.5.1	Major Architectures of Deep Networks	18
2.5.2	Convolutional Neural Networks (CNNs)	19
2.5.3	Deep Learning Applications in Healthcare	21
2.6	Conclusion	22
3	Dataset and Related Work	23
3.1	Introduction	23
3.2	Cardiac function	23
3.2.1	Anatomical structure of the heart	23
3.2.2	Heartbeat	24
3.2.3	Blood flow	25
3.3	CAMUS Dataset	25
3.4	Our approach	26
3.4.1	Model	26
3.4.2	U-NET architecture	27
3.5	EfficientNet architecture	28
3.5.1	Depthwise Separable Convolution	28
3.5.2	Inverse Res	28
3.6	Model specifications	28
3.6.1	Processing part	28
3.6.2	Training part	29
3.6.3	Training results	29
3.7	Conclusion	29
4	Experimental Results	30
4.1	Introduction	30
4.2	Medical image segmentation metrics	30
4.2.1	Region overlap	30
4.2.2	Spatial distances between contours	31
4.3	Results	31
4.4	Conclusion	35
Tools and software	36	
4.5	Tensorflow	36
4.6	SimpleITK	36
4.7	ITK-SNAP	37
4.8	Google Colab	37
4.9	Scikit-learn	38
4.10	Keras	38

Conclusion	40
Bibliographie	41

List of Figures

1.1	The letter “I” placed on a piece of graph paper. Pixels are accessed by their (x;y)-coordinates, where we go x columns to the right and y rows down [4]	2
1.2	Binary image[14]	4
1.3	8-bits grayscale image[15]	5
1.4	8-bits RGB image	5
1.5	TIFF Floating point image (Copyright: Tim Lukins, UoE)	6
1.6	Example of semantic segmentation [11]	7
1.7	Orthogonal planes of a 3 dimensional sonographic volume with transverse and coronal measurements for estimating fetal cranial volume [13].	9
2.1	A labeled training set for supervised learning (e.g., spam classification) [2]	11
2.2	An unlabeled training set for unsupervised learning [2]	12
2.3	Semi-supervised learning [2]	12
2.4	Reinforcement learning [2]	13
2.5	A simple NN that takes the weighted sum of the input x and weights w . This weighted sum is then passed through the activation function to determine if the neuron fires. [4]	14
2.6	The structure of a biological neuron. Neurons are connected to other neurons through their dendrites and enurons. [4]	15
2.7	ANNs performing simple logical computations [2]	16
2.8	Top-left: Step function. Top-right: Sigmoid activation function. Mid-left: Hyperbolic tangent. Mid-right: ReLU activation (most used activation function for deep neural networks). Bottom-left: Leaky ReLU, variant of the ReLU that allows for negative values. Bottom-right: ELU, another variant of ELU that can often perform better than Leaky ReLU. [4]	17
2.9	A Venn diagram describing deep learning as a subfield of machine learning which is in turn a subfield of artificial intelligence [4]	19
2.10	CNNs and computer vision [3]	20
2.11	The convolution operation [3]	21
3.1	The heart, showing valves, arteries and veins. The white arrows show the normal direction of blood flow[17].	24
3.2	Multi-structure segmentation seen as a multi-label classification problem. 2D echocardiographic image with structures in the left and the ground truth mask in the right.	26

3.3	Architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations[8] .	27
3.4	EfficientNet architecture[9]	28
4.1	Tensorflow Logo [18]	36
4.2	SimpleITK Logo [19]	37
4.3	ITK-SNAP Logo [20]	37
4.4	Colab Logo	38
4.5	Scikit-learn Logo [10]	38
4.6	Keras Logo	39

List of Tables

1.1	Common image formats and their associated properties [5]	3
1.2	Summary of medical images file format characteristics [6]	8
3.1	Labels associated to identify cardiac structures	26
4.1	Results of LEFT VENTRICLE: ENDOCARDIUM	32
4.2	Results of LEFT VENTRICLE: EPICARDIUM	33
4.3	Results of LEFT ATRIUM	34

List of Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
CAMUS	Cardiac Acquisitions for Multi-structure Ultrasound Segmentation
CMYK	Cyan Yellow Magenta Black
CNN	Convolutional Neural Networks
DBN	Deep Belief Networks
DL	Deep Learning
D	Dice
ED	End Diastole
EF	Ejection Fraction
ES	End Systole
FP	False Positive
FT	False Negative
GPU	Graphics Processing Unit
HD	Hausdorff Distance
JAC	Jaccard index
LV	Left Ventricle
MAD	Mean Absolute Distance
ML	Machine Learning
NN	Neural Network
RBM	Restricted Boltzmann Machines
RGB	Red Green Blue
RV	Right Ventricle

TN	True Negative
TP	True Positive
UPN	Unsupervised Pretrained Network

General Introduction

AI can improve medical imaging processes like image analysis and aid in patient diagnosis. The medical imaging AI market is expected to experience exponential growth over the next few years.

Analysis of 2D echocardiographic images plays a crucial role in clinical routine to measure cardiac morphology and function and to arrive at a diagnosis. Such an analysis is based on the interpretation of clinical clues which are extracted from low level image processing such as segmentation and tracking.

We present a Deep Learning approach allowing to perform a segmentation of 2D echocardiographic images, by the delineation of the left ventricular endocardium in both end diastole (ED) and end systole (ES) in order to extract the ejection fraction (EF) of the left ventricle (LV).

We will use the CAMUS dataset in order to train our model and test its performance with different metrics.

Chapter 1

Image Processing

1.1 Introduction

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too. Image Processing nowadays refers mainly to the processing of digital images. In this chapter, we will present the basics of the image fundamentals, focusing on medical images.

1.2 Image

A panchromatic image is a 2D light intensity function, $f(x, y)$, where x and y are spatial coordinates and the value of f at (x, y) is proportional to the brightness of the scene at that point. If we have a multi spectral image, $f(x, y)$ is a vector, each component of which indicates the brightness of the scene at point (x, y) at the corresponding spectral band [1].

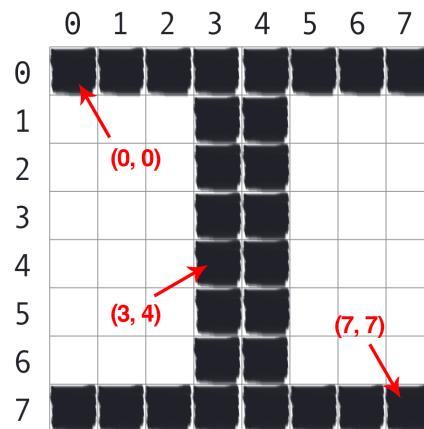


Figure 1.1: The letter “I” placed on a piece of graph paper. Pixels are accessed by their $(x; y)$ -coordinates, where we go x columns to the right and y rows down [4]

1.3 Image formats

From a mathematical viewpoint, any meaningful 2-D array of numbers can be considered as an image. In the real world, we need to effectively display images, store them (preferably compactly), transmit them over networks and recognize bodies of numerical data as corresponding to images. This has led to the development of standard digital image formats. In simple terms, the image formats comprise a file header (containing information on how exactly the image data is stored) and the actual numeric pixel values themselves. There are a large number of recognized image formats now existing, dating back over more than 30 years of digital image storage. Some of the most common 2-D image formats are listed in Table 1.1

Acronym	Name	Properties
GIF	Graphics interchange format	Limited to only 256 colours (8 bit); lossless compression
JPEG	Photographic Experts Group	In most common use today; lossy compression; lossless variants exist
BMP	Bit map picture	Basic image format; limited (generally) lossless compression; lossy variants exist
PNG	Portable network graphics	New lossless compression format; designed to replace GIF
TIF/TIFF	Tagged image (file) format	Highly flexible, detailed and adaptable format; compressed/uncompressed variants exist

Table 1.1: Common image formats and their associated properties [5]

As suggested by the properties listed in Table 1.1, different image formats are generally suitable for different applications. GIF images are a very basic image storage format limited to only 256 grey levels or colours, with the latter defined via a colour map in the file header as discussed previously. By contrast, the commonplace JPEG format is capable of storing up to a 24-bit RGB colour image, and up to 36 bits for medical/scientific imaging applications, and is most widely used for consumer-level imaging such as digital cameras. Other common formats encountered include the basic bitmap format (BMP), originating in the development of the Microsoft Windows operating system, and the new PNG format, designed as a more powerful replacement for GIF. TIFF, tagged image file format, represents an overarching and adaptable file format capable of storing a wide range of different image data forms. In general, photographic-type images are better suited towards JPEG or TIF storage, whilst images of limited colour/detail (e.g. logos, line drawings, text) are best suited to GIF or PNG (as per TIFF), as a lossless, full-colour format, is adaptable to the majority of image storage requirements.

1.4 Image Data Types

The choice of image format used can be largely determined by not just the image contents, but also the actual image data type that is required for storage. In addition to the bit resolution of a given image discussed earlier, a number of distinct image types also exist:

- *Binary images* are 2-D arrays that assign one numerical value from the set $\{0, 1\}$ to each pixel in the image. These are sometimes referred to as logical images: black corresponds to zero (an ‘off’ or ‘background’ pixel) and white corresponds to one (an ‘on’ or ‘foreground’ pixel). As no other values are permissible, these images can be represented as a simple bit-stream, but in practice they are represented as 8-bit integer images in the common image formats. A fax (or facsimile) image is an example of a binary image.



Figure 1.2: Binary image[14]

- *Intensity or grey-scale images* are 2-D arrays that assign one numerical value to each pixel which is representative of the intensity at this point. As discussed previously, the pixel value range is bounded by the bit resolution of the image and such images are stored as N-bit integer images with a given format.
- *RGB or true-colour images* are 3-D arrays that assign three numerical values to each pixel, each value corresponding to the red, green and blue (RGB) image channel component respectively. Conceptually, we may consider them as three distinct, 2-D planes so that they are of dimension C by R by 3, where R is the number of image rows and C the number of image columns. Commonly, such images are stored as sequential integers in successive channel order (e.g. $R_0G_0B_0, R_1G_1B_1, \dots$).
- *Floating-point* images differ from the other image types we have discussed. By definition, they do not store integer colour values. Instead, they store a floating-point number which, within a given range defined by the floating-point precision of the image bit-resolution, represents the



Figure 1.3: 8-bits grayscale image[15]



Figure 1.4: 8-bits RGB image

intensity. They may (commonly) represent a measurement value other than simple intensity or colour as part of a scientific or medical image. Floating point images are commonly stored in the TIFF image format or a more specialized, domain-specific format (e.g. medical DICOM). Although the use of floating-point images is increasing through the use of high dynamic range and stereo photography, file formats supporting their storage currently remain limited.

