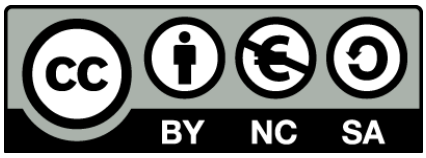
UNIVERSITAT DE BARCELONA

# Optimization of neural networks for deep learning and applications to CT image segmentation

Giuseppe Pezzano





Department of Mathematics
UNIVERSITAT DE BARCELONA

Department of Digital Health
EURECAT TECHNOLOGY CENTER

# Optimization of neural networks for deep learning and applications to CT image segmentation

By

GIUSEPPE PEZZANO

A dissertation submitted to Universitat de Barcelona in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the Faculty of Mathematics, within the course of Computer Science.

28 JULY 2023

**Thesis directed by:** PROF. PETIA RADEVA director

DR. VICENT RIBAS co-director

*To my family, backbone of my world*
*To Elisabetta, cornerstone of my life*

*Alla mia famiglia, colonna portante del mio mondo*
*A Elisabetta, pietra angolare della mia vita*

# Abstract

During the last few years, AI development in deep learning has been going so fast that even important researchers, politicians, and entrepreneurs are signing petitions to try to slow it down. The newest methods for natural language processing and image generation are achieving results so unbelievable that people are seriously starting to think they can be dangerous for society. In reality, they are not dangerous (at the moment) even if we have to admit we reached a point where we have no more control over the flux of data inside the deep networks. It is impossible to open a modern deep neural network and interpret how it processes the information and, in many cases, explain how or why it gives back that particular result. One of the goals of this doctoral work has been to study the behavior of weights in convolutional neural networks and in transformers. We hereby present a work that demonstrates how to invert $3 \times 3$ convolutions after training a neural network able to learn how to classify images, with the future aim of having precisely invertible convolutional neural networks. We demonstrate that a simple network can learn to classify images on an open-source dataset without loss in accuracy, with respect to a non-invertible one. All that with the ability to reconstruct the original image without detectable error (on 8-bit images) in up to 20 convolutions stacked in a row. We present a thorough comparison between our method and the standard. We tested the performances of the five most used transformers for image classification on an open-source dataset. Studying the embedded matrices, we have been able to provide two criteria that can help transformers learn with a training time reduction of up to 30% and with no impact on classification accuracy.

The evolution of deep learning techniques is also touching the field of digital health. With tens of thousands of new start-ups and more than 1B$ of investments only in the last year, this field is growing rapidly and promising to revolutionize healthcare. In this thesis, we present several neural networks for the segmentation of lungs, lung nodules, and areas affected by pneumonia induced by COVID-19, in chest CT scans. The architectures we used are all residual convolutional neural networks inspired by UNet and Inception. We customized them with novel loss functions and layers studied to achieve high performances on these particular applications. The errors on the surface of nodule segmentation masks are not over $1\,mm$ in more than 99% of the cases. Our algorithm for COVID-19 lesion detection has a specificity of 100% and overall accuracy of $97.1 \pm 1.0\%$. In general, it surpasses the state-of-the-art in all the considered statistics, using UNet as a benchmark. Combining these with other algorithms able to detect and predict lung cancer, the whole work was presented in a European innovation program and judged of high interest by worldwide experts.

With this work, we set the basis for the future development of better AI tools in healthcare and scientific investigation into the fundamentals of deep learning.

## Author's declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

## About the author

My interest in mathematics and informatics started when I was a teenage student, among the desks of Leonardo da Vinci high school in Reggio Calabria - Italy. I proved my skills by classifying for the national mathematics olympiad and it was very clear in my mind which path to follow afterward. I enrolled at the University of Pisa and, some years after, I got my Bachelor's degree in physics. My passion for informatics and the whole world of programming, including digital technologies and hardware, started there, in the same years when artificial intelligence was undergoing its most profound revolution.

It was during my master's in physics, while I was attending lectures about pattern recognition in Heidelberg - Germany, that I came across machine learning. It was introduced to me as a very simple kind of software that can learn by examples and can be applied to any kind of problem, only changing the data samples. I have been thunderstruck.

While I was doing my traineeship at the German cancer research center (DKFZ) making computer simulations on hadron therapy for cancer treatment, I dived into the world of computer science and I understood my career was likely going to pivot in that direction. After getting my master's degree in physics, I studied as a self-taught computer scientist with the only focus of mastering, one day, machine learning. Later, I started my Ph.D. at the University of Barcelona, employed at the Eurecat technological center. My background in mathematics and statistics, joined with my extremely positive attitude towards all the new technologies in computer science, helped me to successfully reach the end of my Ph.D. The pandemic of COVID-19 took away from me important opportunities for spending some time abroad or presenting at conferences, for enriching my experience, although I sincerely feel lucky it is only that.

Looking back now at my work in the past four years, I feel satisfied with what I achieved, with four articles, a startup project, and important responsibilities inside the department of Digital health at Eurecat. But, I could not obtain these results without the support and trust I received from my supervisors, Prof. Petia Radeva and Dr. Vicent Ribas, and my unit director Dr. Felip Miralles. They first demanded me from Germany and then guided me, with commitment and (a lot of) patience, on the doctoral path. For that, they will forever have my undying gratitude and respect.

# Acknowledgement

I especially want to thank my supervisors Petia Radeva and Vicent Ribas, for their patience, help, support, and a very long list of other things.

I also thank Felip Miralles, Maria Eugenia Fuenmayor, Universitat de Barcelona, and the Catalan Government, that supported this work also through the funding grant ACCIÓ-Eurecat and Eurecat's *Vicente López* Ph.D. grant program.

A special thank needs to go to all my colleagues at Eurecat and in particular to Xavier Rafael-Palou, who always helped me, especially in my first steps at Eurecat and in the DeepLung project.

Last but not least, I thank Eduard Solér and Andreu Antolín for their dedication to Optimal Lung and for teaching me how to look at things from another perspective.

# Table of Contents


**Page**

**I Fundamental Research in Machine Learning** **1**

**1 Introduction** **3**
1.1 Context . . . 5
1.2 Goals and Objectives . . . 7
1.3 Contributions . . . 9
1.4 Memory Organization . . . 9

**2 Background** **11**
2.1 Convolutional Neural Networks . . . 11
2.1.1 AlexNet . . . 12
2.1.2 ResNet . . . 12
2.1.3 Inception . . . 13
2.1.4 U-Net . . . 15
2.2 Transformers . . . 16
2.3 Vision Transformers . . . 17
2.3.1 Vision Transformer . . . 17
2.3.2 Data-efficient image Transformer . . . 17
2.3.3 Bidirectional Encoder representation from image Transformers . . . 17
2.3.4 Shifted Windows Transformer . . . 18
2.3.5 Cross-Shaped Windows Transformer . . . 18

**3 Invertible CNN** **21**
3.1 Introduction to CNNs . . . 21
3.1.1 State of the Art on invertible CNNs . . . 21
3.1.2 Aim of invertible convolution . . . 22
3.2 Methods for invertible CNN . . . 23
3.2.3 Reformulation of the convolution . . . 24
3.2.5 Proof of Invertibility . . . 26

3.2.6 Applications . . . 28
3.2.7 Test Architectures . . . 29
3.2.8 Metrics . . . 30
3.3 Results for invertible CNN . . . 31
3.3.1 Classification Results . . . 31
3.3.2 Kernel properties evolution . . . 31
3.3.3 Reconstruction Results . . . 33
3.4 Discussions on invertible CNN . . . 34
3.5 Conclusions on invertible CNN . . . 36