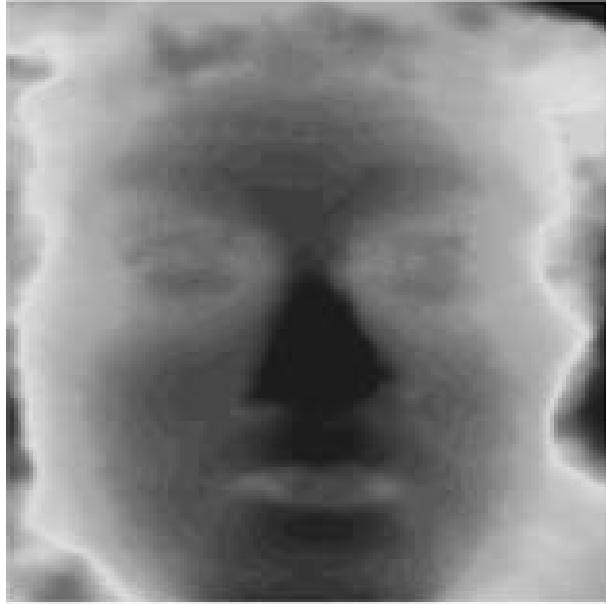


Figure 1.5: TIFF Floating point image (Copyright: Tim Lukins, UoE)

1.5 Semantic segmentation

Image segmentation is a long-standing computer vision task than consists on dividing an image into several parts, normally following some kind of criteria, but not necessarily pursuing an understanding of the image content. On the other hand, semantic segmentation is a different task that goes a step further than image segmentation by trying to divide an image into semantically meaningful parts (note that semantics is a branch of linguistics concerned with the meaning). More specifically, semantic segmentation is concerned about dividing the image into regions with different meaning or belonging to different categories. Very often these categories are a predefined set of objects that allow total segmentation of the image. An example of semantic segmentation is shown in figure 1.6.

1.6 Medical Images

1.6.1 Overview

The rapid progress of medical science and the invention of various medicines have benefited mankind and the whole civilization. Modern science also has been doing wonders in the surgical field. But, the proper and correct diagnosis of diseases is the primary necessity before the treatment. The more sophisticated the bio-instruments are, better diagnosis will be possible. The medical images play an important role in clinical diagnosis and therapy of doctor and teaching and researching etc. Medical imaging is often thought of as a way to represent anatomical structures of the body with the help of X-ray computed tomography and magnetic resonance imaging.

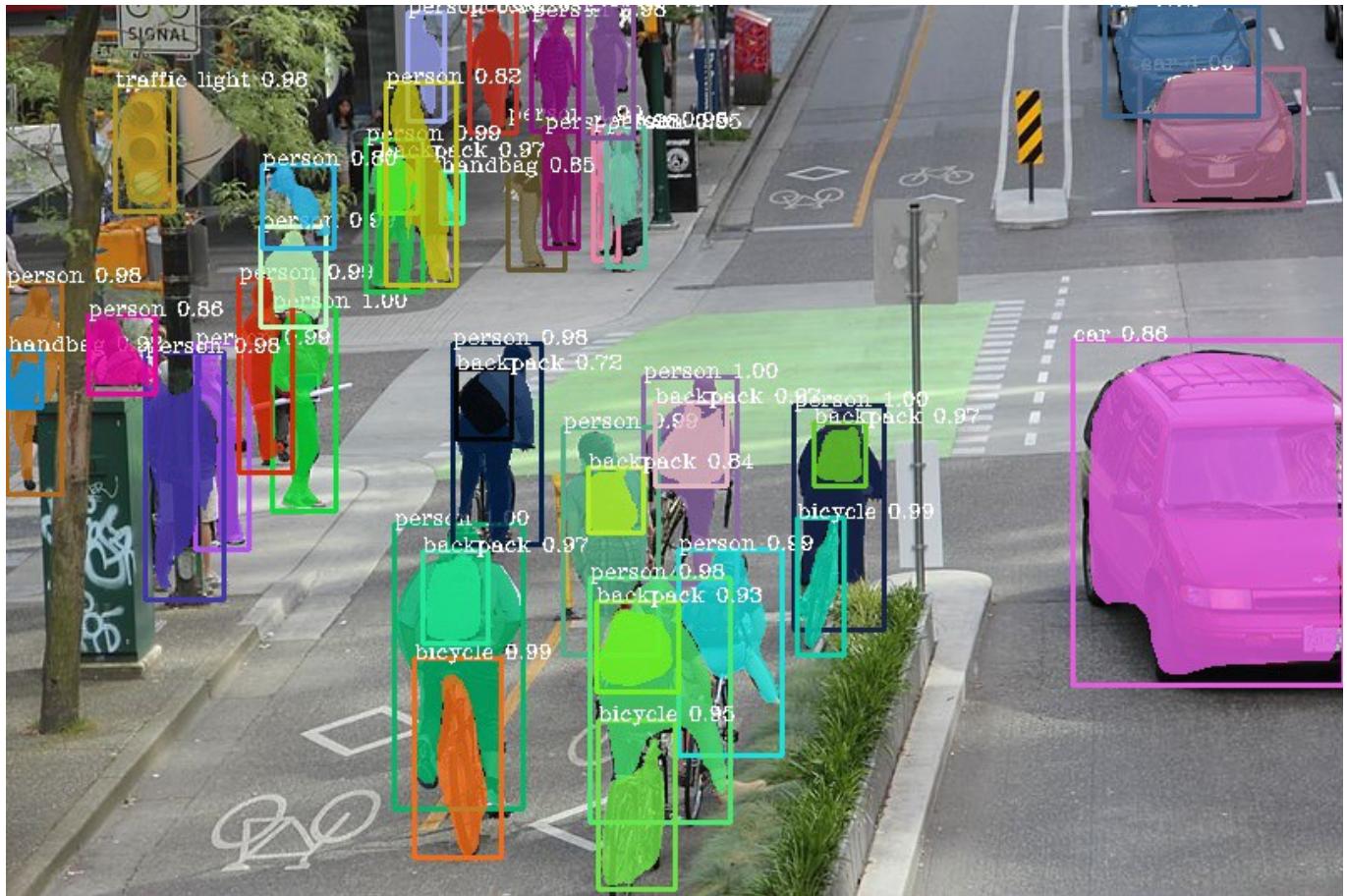


Figure 1.6: Example of semantic segmentation [11]

But often it is more useful for physiologic function rather than anatomy. With the growth of computer and image technology medical imaging has greatly influenced medical field. As the quality of medical imaging affects diagnosis the medical image processing has become a hotspot and the clinical applications wanting to store and retrieve images for future purpose needs some convenient process to store those images in details. Here are some types of medical imaging :

- Radiography
- Magnetic resonance imaging
- Nuclear medicine
- Ultrasound
- Tomography

Format	Header	Extension	Data types
Analyze	Fixed-length: 348 byte binary format	.img .hdr	Unsigned integer (8-bit), signed integer (16, 32-bit), float (32,64-bit), complex (64-bit)
Nifti	Fixed-length: 352 byte binary format a (348 byte in the case of data stored as .img and .hdr)	.nii	Signed and unsigned integer (from 8 to 64 bit), float (from 32 to 128 bit), complex (from 64 to 256 bit)
Minc	Extensible binary format	.mnc	Signed and unsigned integer (from 8 to 32 bit), float (32, 64 bit), complex (32, 64 bit)
Dicom	Variable length binary format	.dcm	Signed and unsigned integer, (8, 16 bit; 32 bit only allowed for radiotherapy dose), float not supported

Table 1.2: Summary of medical images file format characteristics [6]

1.6.2 Medical Ultrasound

Medical ultrasound uses high frequency broadband sound waves in the megahertz range that are reflected by tissue to varying degrees to produce (up to 3D) images. This is commonly associated with imaging the fetus in pregnant women. Uses of ultrasound are much broader, however. Other important uses include imaging the abdominal organs, heart, breast, muscles, tendons, arteries and veins. While it may provide less anatomical detail than techniques such as CT or MRI, it has several advantages which make it ideal in numerous situations, in particular that it studies the function of moving structures in real-time, emits no ionizing radiation, and contains speckle that can be used in elastography. Ultrasound is also used as a popular research tool for capturing raw data, that can be made available through an ultrasound research interface, for the purpose of tissue characterization and implementation of new image processing techniques. The concepts of ultrasound differ from other medical imaging modalities in the fact that it is operated by the transmission and receipt of sound waves. The high frequency sound waves are sent into the tissue and depending on the composition of the different tissues; the signal will be attenuated and returned at separate intervals. A path of reflected sound waves in a multilayered structure can be defined by an input acoustic impedance (ultrasound sound wave) and the Reflection and transmission coefficients of the relative structures. It is very safe to use and does not appear to cause any adverse effects. It is also relatively inexpensive and quick to perform. Ultrasound scanners can be taken to critically ill patients in intensive care units, avoiding the danger caused while moving the patient to the radiology department. The real-time moving image obtained can be used to guide drainage and biopsy procedures. Doppler capabilities on modern scanners allow the blood flow in arteries and veins to be assessed[12].

1.6.3 Echocardiography

When ultrasound is used to image the heart it is referred to as an echocardiogram. Echocardiography allows detailed structures of the heart, including chamber size, heart function, the valves of the heart, as well as the pericardium (the sac around the heart) to be seen. Echocardiography uses 2D, 3D, and Doppler imaging to create pictures of the heart and visualize the blood flowing through each of the four heart valves. Echocardiography is widely used in an array of patients ranging from those experiencing symptoms, such as shortness of breath or chest pain, to those undergoing cancer treatments. Transthoracic ultrasound has been proven to be safe for patients of all ages, from infants to the elderly, without risk of harmful side effects or radiation, differentiating it from other imaging modalities. Echocardiography is one of the most commonly used imaging modalities in the world due to its portability and use in a variety of applications. In emergency situations, echocardiography is quick, easily accessible, and able to be performed at the bedside, making it the modality of choice for many physicians[12].

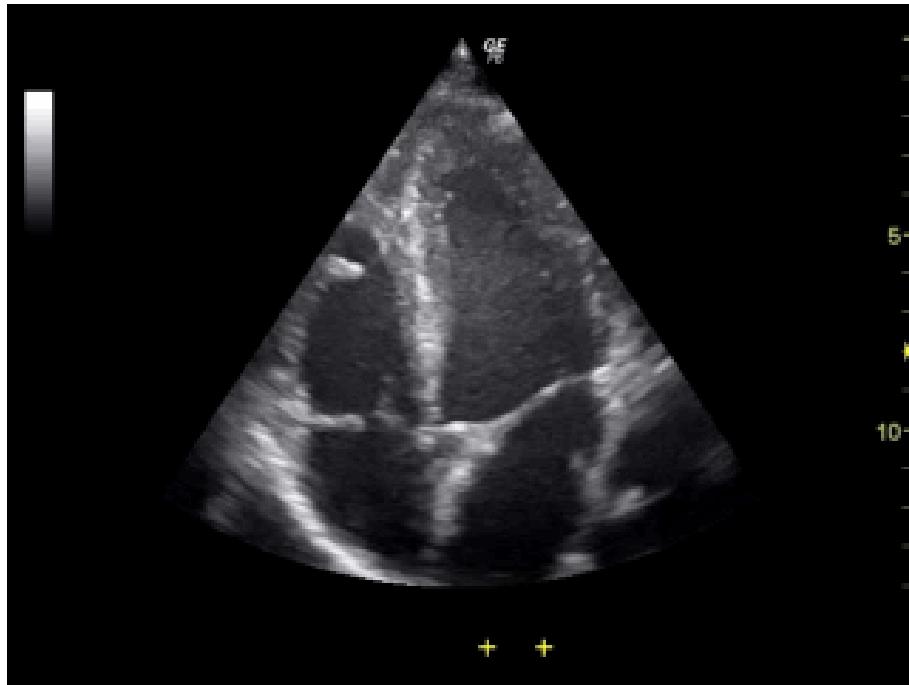


Figure 1.7: Orthogonal planes of a 3 dimensional sonographic volume with transverse and coronal measurements for estimating fetal cranial volume [13].

1.7 Conclusion

In this chapter, we have presented the fundamentals of image processing, detailing the echocardiographic images obtained from ultrasound that we will use later for our work.

Chapter 2

Machine learning and Deep Learning

2.1 Introduction

There are problems for which existing solutions require a lot of hand-tuning or long lists of rules: one Machine Learning algorithm can often simplify code and perform better. The availability of data and powerful hardware (GPU) made it possible to create machine learning and deep learning models. We will take an in-depth look at how we can do that.

2.2 Machine Learning

Machine learning is a subfield of computer science wherein machines learn to perform tasks for which they were not explicitly programmed. In short, machines observe a pattern and attempt to imitate it in some way that can be either direct or indirect [2].

2.2.1 Types of Machine Learning Systems

There are so many different types of Machine Learning systems that it is useful to classify them in broad categories based on:

- Whether or not they are trained with human supervision (supervised, unsupervised, semi-supervised, and Reinforcement Learning)
- Whether or not they can learn incrementally on the fly (online versus batch learning)
- Whether they work by simply comparing new data points to known data points, or instead detect patterns in the training data and build a predictive model, much like scientists do (instance-based versus model-based learning).

2.3 Supervised/Unsupervised Learning

Machine Learning systems can be classified according to the amount and type of supervision they get during training. There are four major categories: supervised learning, unsupervised learning, semi supervised learning, and Reinforcement Learning.

2.3.1 Supervised learning

In supervised learning, the training data you feed to the algorithm includes the desired solutions, called labels (Figure 2.1).

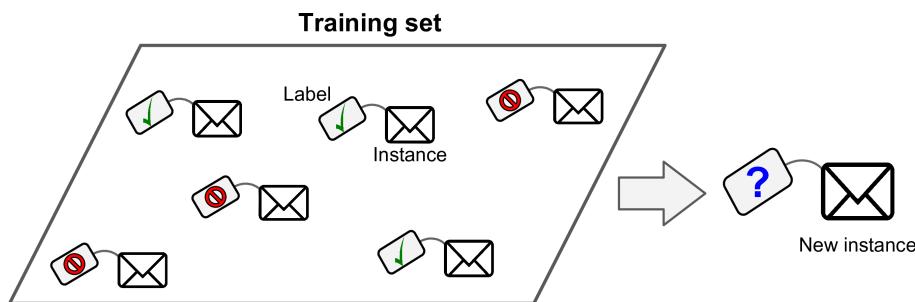


Figure 2.1: A labeled training set for supervised learning (e.g., spam classification) [2]

Here are some of the most important supervised learning algorithms:

- K-Nearest Neighbors.
- Linear Regression
- Support Vector Machines (SVMs).
- Decision Trees and Random Forests
- Neural networks.

2.3.2 Unsupervised learning

In unsupervised learning, the training data is unlabeled. The system tries to learn by itself (Figure 2.2).

Here are some of the most important unsupervised learning algorithms:

- Clustering
 - K-Means
 - Hierarchical Cluster Analysis

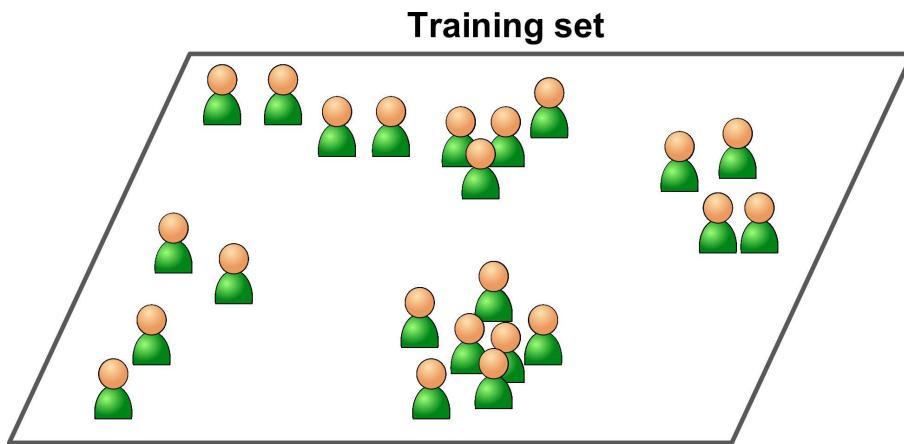


Figure 2.2: An unlabeled training set for unsupervised learning [2]

- Anomaly detection and novelty detection
 - One-class SVM
 - Isolation Forest
- Visualization and dimensionality reduction
 - Principal Component Analysis (PCA)
 - Locally-Linear Embedding (LLE)

2.3.3 Semi-supervised learning

Some algorithms can deal with partially labeled training data, usually a lot of unlabeled data and a little bit of labeled data. This is called *semi-supervised learning* (Figure 2.3).

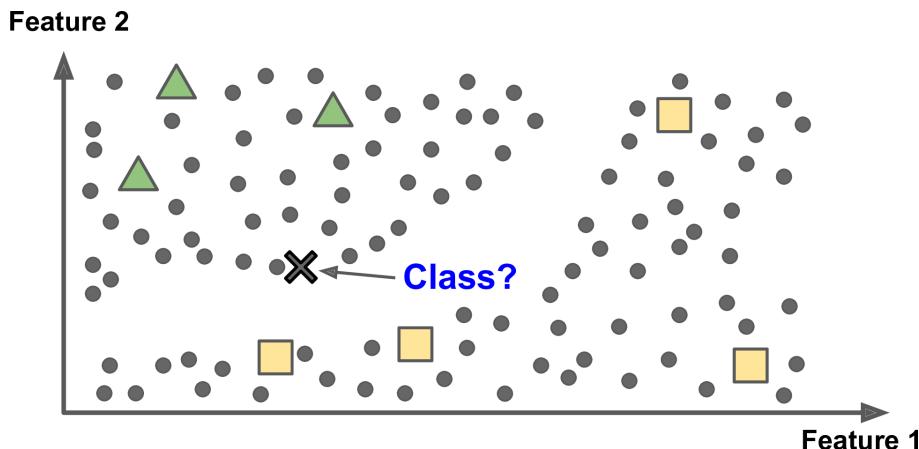


Figure 2.3: Semi-supervised learning [2]

Most semi-supervised learning algorithms are combinations of unsupervised and supervised algorithms. For example, deep belief networks (DBNs) are based on unsupervised components called restricted Boltzmann machines (RBMs) stacked on top of one another. RBMs are trained sequentially in an unsupervised manner, and then the whole system is fine-tuned using supervised learning techniques.

2.3.4 Reinforcement Learning

Reinforcement Learning is a very different beast. The learning system, called an agent in this context, can observe the environment, select and perform actions, and get rewards in return (or penalties in the form of negative rewards, as in Figure 2.4). It must then learn by itself what is the best strategy, called a policy, to get the most reward over time. A policy defines what action the agent should choose when it is in a given situation.

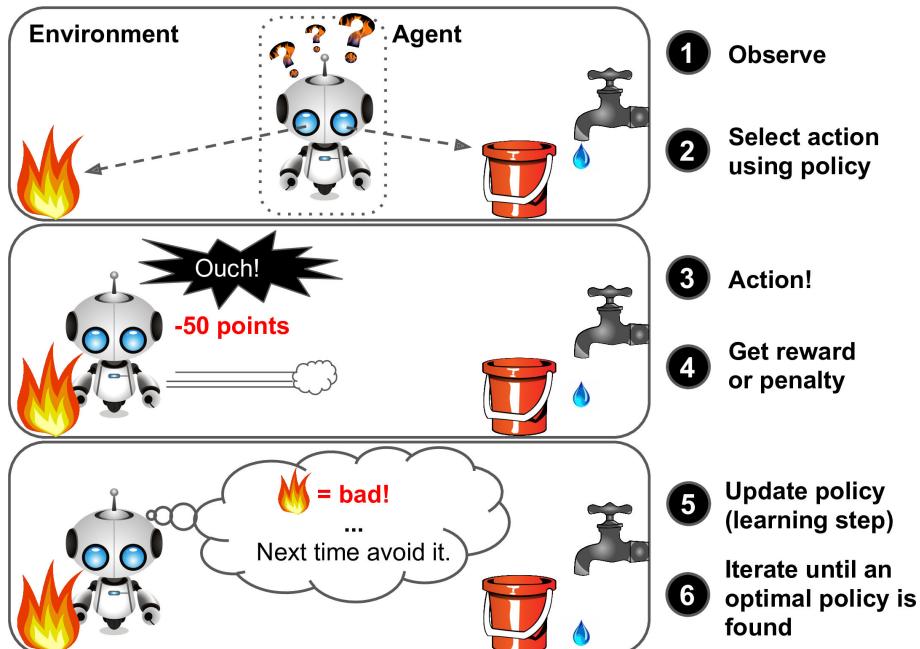


Figure 2.4: Reinforcement learning [2]

2.4 Neural networks

Many inventions were inspired by nature, burdock plants inspired velcro, and countless more inventions were inspired by nature. It seems only logical, then, to look at the brain's architecture for inspiration on how to build intelligent machines.

2.4.1 Definition

The word “neural” is the adjective form of “neuron”, and “network” denotes a graph-like structure; therefore, an “Artificial Neural Network” is a computation system that attempts to mimic (or at least, is inspired by) the neural connections in our nervous system. Artificial neural networks are also referred to as “neural networks” or “artificial neural systems”. It is common to abbreviate Artificial Neural Network and refer to them as “*ANN*” or simply “*NN*”.

For a system to be considered an NN, it must contain a labeled, directed graph structure where each node in the graph performs some simple computation. From graph theory, we know that a directed graph consists of a set of nodes (i.e., vertices) and a set of connections (i.e., edges) that link together pairs of nodes. In Figure 2.5 we can see an example of such an NN graph.