**4 The learning process of visual transformers 37**
4.1 Introduction to Transformers . . . 37
4.2 State of the art on Transformer Optimization . . . 38
4.2.1 Works on Transformers Optimization . . . 38
4.3 Methods for Transformer Optimization . . . 39
4.3.1 Attention . . . 39
4.3.2 Self-attention . . . 40
4.4 Criteria for transformers evolution analysis . . . 40
4.5 Validation of Transformer Optimization . . . 41
4.5.1 Dataset . . . 41
4.5.2 Implementation setting . . . 41
4.5.3 Results and analysis . . . 41
4.6 Conclusions on Transformer Optimization . . . 44

**II Deep Learning in Digital Health 45**

**5 Digital Health 47**
5.1 AI for Lung Diseases . . . 50

**6 Lung nodule segmentation 53**
6.1 Introduction to Lung Nodule Segmentation . . . 53
6.1.1 Aim of Lung Nodule Segmentation . . . 54
6.1.2 Relevant works in Lung Nodule Segmentation . . . 55
6.2 Methods for Lung Nodule Segmentation . . . 57
6.2.1 Model Architecture . . . 57
6.2.2 Loss function . . . 59
6.2.3 Fit to the Segmentation Mask . . . 60
6.2.4 Dataset . . . 61
6.2.5 Evaluation Statistics . . . 62

6.2.6 Implementation Details . . . 63
6.3 Results for Lung Nodule Segmentation . . . 64
6.4 Discussions on Lung Nodule Segmentation . . . 65
6.4.1 Overall Performance . . . 65
6.4.2 Ablation studies . . . 67
6.4.3 Database Consistency . . . 69
6.4.4 Comparison to the State of The Art . . . 70
6.5 Conclusions on Lung Nodule Segmentation . . . 72

**7 COVID-19 Lesion Segmentation 73**
7.1 Introduction to COVID-19 Lesion Segmentation . . . 73
7.1.1 Aim of COVID-19 Lesion Segmentation . . . 74
7.1.2 Relevant Work on COVID-19 Detection and Segmentation . . . 75
7.2 Methods for COVID-19 Pipeline . . . 78
7.2.1 Model Architecture . . . 78
7.2.2 Implementation Details . . . 80
7.2.3 Post-Processing . . . 82
7.2.4 Image Dataset . . . 82
7.2.5 Evaluation Statistics . . . 83
7.3 Results for COVID-19 Pipeline . . . 84
7.3.1 Lung Segmentation . . . 84
7.3.2 COVID-19 lesion Segmentation . . . 84
7.3.3 COVID-19 detection . . . 85
7.4 Discussions on COVID-19 Pipeline . . . 85
7.4.1 Overall Performance . . . 85
7.4.2 Comparison to the State of the Art . . . 86
7.5 Conclusions on COVID-19 Pipeline . . . 91

**8 Optimal Lung project 93**
8.1 Current Practice . . . 94
8.2 Our proposal . . . 95
8.3 Results and Statistics . . . 96
8.3.1 Detection . . . 96
8.3.2 Segmentation . . . 97
8.3.3 Prediction . . . 98
8.4 Evidence . . . 98
8.5 Impact of the solution . . . 99
8.6 Wild Card . . . 100

**9 Conclusions** **103**
9.1 Future Lines . . . . . 104

**List of Tables** **107**

**List of Figures** **109**

**A Appendix A** **111**
A.1 Lemmas on Convolution . . . . . 111

**B Appendix B** **113**
B.1 More about Database Consistency . . . . . 113
B.2 More Interesting Cases . . . . . 113

**Bibliography** **117**

# PART I

## Fundamental Research in Machine Learning

*The works and customs of mankind do not seem to be very suitable material to which to apply* ***scientific induction***

ALAN TURING

CHAPTER 1

# INTRODUCTION

It was 1959 when Arthur Samuel first coined the term *Machine Learning*. This happened on the wave of the experiments of Frank Rosenblatt who, just a year before, developed the very first neural network, the "Frankenstein Monster Designed by the Navy that thinks" as it has been described by the American newspapers. He called it *Perceptron*.

At the time, the idea of a computer that can think was not novel. The concept of *Artificial Intelligence* was spreading rapidly after Alan Turing provided us with the modern definition of an intelligent machine in 1950, i.e. one that could pass his test, the famous *Imitation Game*. Although the technology of the time was insufficient for supporting this branch of research, machine learning was all but dead, but just waiting for its time to come. That period is called *AI winter*. The 1970s witnessed the launch on the market of the *personal computer*, which relentlessly changed the history of humanity. Nevertheless, we had to wait until the 1980s, when computers started to be sold on a massive scale at affordable prices (Figure 1.1), for the world of the scientific community to rediscover their interest in artificial intelligence. *Expert Systems* was the first successful form of a machine emulating the decision-making ability of a human being. They are based on

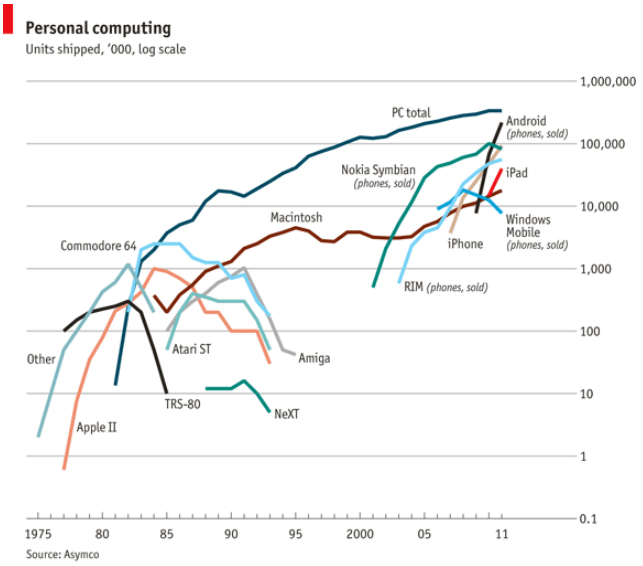



FIGURE 1.1. Thousands of computer units shipped per year in logarithmic scale. Source: The Economist.

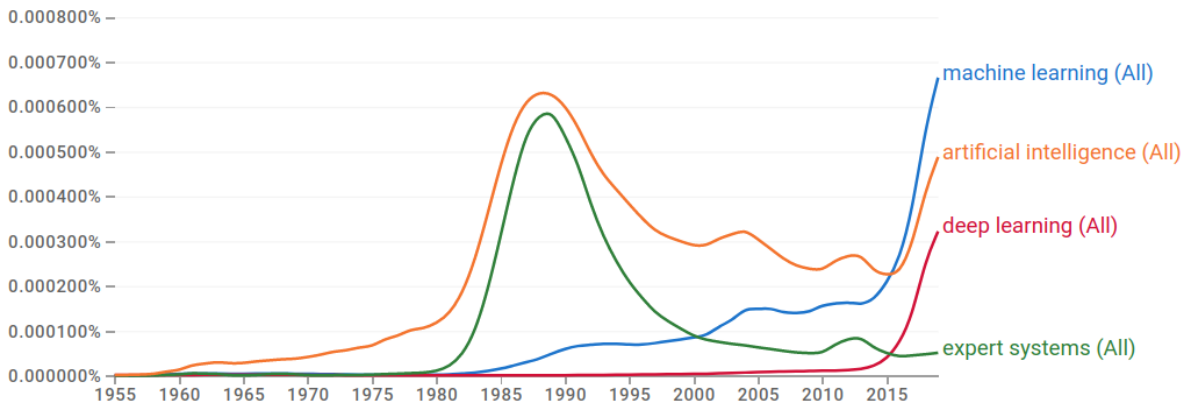



FIGURE 1.2. Usage in time of the wordings: Artificial Intelligence, Machine Learning, Deep Learning, Expert System. Source: Google Books Ngram Viewer.