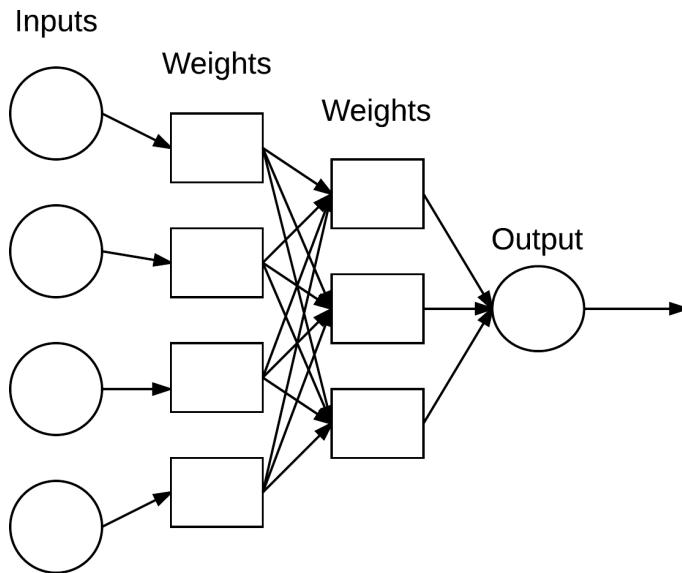


Figure 2.5: A simple NN that takes the weighted sum of the input x and weights w . This weighted sum is then passed through the activation function to determine if the neuron fires. [4]

Each node performs a simple computation. Each connection then carries a signal (i.e., the output of the computation) from one node to another, labeled by a weight indicating the extent to which the signal is amplified or diminished. Some connections have large, positive weights that amplify the signal, indicating that the signal is very important when making a classification. Others have negative weights, diminishing the strength of the signal, thus specifying that the output of the node is less important in the final classification. We call such a system an Artificial Neural Network if it consists of a graph structure (like in Figure 2.5) with connection weights that are modifiable using a learning algorithm.

Human Neuron Anatomy

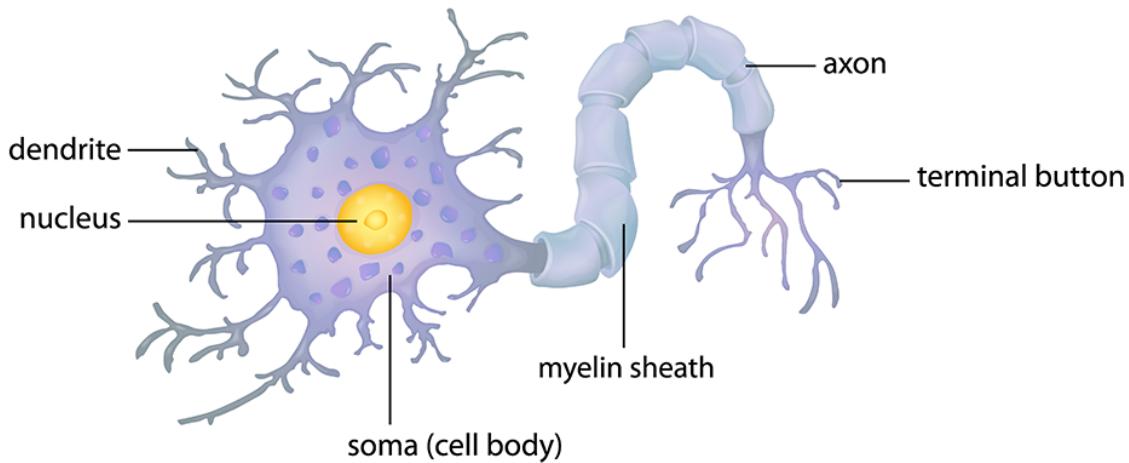


Figure 2.6: The structure of a biological neuron. Neurons are connected to other neurons through their dendrites and enurons. [4]

2.4.2 Relation to Biology

Our brains are composed of approximately 10 billion neurons, each connected to about 10,000 other neurons. The cell body of the neuron is called the soma, where the inputs (dendrites) and outputs (axons) connect soma to other soma (Figure 2.6).

Each neuron receives electrochemical inputs from other neurons at their dendrites. If these electrical inputs are sufficiently powerful to activate the neuron, then the activated neuron transmits the signal along its axon, passing it along to the dendrites of other neurons. These attached neurons may also fire, thus continuing the process of passing the message along.

The key takeaway here is that a neuron firing is a **binary operation – the neuron either fires or it doesn’t fire**. There are no different “grades” of firing. Simply put, a neuron will only fire if the total signal received at the soma exceeds a given threshold.

ANNs are simply inspired by what we know about the brain and how it works. The goal of deep learning is not to mimic how our brains function, but rather take the pieces that we understand and allow us to draw similar parallels in our own work. At the end of the day we do not know enough about neuroscience and the deeper functions of the brain to be able to correctly model how the brain works – instead, we take our inspirations and move on from there.

2.4.3 Artificial Models

Warren McCulloch and Walter Pitts proposed a very simple model of the biological neuron, which later became known as an artificial neuron: it has one or more binary (on/off) inputs and one binary output. The artificial neuron simply activates its output when more than a certain number

of its inputs are active. McCulloch and Pitts showed that even with such a simplified model it is possible to build a network of artificial neurons that computes any logical proposition you want. For example, Here's a few ANNs that perform various logical computations (Figure 2.7), assuming that a neuron is activated when at least two of its inputs are active.

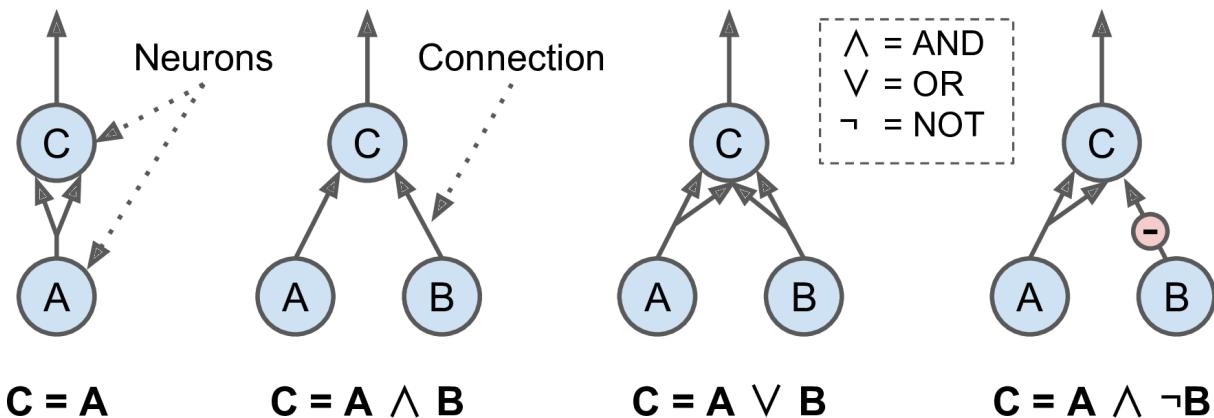


Figure 2.7: ANNs performing simple logical computations [2]

2.4.4 Activation function

Activation functions are mathematical equations that determine the output of a neural network. The function is attached to each neuron in the network, and determines whether it should be activated (“fired”) or not, based on whether each neuron’s input is relevant for the model’s prediction. Activation functions also help normalize the output of each neuron to a range between 1 and 0 or between -1 and 1.

An additional aspect of activation functions is that they must be computationally efficient because they are calculated across thousands or even millions of neurons for each data sample. Modern neural networks use a technique called backpropagation to train the model, which places an increased computational strain on the activation function, and its derivative function. The need for speed has led to the development of new functions such as ReLu and Swish.

2.4.5 Error Function

Defines how far the actual output of the current model is from the correct output. When training the model, the objective is to minimize the error function and bring output as close as possible to the correct value.

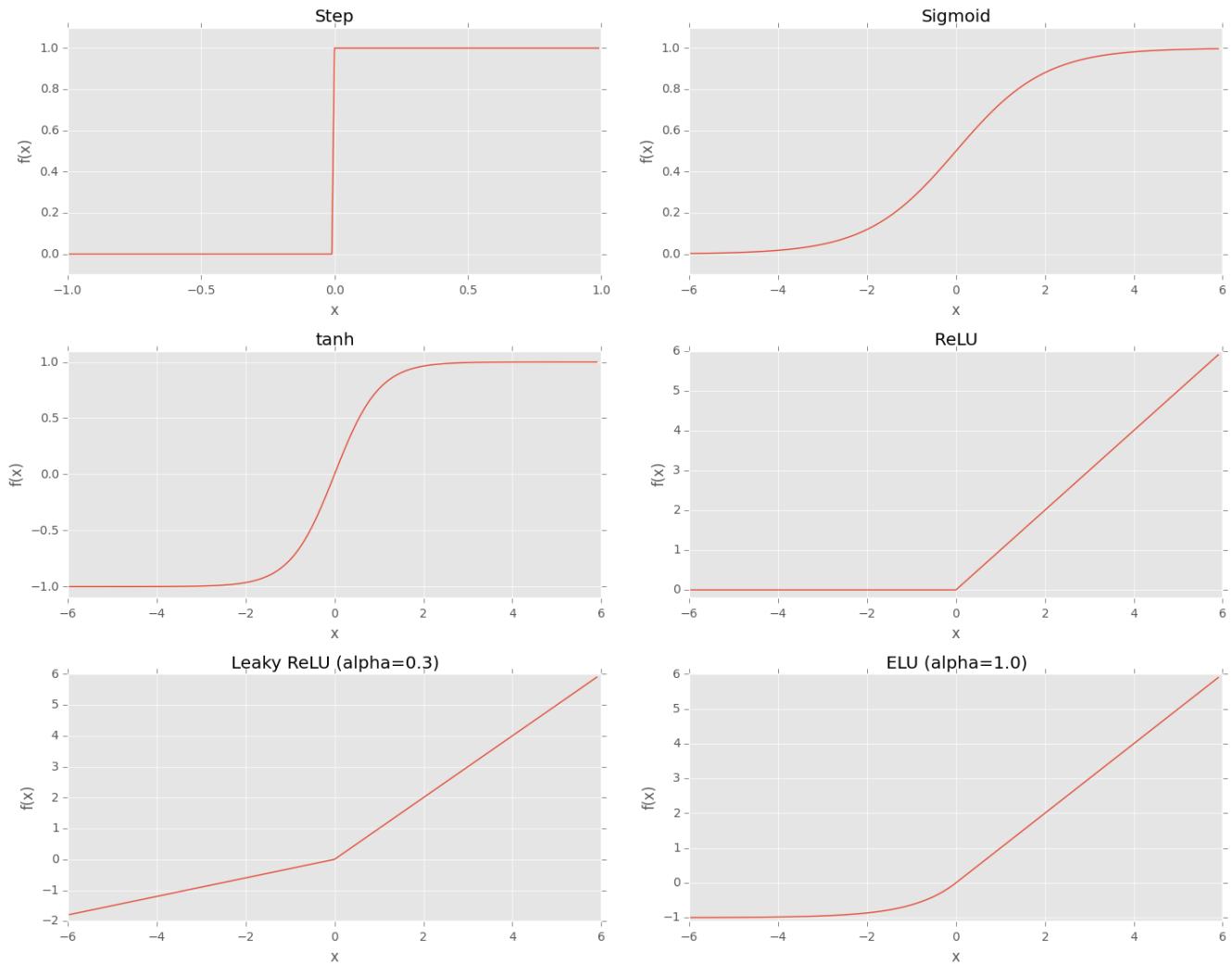


Figure 2.8: **Top-left:** Step function. **Top-right:** Sigmoid activation function. **Mid-left:** Hyperbolic tangent. **Mid-right:** ReLU activation (most used activation function for deep neural networks). **Bottom-left:** Leaky ReLU, variant of the ReLU that allows for negative values. **Bottom-right:** ELU, another variant of ELU that can often perform better than Leaky ReLU. [4]

2.4.6 Backpropagation

In order to discover the optimal weights for the neurons, we perform a backward pass, moving back from the network's prediction to the neurons that generated that prediction. This is called backpropagation. Backpropagation tracks the derivatives of the activation functions in each successive neuron, to find weights that bring the loss function to a minimum, which will generate the best prediction. This is a mathematical process called *gradient descent*.

2.4.7 Hyperparameters

A hyperparameter is a setting that affects the structure or operation of the neural network. In real deep learning projects, tuning hyperparameters is the primary way to build a network that provides accurate predictions for a certain problem. Common hyperparameters include the number of hidden layers, the activation function, and how many times (epochs) training should be repeated.

2.4.8 Overfitting and Underfitting in Neural Networks

Overfitting happens when the neural network is good at learning its training set, but is not able to generalize its predictions to additional, unseen examples. This is characterized by low bias and high variance. Underfitting happens when the neural network is not able to accurately predict for the training set, not to mention for the validation set. This is characterized by high bias and high variance.

2.5 Deep Neural Network

Deep learning is a subfield of machine learning, which is, in turn, a subfield of artificial intelligence (AI).

The central goal of AI is to provide a set of algorithms and techniques that can be used to solve problems that humans perform *intuitively* and *near automatically*, but are otherwise very challenging for computers. A great example of such a class of AI problems is interpreting and understanding the contents of an image – this task is something that a human can do with little-to-no effort, but it has proven to be *extremely difficult* for machines to accomplish.

While AI embodies a large, diverse set of work related to automatic machine reasoning (inference, planning, heuristics, etc.), the machine learning subfield tends to be *specifically* interested in **pattern recognition and learning from data**.

2.5.1 Major Architectures of Deep Networks

Now that we've seen some of the components of deep networks, let's take a look at the four major architectures of deep networks, but we will focus on the Convolutional Neural Networks (CNNs):

- Unsupervised Pretrained Networks (UPNs)
- Convolutional Neural Networks (CNNs)
- Recurrent Neural Networks
- Recursive Neural Networks

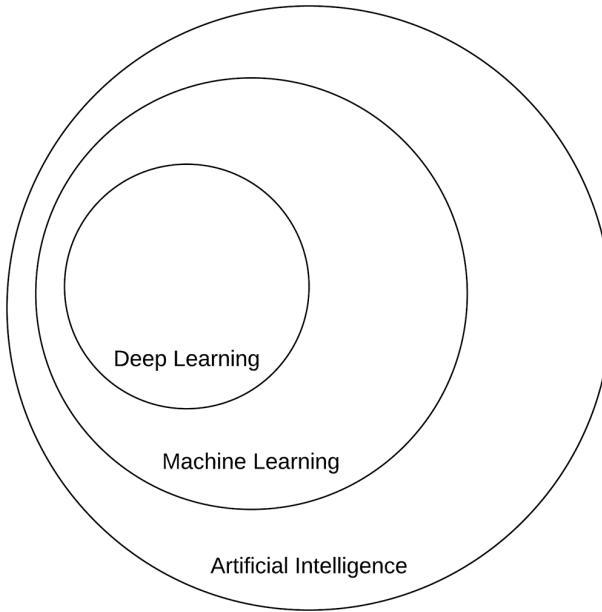


Figure 2.9: A Venn diagram describing deep learning as a subfield of machine learning which is in turn a subfield of artificial intelligence [4]

2.5.2 Convolutional Neural Networks (CNNs)

The goal of a CNN is to learn higher-order features in the data via convolutions. They are well suited to object recognition with images and consistently top image classification competitions. They can identify faces, individuals, street signs, platypuses, and many other aspects of visual data. CNNs overlap with text analysis via optical character recognition, but they are also useful when analyzing words as discrete textual units. They're also good at analyzing sound. [3]

The efficacy of CNNs in image recognition is one of the main reasons why the world recognizes the power of deep learning. As Figure 2.10 illustrates, CNNs are good at building position and (somewhat) rotation invariant features from raw image data.

2.5.2.1 Definition

A Convolutional Neural Network (CNN) is a deep learning algorithm that can recognize and classify features in images for computer vision. It is a multi-layer neural network designed to analyze visual inputs and perform tasks such as image classification, segmentation and object detection, which can be useful for autonomous vehicles. CNNs can also be used for **deep learning applications in healthcare**, such as medical imaging. There are two main parts to a CNN:

- A convolution tool that splits the various features of the image for analysis.
- A fully connected layer that uses the output of the convolution layer to predict the best description for the image.



Figure 2.10: CNNs and computer vision [3]

2.5.2.2 Basic Convolutional Neural Network Architecture

CNN architecture is inspired by the organization and functionality of the visual cortex and designed to mimic the connectivity pattern of neurons within the human brain.

The neurons within a CNN are split into a three-dimensional structure, with each set of neurons analyzing a small region or feature of the image. In other words, each group of neurons specializes in identifying one part of the image. CNNs use the predictions from the layers to produce a final output that presents a vector of probability scores to represent the likelihood that a specific feature belongs to a certain class.

2.5.2.3 CNN layers

A CNN is composed of several kinds of layers:

- Convolutional layer.

creates a feature map to predict the class probabilities for each feature by applying a filter that scans the whole image, few pixels at a time.

- Pooling layer (downsampling).

scales down the amount of information the convolutional layer generated for each feature and maintains the most essential information (the process of the convolutional and pooling layers usually repeats several times).

- Fully connected input layer.
“flattens” the outputs generated by previous layers to turn them into a single vector that can be used as an input for the next layer.
- Fully connected layer.
applies weights over the input generated by the feature analysis to predict an accurate label.
- Fully connected output layer.
generates the final probabilities to determine a class for the image.

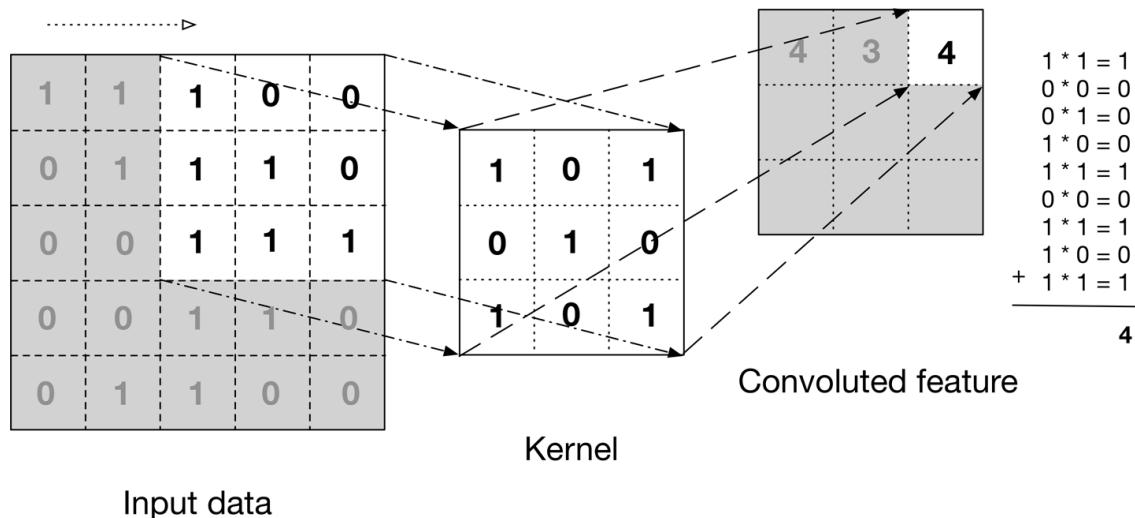


Figure 2.11: The convolution operation [3]

2.5.3 Deep Learning Applications in Healthcare

Deep learning techniques use data stored in EHR records to address many needed healthcare concerns like reducing the rate of misdiagnosis and predicting the outcome of procedures. By processing large amounts of data from various sources like medical imaging, ANNs can help physicians analyze information and detect multiple conditions:

- Analyze blood samples.
- Track glucose levels in diabetic patients.
- Detect heart problems.
- Using image analysis to detect tumors.
- Detecting cancerous cells and diagnosing cancer.

2.6 Conclusion

In this chapter, we have talked about the basics of Machine Learning and Deep Learning with the different learning algorithms used to build models that can help us in several areas. In the next chapter, we will look at a practical example of the use of Deep Learning and see how it would help us in the healthcare field by building a model to perform image segmentation on medical images from the CAMUS dataset.

Chapter 3

Dataset and Related Work

3.1 Introduction

Delineation of the cardiac structures from 2D echocardiographic images is a common clinical task to establish a diagnosis. Over the past decades, the automation of this task has been the subject of intense research. In this chapter, we will talk about the heart and its cardiac function as well as the CAMUS Dataset with which we will train our model to perform segmentation on 2D echocardiographic images.

3.2 Cardiac function

The heart is a hollow muscle, located at the level of the thorax between the lungs and resting on the diaphragm. This muscle is a pump whose function is to propel blood to all the organs of the body [16]. It pumps 5 liters of blood per minute and will beat about 3 billion times in a lifetime.

3.2.1 Anatomical structure of the heart

The internal structure of the heart is made up of 3 layers with from the inside to the outside: the **endocardium**, **myocardium** and **pericardium**. The pericardium is a double-walled sac that envelops the heart. The myocardium constitutes the heart muscle proper; it is a striated muscle, that is to say it has the same structure as the muscles of the limbs and the same force of contraction. The endocardium is an endothelial membrane that lines the inner surface of the heart [16]. The heart is divided into 4 chambers:

- 2 upper cavities: the right and left atria separated by the interatrial septum.
- 2 lower cavities: the right and left ventricles separated by the interventricular septum.

The atria communicate with the ventricles through the atrioventricular openings. We thus distinguish the right heart consisting of a right atrium and ventricle communicating through the tricuspid orifice and the left heart consisting of an atrium and a left ventricle communicating

through the mitral orifice. Each atrioventricular orifice includes a valve apparatus formed by a fibrous ring, valves, and cords connecting the valves to the muscle pillars inserting into the endocardium. The aortic (located at the entrance to the aorta) and pulmonary (located at the entrance to the pulmonary artery) orifices are made up of a fibrous ring and three so-called sigmoid valves[16].