*if-then rules*, which is their selling point but also their main limitation. Indeed, they are very consuming in terms of programming time and cannot be applied to another problem without being largely re-programmed. Their success has been enormous but they have been largely forgotten with equal rapidity (Figure 1.2).

One of the sparks that re-ignited the motor of machine learning can be conferred to the *Harmonium*. First designed by Paul Smolensky in 1986, it became very notorious in the 2000s from the work of Geoffrey Hinton and colleagues. This kind of networks are mostly known today under the name of *Restricted Boltzmann machines*. The introduction of Convolutional Neural Networks (CNN) played a leading role in the transformation of machine learning because it allows to process larger images and to go deeper, using fewer computational resources. But, the world scientific community was not frowned upon by this technology or neural networks in general. Yann LeCun himself declared that he removed the word “neural” from his greatest invention because this could give him more chances of publishing in journals and international conferences.

In the first 2000s, the table turned and new machine learning methods started to substitute the expert systems, but the real turning point arrived in 2012. At the annual Conference on Neural Information Processing Systems (NeurIPS), Geoffrey Hinton, together with two of his students, presented their new CNN, called *AlexNet*. Ironically, NeurIPS was the same conference that some years before rejected Hinton’s work because they accepted another article on neural networks and they thought it would be unseemly to accept two in the same year. At this same conference, Hinton’s team also sold to Google a company consisting only of a social capital *of a few dollars* and a very minimal website, offering nothing, but the name “DNN-research”. The whole for 44 million dollars! Actually, they were selling much more than that. The buyers wanted to be the first ones to implement and use Hinton’s newest AI technology, *Deep Learning*, at a large scale.

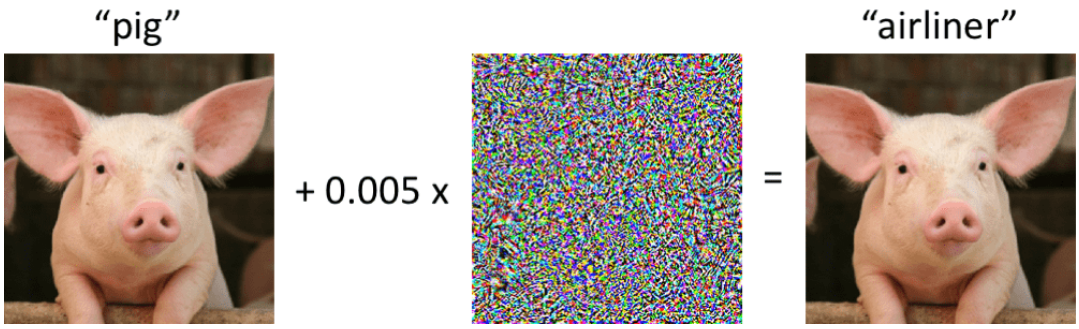



FIGURE 1.3. Adversarial example on EfficientNet. Source: Gradient Science.

## 1.1 Context

Forbes magazine recently stated that the global Machine Learning (ML) market, valued at 1.58 Billion $ in 2017, is expected to reach 20.8 Billion $ in 2024 [77], and the increment of job proposals seeking Artificial Intelligence (AI) experts is around 74% every year, on average [60]. Those are, in numbers, the equivalent of trust that the world is giving to this innovative technology. In fact, in these times, we interact with AI software on a daily basis, whether they are the dictation algorithms installed on the keyboard of our smartphones, the face recognition tools of social networks, or the algorithm helping us decide which movie to watch tonight, or the driver assistant facilitating us to drive our brand-new car. The vast majority of those algorithms are Neural Networks (NNs). However, we do not comprehend everything about the way NNs indeed work. Too many times we read they are black boxes and that the user does not have control over their data embedding. This is ironic, given the fact that machine learning algorithms are mathematically rigorous and we know at each instant the exact state of the system[1]. In reality, machine learning algorithms are far from perfect. They run with a series of unproven approximations to be globally allowed, especially in large models. They approximate the loss function to the first order, linearly, neglecting all the higher-order terms determining loss curvature. Moreover, they calculate the gradients with partial derivatives, evaluating the contribution of each feature in isolation and ignoring their inter-dependencies [102]. These are also the reasons why neural networks severely suffer *adversarial examples* [40] and can be induced into mistakes with robust precision[2] [85], as in the example in Figure 1.3. These approximations generate further issues also into the core process of Deep Learning algorithms: back-propagation. As this algorithm advances backward, from the output layer towards the input layer, the gradients could often get smaller and smaller and approach zero which eventually leaves the weights of the initial

[1] Schrödinger explained to us this is not possible with quantum states. What will happen when we will implement machine learning on quantum computers?

[2] Pytorch deep learning library offers an automatic method for analytical adversarial examples generation.

or lower layers nearly unchanged. As a result, the gradient descent never converges to the optimum, causing the network to stop learning. This is recognized as the *vanishing gradients problem*, it affects mostly the neural networks with many layers and the recurrent neural networks. On the contrary, in some cases, the gradients keep on getting larger and larger as the back-propagation algorithm progresses. This, in turn, causes very large weight updates and induces the gradient descent to diverge. This is known as the *exploding gradients problem* [45].

FIGURE 1.4. Marble statue using a computer. Generated with Stable Diffusion.

Both problems are caused by the fact that, in an $n$-layers neural network, the updates on the weights are computed from the gradients and all the changes are propagated exponentially. Many works claimed to have been able to overcome those problems, when in fact, they only mitigate them [92]. Our troubles in deciphering neural networks also pass through our limits in abstraction, and in working in high-dimensional spaces, such as the embedding spaces of the majority of neural networks, where the loss functions lay. All the attempts to create smaller latent spaces containing completely the information of the datasets have not collected adequate results whatsoever and are still largely under development, an example is Variational Auto-Encoders (VAE) [122]. All these pitfalls cause the results to be sometimes below expectation, and it is particularly evident in generative models. The technology in image generation is improving constantly. However, we still do not manage to identify a solution regarding the problems of perspective and structure. As people noticed in the most recent creations of diffusion models, watching carefully at the images, some parts of them appear totally unnatural. Those are usually the extremities and the eyes of the characters appearing in the images, but also regular-shaped objects sketched in a cubist perspective. The lack of a three-dimensional vision of the neural network is one of the causes, but not the only one. Indeed, generative models are capable to reconstruct precisely an object or a figure that they already saw but are unable to link two objects. Let us consider for example the sculpture in Figure 1.4. The details are impressive, but the hand that is typing on the keyboard is deformed, as is the keyboard itself. While the eyes of the statue are staring into space instead of watching the computer. Or in the figure at the beginning of Part I, the feet and the hand of the jockey are not displayed and the saddle is fused around his/her waist. The reason is conceptually very simple. Generative models are extremely good at detecting patterns

(or objects) and guessing what could appear around them until they have already seen something similar. But when there are two objects belonging to different worlds (like a marble statue and a computer, or the example in Figure 1.5), the neural networks do not have experience with their interaction and, at this point, what they do is like guessing.

## 1.2 Goals and Objectives

Despite the excellent results obtained so far by machine learning, there are many things to grasp and significant problems still need to be fixed. One of the core issues to solve is the one related to learning time and efficiency. The lack of control over gradient-related problems and over the generalization of the datasets (never big enough! [49]) leads to the need of finding alternative ways for increasing the efficiency of learning, without increasing exponentially the size of the dataset. This is wired to the concept of *explainability*, i.e. the capacity of a black-box model to be explained using external resources [43] (Figure 1.6).