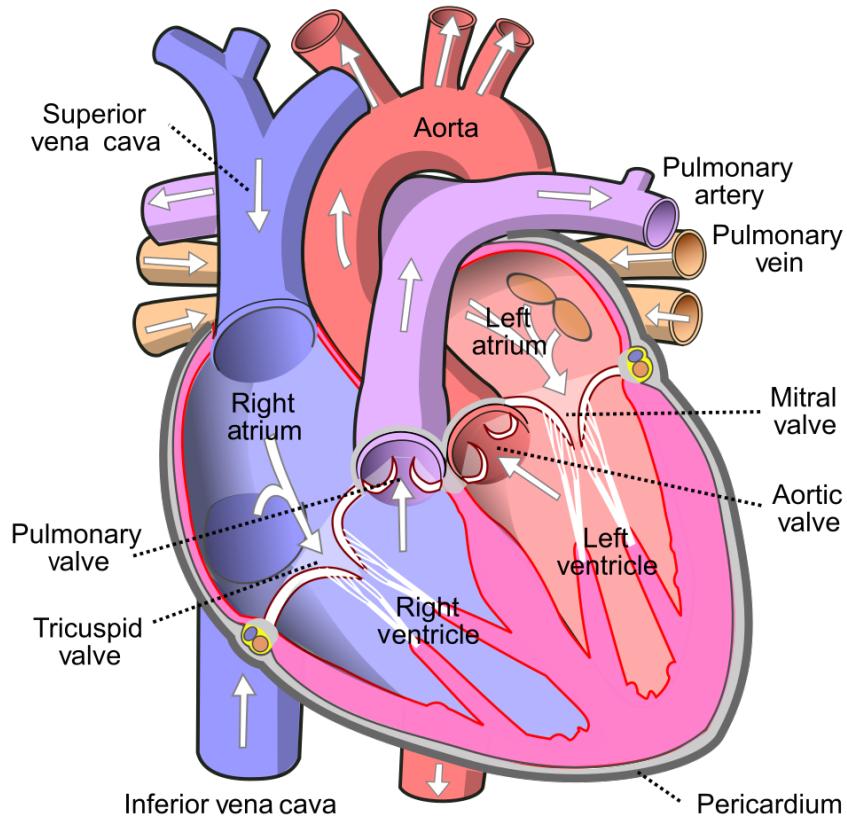


Figure 3.1: The heart, showing valves, arteries and veins. The white arrows show the normal direction of blood flow[17].

3.2.2 Heartbeat

Heart rate depends on 2 components:

- A mechanical component or cardiac cycle that can be simplified in 3 phases:[16]
 - A phase of relaxation called **diastole** allowing the filling of the heart chambers with blood;
 - A phase of contraction called **systole** which is characterized by an increase in intracavitory pressure;

- A phase of ejection of blood into the circulatory network;
- An electrical component directly responsible for the mechanical phase. The heart muscle is gifted with automatism: it has specialized muscle fibers that generate spontaneous repetitive electrical activity. These fibers make up what is called nodal tissue, which has a first cell cluster in the wall of the right atrium near the mouth of the superior vena cava: the sinoatrial node. This has the property of rhythmically generating an electrical impulse which propagates to the two atria and causes them to contract. Then the electrical signal will be transmitted to the atrioventricular node and then relayed to the ventricles thanks to the bundle of His and the network of Purkinje. The contraction of the ventricles occurs a few fractions of a second after that of the atria, due to the propagation time of the nerve impulse. On the other hand, there are quasi-simultaneous right and left atrial systoles (followed by diastoles) and also simultaneous systoles (followed by diastoles) of the right and left ventricles. Resting heart rate averages 60 to 80 beats per minute[16].

3.2.3 Blood flow

The left heart ensures the "systemic" circulation, also called "great circulation" feeding all the peripheral organs. The right heart ensures "the small circulation or pulmonary circulation" which aims to renew the blood gases in the pulmonary alveoli. Blood from organs, poor in oxygen and rich in wastes, is collected by the lower and upper vena cava to be returned to the right atrium. The atria contract and blood then passes from the right atrium to the right ventricle through the tricuspid valve. After the right ventricle contracts, the blood will be directed through the pulmonary artery to the lungs for oxygen. At the entrance to the pulmonary artery, the function of the pulmonary valve is to prevent the backflow of blood into the right ventricle. Once oxygenated, the blood returns to the left atrium through the pulmonary veins. Blood passes from the atrium to the left ventricle through the mitral valve. Finally, the oxygenated blood is propelled by the left ventricle into the aorta and will flow to all the organs. Again, the aortic valve prevents backflow of blood into the left ventricle[16].

3.3 CAMUS Dataset

CAMUS stands for "Cardiac Acquisitions for Multi-structure Ultrasound Segmentation". It contains 2D echocardiographic sequences with two and four-chamber views of 500 patients that were acquired with the same equipment in the same medical center. The size of this dataset and its tight connection to everyday clinical issues give the possibility to train deep learning methods to automatically analyze echocardiographic data. In addition, CAMUS includes manual expert annotations for the left ventricle endocardium (LV_{Endo}), the myocardium (epicardium contour more specifically, named LV_{Epi}) and the left atrium (LA)[7]. The purpose of this clinical dataset is to :

- enable to appropriately train machine / deep learning models;
- allow a meaningful comparison between state-of-the-art methods;
- evaluate how far supervised learning methods can go at assessing 2D echocardiographic images i.e. segmenting cardiac structures as well as estimating clinical indices.

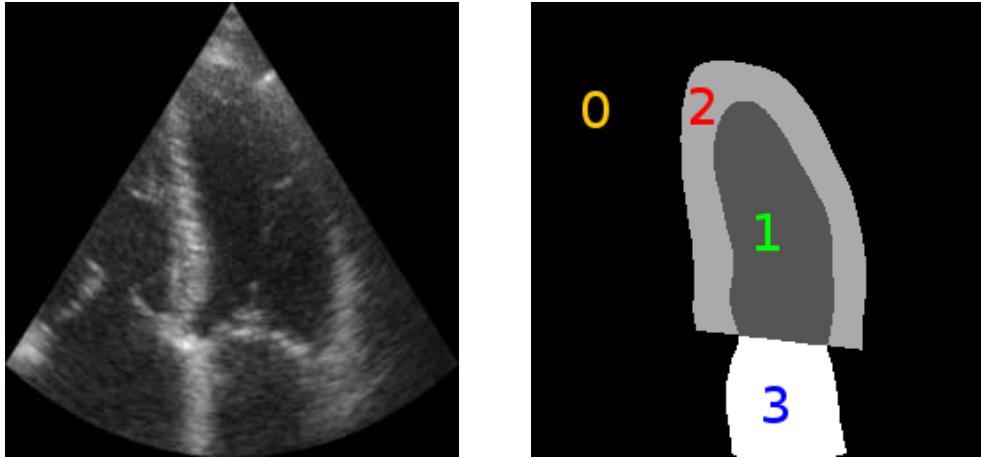


Figure 3.2: Multi-structure segmentation seen as a multi-label classification problem. 2D echocardiographic image with structures in the left and the ground truth mask in the right.

Structure	Left ventricle	Myocardium	Left atrium	Other (background)
Label	1	2	3	0

Table 3.1: Labels associated to identify cardiac structures

The CAMUS dataset consists of clinical exams from 500 patients, acquired at the University Hospital of St Etienne (France) and included in this study within the regulation set by the local ethical committee of the hospital[7].

450 of the 500 patients were made available with the mask to perform supervised learning, the other 50 are without a mask in order to measure the performance of for the segmentation of cardiac structures, Via a dedicated online platform. Note that the images of a patient have different quality (poor, medium, good).

3.4 Our approach

3.4.1 Model

The model being used here is a modified U-Net. A U-Net consists of an encoder (downsampler) and decoder (upsampler). In-order to learn robust features, and reduce the number of trainable

parameters. The encoder for this task will be EfficientNet B0 model, whose intermediate outputs will be used, and the decoder will be the upsample block.

3.4.2 U-NET architecture

U-Net is a convolutional neural network that was developed for biomedical image segmentation at the Computer Science Department of the University of Freiburg, Germany. The network is based on the fully convolutional network and its architecture was modified and extended to work with fewer training images and to yield more precise segmentations[8].

Unet architecture uses standard 3x3 convolutions followed by a ReLU activation function and a 2x2 max pooling layer this architecture has 23 convolution layers in total and it ends with a 1x1 convolution and a sigmoid as an activation function.

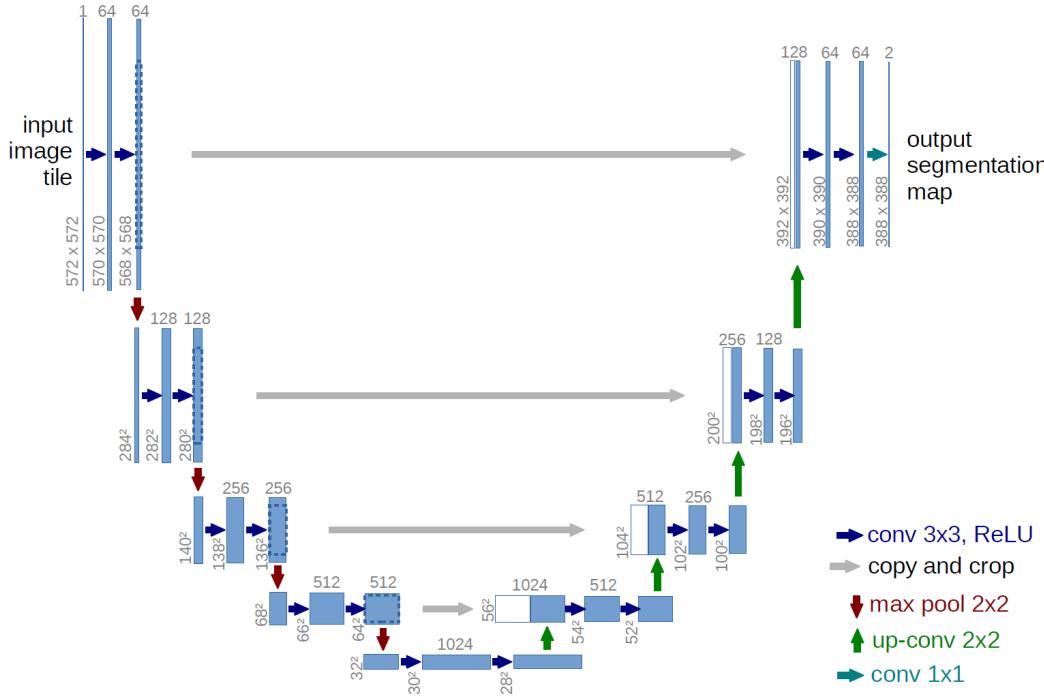


Figure 3.3: Architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations[8]

And although Unet is an old architecture (2015) but it remains difficult to beat in the field of segmentation of medical images and all this thanks to its skip connections.

3.5 EfficientNet architecture

The EfficientNet-B0 architecture is not developed by engineers, but by the neural network itself. They developed the model using research on a multi-objective neural architecture that optimizes precision and floating-point operations. Based on B0, the author developed a complete series of EfficientNets from B1 to B7, which achieved the highest accuracy on ImageNet and were also very effective for competitors.

EfficientNet architecture is a very recent architecture (2019) it has proven its efficiency in terms of precision and speed its power comes from three main elements.

3.5.1 Depthwise Separable Convolution

By adopting the original two-step convolution depthwise and pointwise, the calculation cost can be greatly reduced, and the loss of precision can be reduced at the same time.

3.5.2 Inverse Res

Linear activation is used in the last layer of each block to avoid losing ReLU information. The main component of EfficientNet is MBConv, which is a reverse bottleneck conveyor belt, originally called MobileNetV2. By connecting a smaller number of channels (relative to the expansion layer) using a shortcut between the bottlenecks, combining it with a separable deep convolution, which reduces the amount of computation of nearly a thousand. Compared to the traditional layer, k represents the kernel size, which specifies the height and width of the two-dimensional convolution window.

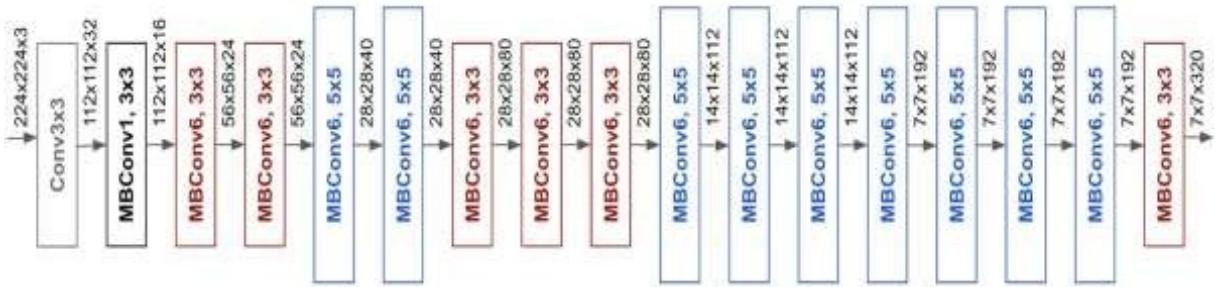


Figure 3.4: EfficientNet architecture[9]

3.6 Model specifications

3.6.1 Processing part

- Convert images from *mhd* format to *nii* format;

- Select only good and medium quality images;
- Resize all images to the shape (352, 352, 1);
- Normalise images to speed up training phase;
- Data augmentation with a flip from left to right;

3.6.2 Training part

- Splitting the dataset to 90% portion for training and 10% for validation;
- Shuffle the training portion;
- Batch Size of 1;
- Adam as an optimizer;
- Sparse Categorical Crossentropy from logits as a loss;
- Accuracy as metrics;
- 60 epochs;
- NVIDIA T4 GPU;

3.6.3 Training results

after the final epoch of the training phase, we obtained a 97% accuracy percentage on the training data as well as validation data with a total duration of almost 2 hours, which is very fast thanks to the powerful GPU we used.

3.7 Conclusion

We presented in this chapter, general information on the heart and its anatomical structure, as well as the CAMUS challenge to achieve a segmentation of cardiac structures by using our approach. In the next chapter we will talk about performance indices and test results obtained through our approach.

Chapter 4

Experimental Results

4.1 Introduction

In the previous chapter, we talked about the cardiac structure of the heart as well as the CAMUS dataset and our model based on U-NET and EfficientNet. In this chapter, we will show the results obtained from 31 patients with our approach, but before that we will define the performance metrics used by the CAMUS challenge.

4.2 Medical image segmentation metrics

Standard evaluation in medical imaging requires the establishment of task-specific sets of quality criteria designed to provide meaningful information over the degree of agreement between method predictions and expert annotations. Segmentation metrics traditionally focus on the accuracy of contouring. We present here the classical metrics used in the case of binary segmentation which separates a single structure (labeled 1) from the background (labeled 0).

4.2.1 Region overlap

Overlap metrics give global scores on the quality of the image segmentation. All overall indices are derived from the four cardinalities of the confusion matrix:

1. True Positive (TP): number of elements (pixels / voxels) correctly assigned to 1;
2. False Positive (FP): number of elements wrongly assigned to 1;
3. True Negative (TN): number of elements correctly assigned to 0;
4. False Negative (FN): number of elements wrongly assigned to 0;

The sensitivity (true positive rate), specificity (true negative rate) and fallout (false positive rate) are often used to evaluate classification tasks. However, they are not truly appropriate for

segmentation as they are very sensitive to objects size. The overlap index between two segmentations A and B is called the Dice, and noted $D(A, B)$ thereafter. The Dice and the Jaccard Index JAC (also called the intersection of union IOU) are less sensitive to the number of elements and more frequently used in the literature:

$$D(A, B) = 2 * \frac{A \cap B}{|A| + |B|} = 2 * \frac{TP}{2TP + FP + FN} = 2 * \frac{JAC}{1 + JAC}$$

In our study, we use the Dice (best score of 1) to represent the overall performance of segmentation methods.

4.2.2 Spatial distances between contours

Spatial distance based metrics allow a better assessment of the contouring accuracy. They are often restricted to the elements of the contours for computational efficiency. The Mean Absolute Distance (MAD) represents the average error in segmentation, while the Hausdorff Distance (HD) shows the maximum error. Let $d_{C_1}(C_2)$ be the set of euclidean distances obtained from perpendicularly projecting the points of contour C_2 onto the contour C_1 . To ensure a symmetric behavior, we use the corresponding formulations for *MAD* and *HD*:

$$MAD(C_A, C_B) = \frac{\overline{d_{C_A}(C_B)} + \overline{d_{C_B}(C_A)}}{2}$$

$$HD(C_A, C_B) = \max(\max d_{C_A}(C_B)), \max d_{C_B}(C_A)$$

Where C_A and C_B are the sets of points of each object, d the euclidean distance, and $\overline{\bullet}$ the average operator.

The association of Dice and HD enables to encode both local and global accuracy through the measurement of the overlap and maximum error. In this study, we also indicate the MAD to represent the average closeness of the borders.

4.3 Results

In these 3 tables, we present the results obtained after submitting the images of the 50 test patients on the online platform of the CAMUS challenge. Note that we have only presented 31 patients, as the other 19 have metadata issues (which we will try to fix in the future).

	ED dice	ES dice	ED hausdorff	ES hausdorff	ED mad	ES mad
mean	0.917	0.88	27.1	43.7	10.48	13.06
patient1	0.924	0.913	48.9	55.8	11.3	13.4
patient10	0.929	0.877	35.9	45.3	7.8	10.9
patient11	0.943	0.918	47.0	57.9	10.1	12.5
patient12	0.937	0.923	29.4	38.1	6.6	8.8
patient14	0.941	0.919	54.1	55.0	11.1	12.4
patient16	0.900	0.837	40.5	48.0	12.7	15.5
patient17	0.917	0.937	53.5	54.8	11.6	11.0
patient18	0.919	0.888	50.2	65.3	11.5	17.0
patient19	0.912	0.899	44.7	53.3	11.1	13.8
patient22	0.936	0.908	43.2	48.7	11.1	12.4
patient23	0.903	0.893	39.3	44.0	8.8	9.4
patient24	0.933	0.903	32.4	40.8	8.5	11.9
patient25	0.931	0.909	33.9	42.1	8.6	11.0
patient26	0.927	0.905	48.6	55.7	9.5	12.7
patient27	0.857	0.875	50.1	59.1	12.3	15.3
patient28	0.894	0.840	62.3	73.9	14.3	22.3
patient29	0.935	0.879	44.3	55.5	11.7	14.8
patient3	0.946	0.941	50.0	46.8	9.6	10.1
patient30	0.944	0.921	44.2	50.2	10.5	11.6
patient31	0.901	0.764	40.2	52.9	11.4	15.1
patient32	0.903	0.864	41.3	50.5	11.6	13.9
patient34	0.895	0.831	30.4	39.3	9.5	12.2
patient35	0.929	0.879	41.0	48.9	9.7	12.9
patient36	0.927	0.908	35.0	46.5	9.4	11.8
patient39	0.891	0.881	31.4	43.2	11.0	12.9
patient4	0.943	0.950	56.8	58.5	11.4	11.9
patient41	0.913	0.867	53.0	62.4	11.4	14.7
patient43	0.911	0.874	40.7	47.3	10.4	13.1
patient47	0.904	0.881	55.0	64.2	13.3	17.2
patient5	0.917	0.919	36.7	43.5	9.1	10.6
patient6	0.909	0.859	39.4	47.1	8.6	11.0

Table 4.1: Results of LEFT VENTRICLE: ENDOCARDIUM

	ED dice	ES dice	ED hausdorff	ES hausdorff	ED mad	ES mad
mean	0.956	0.946	43.4	51.0	10.5	14.0
patient1	0.958	0.956	48.9	55.8	11.8	15.1
patient10	0.964	0.956	35.9	45.3	7.6	11.2
patient11	0.970	0.960	47.0	57.9	10.6	15.0
patient12	0.948	0.946	29.4	38.1	6.3	9.3
patient14	0.956	0.948	54.1	55.0	14.6	15.8
patient16	0.944	0.941	40.5	48.0	9.6	12.3
patient17	0.947	0.954	53.5	54.8	16.1	14.4
patient18	0.969	0.948	50.0	62.8	12.2	20.1
patient19	0.950	0.944	44.7	53.3	11.1	14.8
patient22	0.962	0.964	43.2	48.7	10.1	12.3
patient23	0.938	0.934	39.3	44.0	8.9	10.5
patient24	0.962	0.965	32.4	40.8	6.6	10.8
patient25	0.961	0.939	33.9	42.1	7.4	11.6
patient26	0.951	0.960	44.2	55.4	10.6	15.0
patient27	0.942	0.928	50.1	59.1	12.8	17.1
patient28	0.953	0.895	61.8	65.7	17.0	23.9
patient29	0.972	0.956	44.3	55.5	10.5	16.3
patient3	0.968	0.952	50.0	46.8	10.4	11.4
patient30	0.966	0.967	44.0	50.2	10.2	12.4
patient31	0.957	0.937	40.2	52.9	10.1	15.3
patient32	0.955	0.944	41.3	50.5	9.9	13.6
patient34	0.956	0.930	30.4	39.3	7.0	10.6
patient35	0.964	0.929	40.5	48.1	9.9	13.3
patient36	0.966	0.966	35.0	46.5	7.1	11.4
patient39	0.944	0.943	31.4	43.2	6.9	11.2
patient4	0.968	0.960	56.8	58.5	15.9	16.5
patient41	0.963	0.930	53.0	62.4	13.2	18.3
patient43	0.967	0.956	40.5	47.3	10.0	13.1
patient47	0.963	0.950	55.0	64.2	15.3	19.1
patient5	0.964	0.953	36.7	43.5	7.5	10.2
patient6	0.920	0.932	39.4	47.0	9.2	12.1

Table 4.2: Results of LEFT VENTRICLE: EPICARDIUM

	ED dice	ES dice	ED hausdorff	ES hausdorff	ED mad	ES mad
mean	0.889	0.92	6.0	7.7	2.4	1.9
patient1	0.947	0.962	4.0	3.3	1.3	1.1
patient10	0.927	0.932	3.8	4.9	1.2	1.4
patient11	0.926	0.948	5.1	4.1	1.6	1.3
patient12	0.918	0.917	3.0	5.6	1.3	1.8
patient14	0.942	0.917	3.5	5.3	1.4	2.1
patient16	0.891	0.917	4.9	4.3	2.0	1.6
patient17	0.861	0.905	10.8	6.9	4.1	2.5
patient18	0.933	0.954	5.0	4.2	1.4	1.3
patient19	0.929	0.909	4.1	5.2	1.4	2.3
patient22	0.913	0.923	4.5	4.5	1.9	1.8
patient23	0.926	0.946	3.4	3.1	1.2	1.1
patient24	0.912	0.913	3.1	4.1	1.4	1.8
patient25	0.913	0.920	3.4	4.3	1.4	1.5
patient26	0.907	0.946	3.9	3.8	1.9	1.1
patient27	0.897	0.907	7.7	6.8	2.4	2.4
patient28	0.919	0.941	5.6	36.4	2.1	2.0
patient29	0.944	0.949	3.5	5.7	1.1	1.3
patient3	0.770	0.800	15.4	15.8	4.8	5.2
patient30	0.903	0.938	5.9	4.3	1.9	1.4
patient31	0.913	0.926	5.3	43.2	1.8	3.5
patient32	0.428	0.922	22.6	4.0	11.6	1.9
patient34	0.871	0.879	5.8	6.9	2.0	2.1
patient35	0.839	0.861	6.8	10.8	3.6	3.5
patient36	0.925	0.932	3.1	3.0	1.2	1.4
patient39	0.922	0.877	3.1	6.3	1.2	2.8
patient4	0.935	0.953	9.1	6.3	2.0	1.4
patient41	0.932	0.929	3.4	4.5	1.4	1.9
patient43	0.905	0.927	4.3	4.4	1.8	1.6
patient47	0.889	0.947	8.7	5.8	3.0	1.5
patient5	0.804	0.911	10.3	4.8	3.9	1.8
patient6	0.897	0.917	4.3	6.0	1.8	1.8

Table 4.3: Results of LEFT ATRIUM

4.4 Conclusion

In this chapter, we have defined the metrics used in the CAMUS challenge to evaluate the performance of the segmentation methods. We can see from the data in the tables 4.1, 4.2, 4.3, that our model based on U-NET and EfficientNet gives very good results, especially in the dice score, most of which exceed 0.8.