This is one of the crucial steps that could allow the further spread of AI into our society. In fact, one of the limits of current methods is that sometimes we hardly explain why a neural network has made a certain decision. This generates ethical and legal problems when it comes to algorithms adopting decisions that can influence people's lives [7], for example, a neural network that defines if a person represents a good payer and can or cannot receive a loan, or one that forecasts which is the level of risk for an individual of suffering from cancer. Suppose those, influenced by past experiences, start to discriminate against people based on gender, race, culture, or other features unrelated to the problem. In that case, we must avoid it, or alternatively, recognize it and correct it. As a matter of principle, if we were able to understand which part of the NN contributes the most to the misbehavior, we could isolate it and inhibit it. In the same way, we could detect which part of the NN contributes the most to learning and optimize the results. Our first goal is to develop a method to rate the contribution to the learning of each layer, in the most common Transformers. Studying the embedded matrices, we aim at identifying a method for optimizing transformer learning, by reducing training time while increasing accuracy. This does

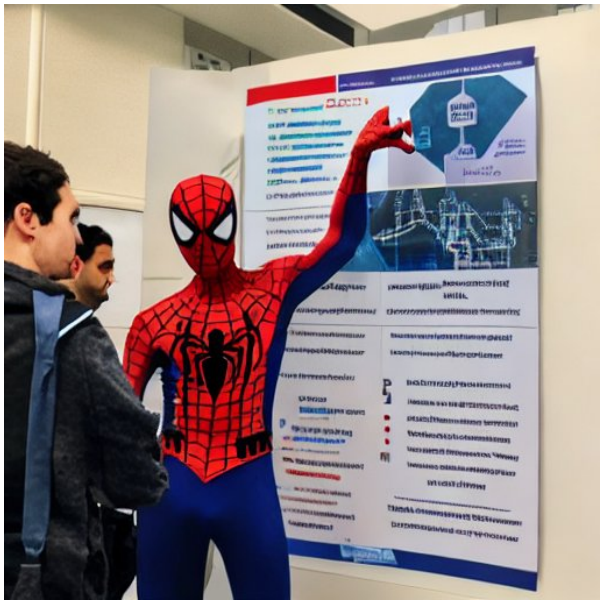


FIGURE 1.5. Spiderman presenting his poster at data science conference.
Generated with Stable Diffusion.

not mean we will be eventually capable of explaining all the results. But being able to positively influence learning, constitutes a step forward to the solution of the problem.

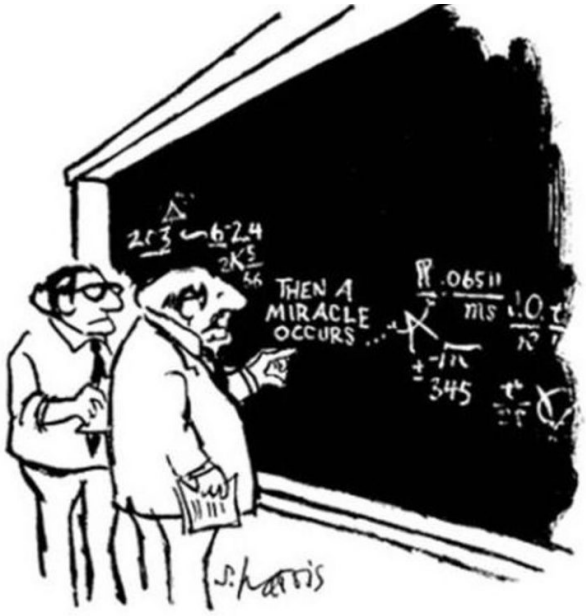



FIGURE 1.6. Example of a non-explicable process. Then a miracle occurs (Harris, 2010).

Opening a deep neural network is incredibly difficult or sometimes impossible. Moreover, understanding and explaining how it elaborates every possible type of input represents an extremely complex task. Especially in computer vision, the space of inputs is gigantic and when we process such images with a non-linear network of several million parameters, the real comprehension of its workflow is definitely out of our reach. Keeping in mind that is not required to explain the whole logic behind the black box but only the reasons for the choice of a particular instance [43], our second objective is to explain the way neural networks learn (or *think*) by working on the inverse problem. We believe that, by being able to reconstruct the original space of inputs generating a certain precise output, the dimension of the problem can be reduced, allowing its, even partial, solution. In order to do so, invertible networks are required. So far, invertible convolutional neural networks have given only approximate results, because convolution is a generally non-invertible operation and because of the limits imposed by the machines when it comes to calculating the inverse of very large matrices. The goal is to find a mathematical formulation of precisely invertible convolutional kernels that can still learn during training, avoiding the reduction of the number of free parameters inside the convolutional kernel as much as possible.

## 1.3 Contributions

This work contributes to research in the field of fundamental machine learning and digital health with four articles. Two of these have already been published and belong to the applications, and two are related to the part of fundamental research and are under revision.

[Published - 34 citations] CoLe-CNN: Context-learning convolutional neural network with adaptive loss function for lung nodule segmentation

[Published - 10 citations] CoLe-CNN+: Context learning - Convolutional neural network for COVID-19-Ground-Glass-Opacities detection and segmentation

[In review] A CNN with rigorously invertible convolutions

[Sent] Framework to analyze the learning process of visual transformers

In our work on invertible convolutional neural networks, We employ a novel formal method for inverting convolution kernels and training them for classification purposes on the most successful type of neural network: CNNs. This has been tested on an open dataset with encouraging results. This work is presented in chapter 3. Our following work belongs to the same field and aims at optimizing transformer learning for image recognition. Our method has been tested on five of the most commonly used architectures with state-of-the-art results in testing and performances reasonably increased. This work is presented in chapter 4.

The other two works belong to machine learning in healthcare and both approach the problem of image segmentation in Computed Tomography (CT) scans. The first one aims to the development of a neural network for lung nodules (cancerous and not) segmentation. Our proposal achieves state-of-the-art estimation of the volume of a nodule, with errors on the surface that are not over $1mm$ in more than 99% of the cases. A complete summary is presented in chapter 6. Due to the high impact of the topic, we also worked on a solution to detect and segment the area of the lungs attacked by COVID-19-induced pneumonia. Also in this case the results have been commendable. The work has been published and is entirely included in this thesis in chapter 7. As a demonstration of their goodness, these two works [89] [90] received a total of 44 citations[3] in less than two years.

## 1.4 Memory Organization

This thesis collects the scientific work on deep learning during this doctoral study and it is divided into two parts: (i) fundamental research in machine learning and (ii) deep learning in digital health. This first part gathers a work on transformers optimization and one on convolutional neural networks, presenting a novel method for the analytical inversion of convolutions for

[3] Number of citations taken from Google Scholar on date 24/04/23.

deep learning purposes. The second part gives a summary of AI in healthcare (so-called digital health) and contains two works on the segmentation of clinical objects and areas of interest in three-dimensional chest CT scans. Finally, we report our experience in a field related to digital health, but from a different perspective, the one of the market.

*I think the way we're doing computer vision is just **wrong**.*
*It works better than anything else at present*
*but that doesn't mean it's right.*

GEOFFREY HINTON

CHAPTER 2

# BACKGROUND

Finding a way to understand and imitate the behavior of the brain has always been the Holy Grail of the scientific community and the source of inspiration for computer scientists working on artificial intelligence. In fact, Artificial Neural Networks (ANN) were conceived to be the computational counterpart of biological neural networks and, to those, they owe their name. This does not imply that ANNs work in the same way as our brain does. But one may see the analogy between, for example, the activation function and a synapse or a hidden layer and a biological neuron. From this perspective, Convolutional Neural Networks (CNN) are not an exception. In this section, we will introduce the concepts that influenced this work the most.