Tools and software

4.5 Tensorflow

TensorFlow is a powerful library for numerical computation, particularly well suited and fine-tuned for large-scale Machine Learning (but you could use it for anything else that requires heavy computations). It was developed by the Google Brain team and it powers many of Google's large-scale services, such as Google Cloud Speech, Google Photos, and Google Search. It was open sourced in November 2015, and it is now the most popular deep learning library (in terms of citations in papers, adoption in companies, stars on github, etc.): countless projects use TensorFlow for all sorts of Machine Learning tasks, such as image classification, natural language processing (NLP), recommender systems, time series forecasting, and much more.[2]

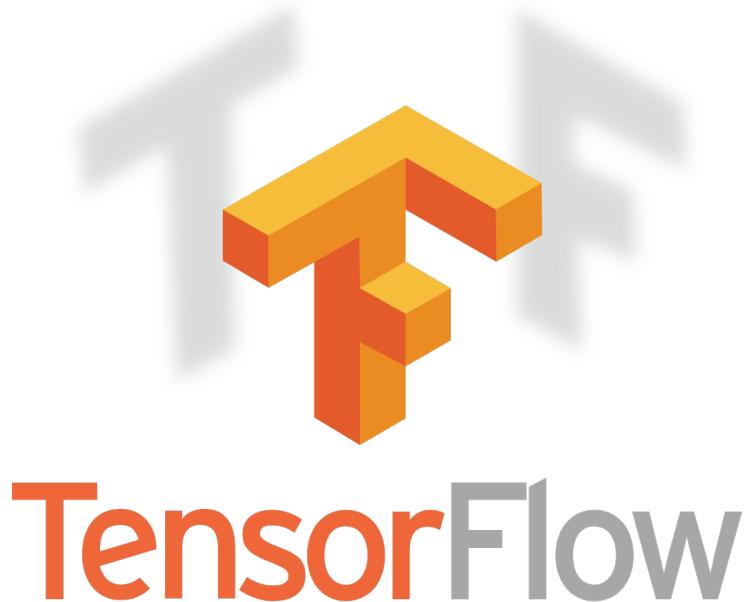


Figure 4.1: Tensorflow Logo [18]

4.6 SimpleITK

Open-source multi-dimensional image analysis in Python, R, Java, C, Lua, Ruby, TCL and C++. Developed by the Insight Toolkit community for the biomedical sciences and beyond.[19]



Figure 4.2: SimpleITK Logo [19]

4.7 ITK-SNAP

ITK-SNAP is a software application used to segment structures in 3D medical images. It is the product of a decade-long collaboration, whose vision was to create a tool that would be dedicated to a specific function, segmentation, and would be easy to use and learn. ITK-SNAP is free, open-source, and multi-platform. ITK-SNAP provides semi-automatic segmentation using active contour methods, as well as manual delineation and image navigation. In addition to these core functions, ITK-SNAP offers many supporting utilities.[20]

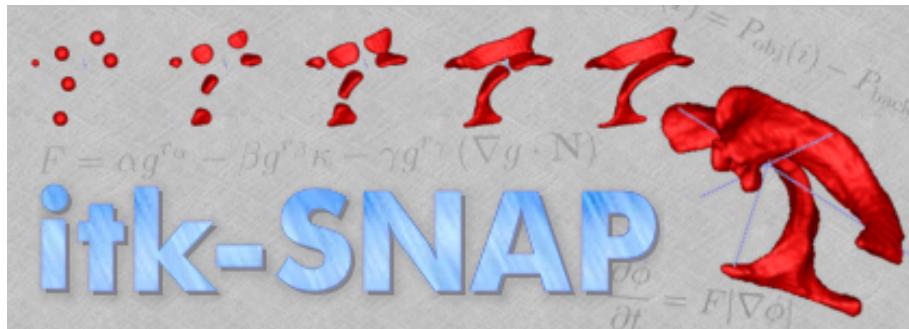


Figure 4.3: ITK-SNAP Logo [20]

4.8 Google Colab

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.



Figure 4.4: Colab Logo

4.9 Scikit-learn

Scikit-learn is an open source machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection and evaluation, and many other utilities. [10]



Figure 4.5: Scikit-learn Logo [10]

4.10 Keras

tf.keras is TensorFlow's high-level API for building and training deep learning models. It is used in rapid prototyping, advanced research and going into production. It has three major advantages:

- Friendliness

Keras has a simple and consistent interface optimized for common use cases. It provides clear and concrete information about user errors.

- Modularity and ease of composition

Keras models are created by connecting configurable components, with some restrictions.

- Ease of extension

Compose custom building blocks to express new research ideas. Create layers, metrics, and loss functions, and develop cutting-edge models.[21]

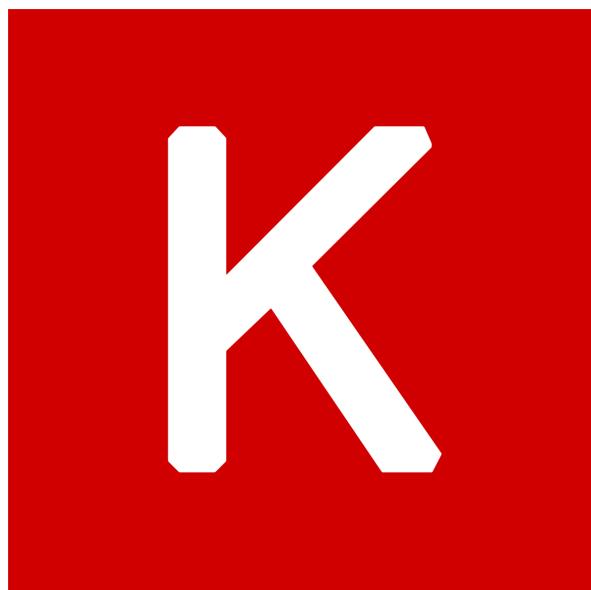


Figure 4.6: Keras Logo

Conclusion

Part of the work carried out within the framework of this thesis involves the automatic segmentation of echocardiographic images.

The separation and identification of the different structures from accurate delineation, called semantic segmentation, is the first step to measure surfaces or volumes.

However, segmentation in echocardiography is a particularly difficult task due to the lack of clear boundaries, a low signal-to-noise ratio, the speckled texture specific to ultrasound images, and the presence of numerous and complex image artifacts such as reverberations and loss of signal.

We have presented a fully automatic deep learning approach based on the U-NET architecture by integrating EfficientNet as an encoder.

Our network has achieved 97% accuracy on training data as well as validation data, which makes our network powerful.

The results of the test on the CAMUS challenge dataset clearly show this with a Dice score above 0.8. As prospects, we want to make changes to our network using the transfer learning technique in order to improve it and solve the problem of metadata of the 19 patients.

Bibliography

- [1] M. PETROU and C. PETROU, *Image Processing: The Fundamentals*, 2nd edition, Wiley, Chichester, 2010.
- [2] A. GÉRON, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd edition, O'Reilly, Sebastopol, 2019.
- [3] J. PATTERSON and A. GIBSON *Deep Learning*, first edition, O'Reilly, Sebastopol, 2017.
- [4] A. ROSEBROCK, *Deep Learning for Computer Vision with Python*, First edition, PYIMAGE-SEARCH, 2017.
- [5] C. SOLOMON and T. BRECKON, *Fundamentals of Digital Image Processing*, 2nd edition, Wiley-Blackwell, Chichester, 2011.
- [6] M. Larobina, L. Murino, Medical Image File Formats, Journal of Digital Imaging, 2013.
- [7] S. Leclerc et al., "Deep Learning for Segmentation Using an Open Large-Scale Dataset in 2D Echocardiography," in IEEE Transactions on Medical Imaging, vol. 38, no. 9, pp. 2198-2210, Sept. 2019, doi: 10.1109/TMI.2019.2900516.
- [8] Ronneberger, Olaf; Fischer, Philipp; Brox, Thomas (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". arXiv:1505.04597 [cs.CV].
- [9] Mingxing Tan, Quoc V. Le Thomas (2019). "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". arXiv:1905.11946 [cs.LG].
- [10] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
- [11] Towards Data Science, <https://towardsdatascience.com/image-segmentation-with-six-lines-of-code-acb870a462e8>
- [12] Wikipedia, https://en.wikipedia.org/wiki/Medical_imaging, accessed 03/10/2020.
- [13] Wikipedia, https://en.wikipedia.org/wiki/File:Ultrasound_of_human_heart_apical_4-chamber_view.gif, accessed 03/10/2020.
- [14] Wikipedia, https://en.wikipedia.org/wiki/Binary_image/media/File:Neighborhood_watch_bw.png, accessed 03/10/2020.

- [15] Wikipedia, <https://en.wikipedia.org/wiki/Grayscale>/media/File:Grayscale_8bits_palette_sample_image.png, accessed 03/10/2020.
- [16] Santé sur le net, <https://www.sante-sur-le-net.com/maladies/cardio/generalites-coeur/>, accessed 03/10/2020.
- [17] Wikipedia, <https://en.wikipedia.org/wiki/Heart>/media/File:Diagram__the_human_heart_(cropped).svg, accessed 03/10/2020.
- [18] Wikipedia, <https://fr.wikipedia.org/wiki/TensorFlow>/media/Fichier:TensorFlowLogo.svg, accessed 03/10/2020.
- [19] SimpleITK, <https://simpleitk.org/>, accessed 03/10/2020.
- [20] ITK-SNAP, <http://www.itksnap.org/pmwiki/pmwiki.php>, accessed 03/10/2020.
- [21] Tensorflow, <https://www.tensorflow.org/guide/keras?hl=fr>, accessed 03/10/2020.