## 2.1 Convolutional Neural Networks

In the 1960s was demonstrated that a cat's brain contains neurons that react individually to small regions of the visual field [55] (the *receptive field*). Those neurons are responsible to process the images, shifting the location, inside the cortex. The pieces of information are then passed to other neurons that join them and form a complete map of visual space. This is roughly the idea on which convolutional neural networks are based. The first CNN was developed by Fukushima and was called the *Neocognitron* [35]. It already included the concepts of convolutional layers and downsampling but it had issues due to the learning method because backpropagation, the modern standard, did not exist yet. In 1989, appeared one of the pioneers in this field, it is LeNet-5 [66]. It is a 7-layer convolutional neural network designed to classify hand-written numbers in images of size $32 \times 32$. It already included most of the modern features of modern CNNs and, even though it had limited applications, the research achieved great success. This time the limit was set by computational resources and we have to wait until 2012 to see the very first successful CNN on a large scale.

### 2.1.1 AlexNet

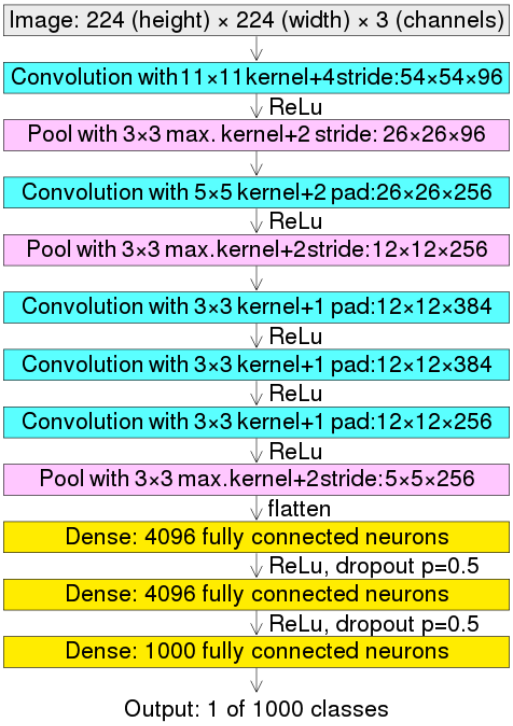



FIGURE 2.1. AlexNet layers details. Source: Wikipedia.

AlexNet [64] is a Convolutional Neural Network designed by Alex Krizhevsky, under the supervision of Geoffrey Hinton. Composed of five convolutional layers and three fully-connected layers, it has been developed with the direct purpose of minimizing overfitting. The main innovations introduced by this work are the combination of the Rectified Linear Unit activation function (ReLU) and the dropout regularization method, in fully connected layers (Figure 2.1). ReLU is a positive function that sends to zero all the negative values namely:

$$f(x) = max(0, x)$$

The authors demonstrated that deep CNNs with ReLUs train several times faster than their equivalents with other activation units, because of their non-saturating properties, without requiring normalization. Dropout is a technique consisting of setting to zero the output of every hidden neuron with a variable probability, usually of 50%. Those neurons which are "dropped out" do not contribute to the forward pass and do not participate in backpropagation. In practice, the neural network assumes a different architecture every time an input is presented, but it always shares the same weights.

As we said before, AlexNet is not the first CNN and neither the first neural network running on a Graphics Processing Unit (GPU), but it became famous for two main reasons: it won four machine learning competitions between 2011 and 2012, and it was the first prototype of the modern deep learning algorithm. It represented a revolution in the whole machine learning field and it has been cited over 100000 times.

### 2.1.2 ResNet

How could we increase the level of abstraction of a convolutional neural network to make it more precise? Everyone's first guess to this question would be to add more layers so we can extract more information. This answer is not wrong, but it was not applicable in the case of AlexNet because, as we said in Chapter 1, deep neural networks with many layers suffer from gradient-related problems. So, when a deep neural network starts converging, it is exposed to a degradation problem. Consequently to the increase in the depth of the network, accuracy gets

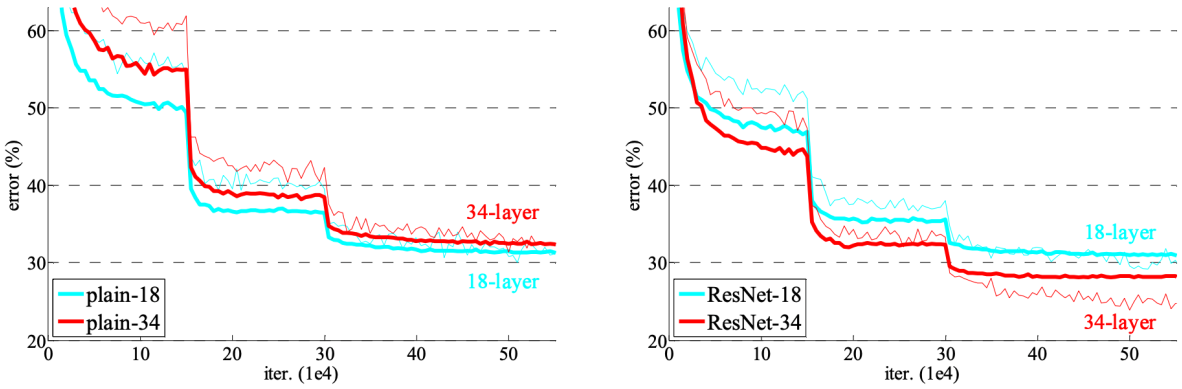



FIGURE 2.2. On the left, two plain networks (not using residuals) of 18 and 34 layers. On the right, ResNets of 18 and 34 layers, with the same exact architecture as the left ones. For each of them, the training curve (in bold) and the validation (in thin) are displayed. All the networks have been trained on ImageNet. [48]

saturated and degrades rapidly. Surprisingly, that degradation is not induced by overfitting, and the more layers are added, the higher the error (Figure 2.2, left).

In 2015, researchers at Microsoft found a partial solution to this problem and developed a *Residual neural Network* (ResNet) (He *et al.,* [48]), with a paper cited more than 150 000 times. The authors explicitly reformulate some of the layers to be learning residual functions, collecting information both from the previous layer and from three layers before. Intuitively, this helps the network to "keep the focus" on the important patterns, instead of learning unreferenced functions. This reformulation mitigates the phenomena of degradation, as can be seen in (Figure 2.2, right).

There are several versions of ResNet, usually coupled with the number of layers of the network. The first ones published were ResNet-18, 34, 50, 101, and 152, with a number of parameters that ranges between 11M to 58M, roughly. And their applications encompass all the fields of computer vision.

### 2.1.3 Inception

In a series of publications, from 2015 to 2017, researchers at Google proposed a new backbone for residual connections in convolutional neural networks [107] [108] [109] They called this method *Inception*, as a tribute to the concept of Network In Network [71] and to the famous movie directed by Christopher Nolan. Inception layers are based on the idea of processing in parallel the same input through a series (usually four) of convolutions of different sizes, and then to concatenate the result before feeding it to the next layer (examples in Figure 2.3). The benefit of this architecture is that allows to increase the number of units[1] at each level, without a significant increase of

[1] A unit, in this case, refers to a branch of the inception layer. In other words, the number of units measures the width of the layer.

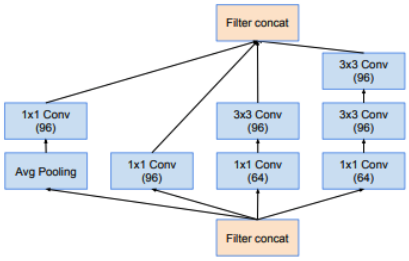



Figure 4. The schema for 35 × 35 grid modules of the pure Inception-v4 network. This is the Inception-A block of Figure 9.

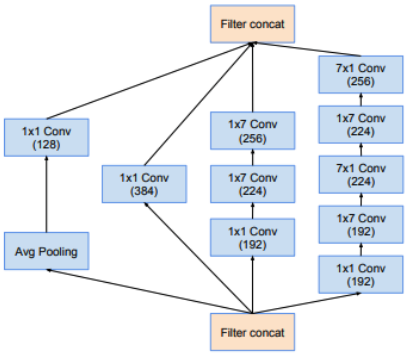



Figure 5. The schema for 17 × 17 grid modules of the pure Inception-v4 network. This is the Inception-B block of Figure 9.

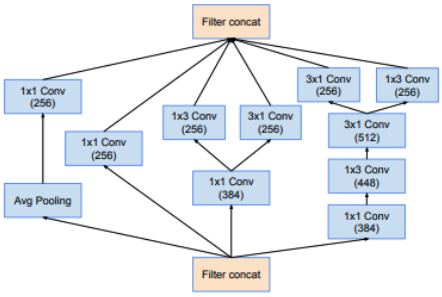



Figure 6. The schema for 8 × 8 grid modules of the pure Inception-v4 network. This is the Inception-C block of Figure 9.

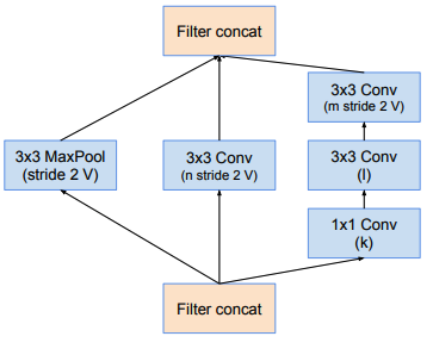



Figure 7. The schema for 35 × 35 to 17 × 17 reduction module. Different variants of this blocks (with various number of filters) are used in Figure 9, and 15 in each of the new Inception(-v4, -ResNet-v1, -ResNet-v2) variants presented in this paper. The $k$, $l$, $m$, $n$ numbers represent filter bank sizes which can be looked up in Table 1.

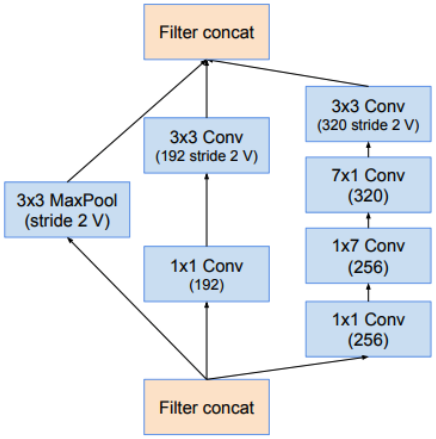



Figure 8. The schema for 17 × 17 to 8 × 8 grid-reduction module. This is the reduction module used by the pure Inception-v4 network in Figure 9.

FIGURE 2.3. Five examples of Inception layers used as backbone of Inception-v4, and Inception-ResNet-v1 and v2. [108]

the computational complexity. Because the results of each unit are concatenated, the number of filters should grow up geometrically, but it can be kept under control using $1 \times 1$ convolutions. They have the function of reducing the dimension of the filters and apply a weight, before feeding more computationally expensive operations. The most interesting aspect is that these particular kind of layers use an approach conceptually similar to the one explained at the beginning of Section 2.1. They work on the intuition that visual information should be processed at various levels, allowing to extract distinct patterns at different scales simultaneously. During third-party experiments [12], it came out that Inception networks were able to obtain better results than ResNets, on ImageNet dataset, with a significantly lower number of parameters and operations per cycle (Figure 2.4).

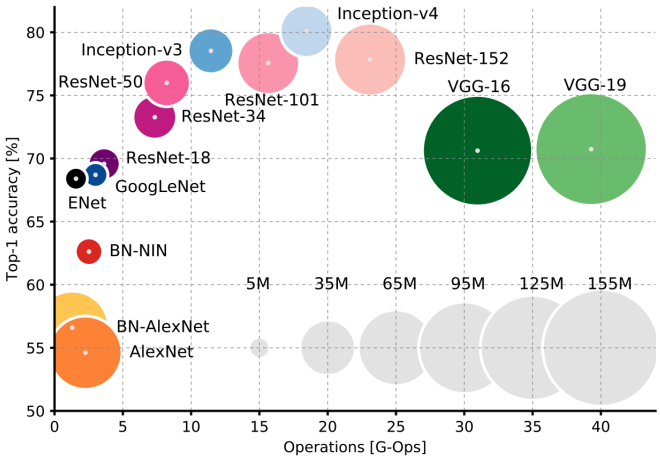



FIGURE 2.4. Top-1 accuracy on ImageNet vs. number of operations per cycle, for several CNNs. The size of the blobs indicates the number of parameters. [12]

### 2.1.4 U-Net

Another kind of residual convolutional neural network that gained vast success in the field of computer vision is U-Net [67]. It is a technology developed in 2014, but is still largely in use. Based on an encoder-decoder structure, U-Net is composed of a series of $3 \times 3$ convolutions organized in four levels of depth (Figure 2.5). The height and width of the input are scaled by a factor of 2, at each level of depth. The encoder is the first half of the network, which goes from the first layer to the deepest level of embedding, called the bridge. The bridge contains the same number of parameters of the input, because the size reduction in width and height is compensated by a corresponding increase in the third dimension, the so-called *channel*. The decoder is the counterpart of the encoder, it takes the information produced by the encoder and builds an output of the same size as the input. The main difference with previous neural networks is that the decoder does not have one single input. The first input of the decoder is the bridge. Then, at every level of depth, the network takes the outputs of the layers of equal dimension in the encoder and the decoder, stacks them, and processes them simultaneously. This is its main innovation and it is called *residual connection*. As it is easy to comprehend, the amount of information is drastically reduced at each step of pooling. Therefore, these connections are helpful in delivering to the decoder the information that has been lost. But this is not the only reason. In

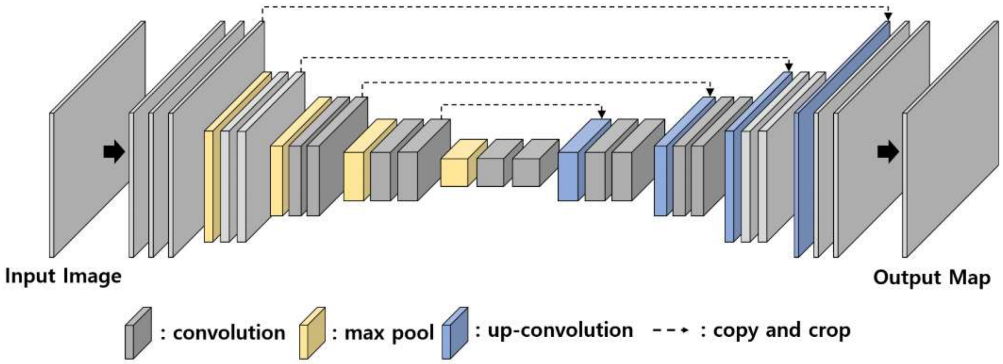



FIGURE 2.5. Example of U-Net architecture. [67]

fact, residual connections help to keep under control the phenomena of gradient explosion and vanishing gradients. U-Net is a precursor of a long list of residual neural networks that have been, and still are used in all the fields of image processing. This architecture has been widely used in medical imaging analysis because, thanks to its lightweight and robustness, it can be applied also to 3D images, without requiring excessive resources. U-Net architecture is also used as the backbone of new models, like diffusion.

## 2.2 Transformers

Another trend that attracted our attention, which we decided to investigate more on, is *Transformers*. They are based on the mechanism of attention or self-attention, differentially attributing a weight to all the distinct parts of the input, explained in detail in Section 4.3.1 and 4.3.2. It is frequently employed in natural language processing, but it has also demonstrated its value in image object detection. Similarly to recurrent neural networks (RNNs), transformers have been developed to elaborate sequential input data, such as videos or strings of text, but process the whole input at the same time. The particular feature of these neural networks is that, unlike CNNs, they preserve information about the position of the objects in the input data. Transformers can possess diverse architectures depending on the purpose and the typology of data they work on. A very well-known paper called "Attention is All You Need" [114], explains the basic structure of a vision transformer. As shown in Figure 2.6, the input is divided into windows that are then given to the network in form of a sequence. Thanks to the attention layers, the transformer can estimate the correlation between each element of the sequence, being able to understand which parts of the image are influential or not in the classification. The eventual classification is performed by a Multi-Layer Perceptron (MLP) or can be carried out with any series of fully connected layers. Transformers are currently used for interpreting and generating text as well. The most successful model has been GPT-4 thus far, an open-source model by Open AI[2].

[2]Here there are some examples of what GPT-4 is capable of doing.

## 2.3 Vision Transformers

The gold standard in image-related tasks has been set by Convolutional Neural Networks for a long time until vision transformers appeared.

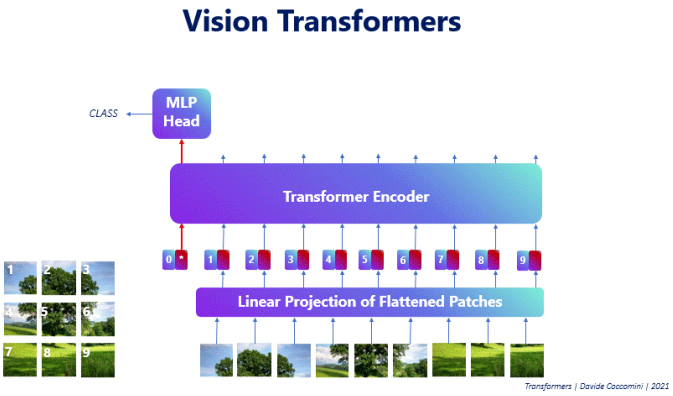



FIGURE 2.6. Vision Transformer example. Source: Wikipedia.

### 2.3.1 Vision Transformer

This transformer was presented by Dosovitskiy et al. [24] in 2020. Vision Transformer (ViT) splits the image into patches and applies a linear embedding to each patch to obtain a vector (Figure 2.6), simulating the input of the original transformer. Differently from previous works, ViT does not have image-specific inductive biases introduced manually. Instead, they are all learned during training. In addition, ViT has only the encoder part and implements a class token, which is also a learnable embedding, whose final state is used to classify each image. Its reported accuracy is 90.45% on ImageNet.

### 2.3.2 Data-efficient image Transformer

Data-efficient image Transformer (DeiT) was presented in December 2020 [112] and tries to improve ViT performance by reducing learning time. To achieve this, it implements a teacher-student strategy. The student, using a distillation token, aims to reproduce the label indicated by the teacher, which is typically a CNN. Except for this new strategy, the rest of the architecture is equivalent to the ViT, with only an encoder and a class token, an example in Figure 2.7a. DeiT is able to obtain better results than its teacher network achieving an accuracy of 85.2% versus the 82.9% accuracy of RegNetY-16GF (teacher) on ImageNet.

### 2.3.3 Bidirectional Encoder representation from image Transformers

Presented in June 2021, Bidirectional Encoder representation from image Transformers (BEiT) imitates the Natural Language Processing (NLP) dedicated transformer BERT (Bao et al. [5]). In this transformer (Figure 2.7b), each image is represented as a tensor of visual tokens, split into patches, as in ViT. From each image, it extracts a matrix of $14 \times 14$ visual tokens where each one corresponds to a specific part of the image. During training, some patches are masked and the transformer gives back the visual tokens of the masked patches. BEiT is pre-trained in a self-supervised manner and only at a second time, the task layers of the encoder are activated for fine-tuning the model parameters and adapting them to the specific task. Thanks to this

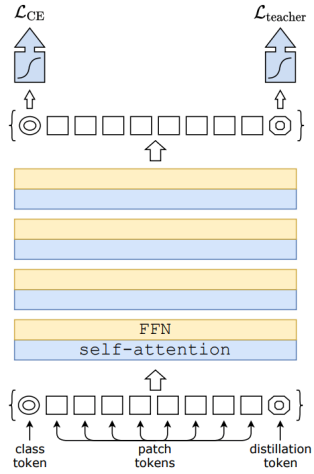
approach, BEiT is a general-purpose visual transformer, which has obtained great results in semantic segmentation and image classification achieving an 88.6% accuracy on ImageNet.

### 2.3.4 Shifted Windows Transformer

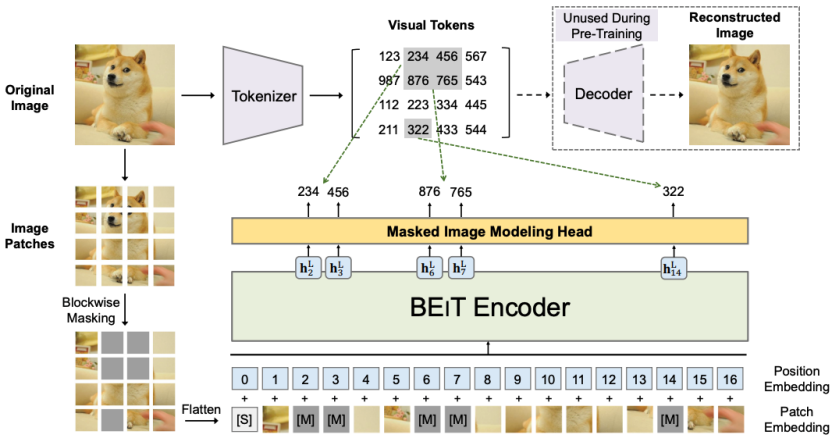
Shifted Windows Transformer (SWin) [75] Presented in March 2021, it implements a hierarchical system with a shifted windows mechanism, similar to the Temporal Shifted Module (TSM) [70]. This transformer uses smaller non-overlapping windows that allow the transformer to obtain information about smaller details. Afterward, the windows get larger to obtain a more general vision of the image. The aim of the shifted windows mechanism is to improve connections across the windows to enhance the transformer modeling power. The architecture is shown in Figure 2.7c. This is a hierarchical transformer and its computational cost is linear with image resolution compared to ViT which has a quadratic cost. It achieves a 90.17% accuracy on ImageNet.

### 2.3.5 Cross-Shaped Windows Transformer

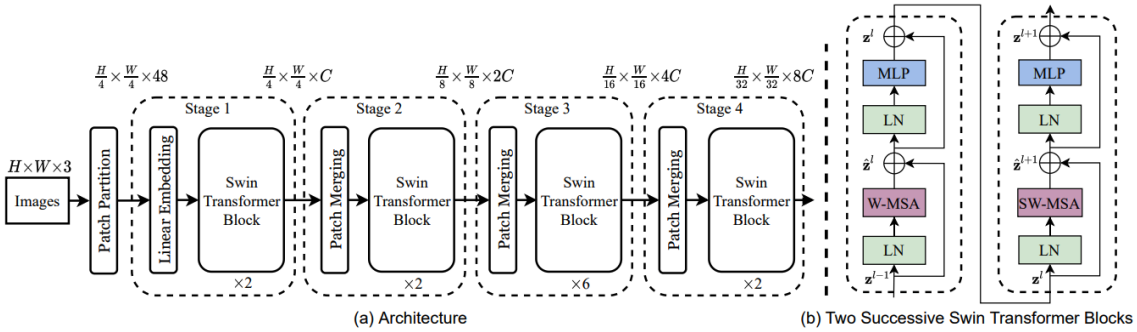
It was released in July 2021 [23] and implements a hierarchical system with horizontal and vertical stripes instead of squared windows. The vertical stripes and the horizontal stripes are non-overlapping between themselves. These stripes get wider within each layer. Cross-Shaped Windows Transformer (CSWin) also implements a new method called Locally-enhanced Positional Encoding (LePE) carrying the positional information inside every block. Simultaneously, it stands on the hypothesis that the most important positional information comes from the close surroundings of each element, appending an additional term of depth-wise convolution to the attention formula (Figure 2.7d). It reaches results similar to the SWin on ImageNet.



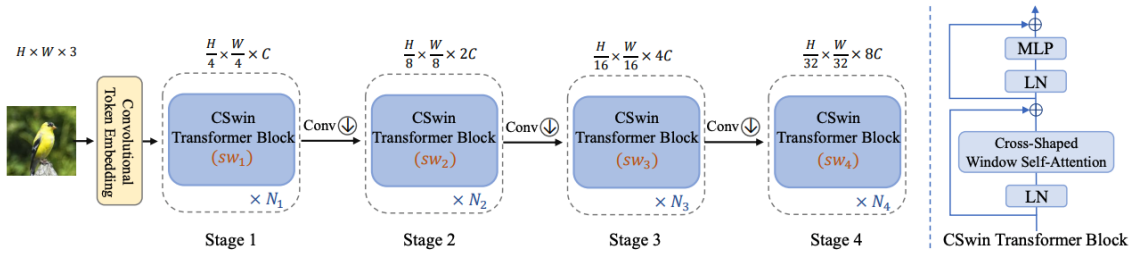

(A) Data-efficient image Transformer example [112].




(B) Bidirectional Encoder from image Transformer example [5].




(C) Shifted Windows Transformer example [75].




(D) Cross-Shaped Window Transformer example [23].

FIGURE 2.7. Pipelines of the transformers used in this work.

CHAPTER 3

*A picture may be worth a thousand words,*
*a **formula** is worth a thousand pictures*

EDSGER W. DIJKSTRA

# INVERTIBLE CNN

## 3.1 Introduction to CNNs

Convolutional filters have been largely used in the history of signal and image processing. For image processing, and depending on their element values, they can be used for a large variety of operations on images, such as filtering, blurring, sharping, or enhancing. Convolutional Neural Networks (CNN) (LeCun *et al.,* [66]) use the strength of these operations and have become one of the most powerful Deep Learning (DL) methods in image processing. The basic idea is very simple: stack several convolutions in a row and make the network learn by itself the values of the kernel to approach a certain label as accurately as possible. The evolution of parameters of the network is clearly established with mathematical precision by the processes of gradient calculation, optimization, and back-propagation. However, we have little control over the properties of those kernels.
Many works tried to understand how we can extract information from the mathematical properties of the convolutional kernels and how we can use these properties to positively influence training and improve efficiency. As we noticed during our research, after the learning process, most of the resulting convolution kernels are mathematically non-invertible. This appears to be a recurrent behavior of CNNs in our experiments and may imply that, after each layer, the amount of information embedded in the hidden layers can decrease at each layer. Even though this seems to be a key point to compress useful information disregarding redundant data and noise and understanding CNNs behavior, the scientific literature does not shed much light on this issue.

### 3.1.1 State of the Art on invertible CNNs

One of the oldest articles regarding invertible convolutions for neural networks, and one of those which inspired our work, has been by Gilbert *et al.,* [36]. Using special convolution operators

associated with orthonormal matrices, they are able to rigorously invert those convolutions and reconstruct the original input with an error that is equal to the rounding error of the machine. They use three-dimensional tensors to represent a series of bi-dimensional kernels. This implies the use of a large number of parameters and memory. Furthermore, the constraint of orthonormal kernels is very strict and reduces the number of free parameters in the kernel, being only 3 for a $3 \times 3$ kernel, for example.
Another interesting early work is by Gomez *et al.,* [37]. They developed an invertible residual network that by construction needs to run at least twice the number of operations compared to its non-invertible counterpart to obtain the same level of abstraction. Upon that, there is a slight increase in the number of parameters because of residuals. The reconstruction error is equal to the rounding error of the machine and its impact is described as "minimum", negligible in most of the cases. Similar research has been conducted by Jacobsen *et al.,* [56], it is called i-RevNet and has the same backbone as the previous one. The main difference is that they use a larger number of parameters, up to 7 times more than the relative ResNet version, and achieves good classification results. The mean reconstruction error has been measured to be $3 \cdot 10^{-6}$ on ImageNet, for a single layer. Similar considerations can be done for Behrmann *et al.,* [6]. It uses the approach of Gomez *et al.,* [37] on the architecture of Jacobsen *et al.,* [56]. The differences stand in the formalism and the computation of the log-determinant of the Jacobian matrix associated with the filters, done using power series on the trace.

Karami *et al.,* [59] proposes another kind of residual CNN that allows the inversion of the convolutions through the use of circulant matrices. Those are particular cases of Toeplitz matrices and have very similar properties to Toeplitz matrices, but have a lower number of free parameters. This method has a high computational cost too. The invertible convolutional network of Finzi *et al.,* [33] uses a method very similar to ours, but with stricter conditions. The condition imposed during training is that the kernel must be Gaussian. This reduces the number of free parameters of the kernel from 9 to 2, amplitude and variance[1]. They use Fourier transform for formal inversion of the operation. The computation complexity at each training step is $\mathcal{O}(n^3)$.

### 3.1.2 Aim of invertible convolution

**Definition 3.1.3.** ***Invertibility***

*Let $A$ be an $n \times n$ matrix. It is called* invertible *or* nonsingular*, if there exists $B$ an $n \times n$ matrix such that:*

$$AB = BA = \mathbb{I}_n$$

*where $\mathbb{I}_n$ denotes the identity matrix of size $n$. If $B$ exists, then it is uniquely determined by $A$, and it is called the inverse of $A$, denoted by $A^{-1}$. A square matrix that is not invertible is called* singular *or* degenerate*. A square matrix is singular if and only if its determinant is zero.*

[1] Amplitude and variance are the only free variables of those kinds of Gaussian kernel because, in order to be invertible, the mean value must be fixed to zero and the variance must be equal on both axes.

**Definition 3.1.4.** ***Invertibility conditions***

*Let us suppose we have an $n \times n$ matrix $\kappa : \mathbb{R}^n \to \mathbb{R}^n$, and we define the linear map $f$ by $f(x) = \kappa x$, with $x \in \mathbb{R}^n$. Then $\kappa$ is invertible if and only if $\kappa$ has rank $n$, i.e. $\kappa$ has full rank. In addition, $\kappa$ is invertible if and only if its determinant is finite and different from zero.*

If an operator is not invertible, it means that the information contained in the original input may not be reconstructed entirely. In the field of deep learning, this means that some of the features of the input image will not be processed by the network in the latest layers, which may negatively influence the learning process. In addition, if the convolution operator has full rank, it can also be inverted and we can reconstruct the input exactly as it was.
In this work, we propose to explicitly describe and visualize what happens to the parameters of convolution during training, and use this information to infer where a loss of information can occur and how to avoid it. We use a reformulation of the convolution to define the backbone of a rigorously invertible CNN, test it on a very common dataset, and prove its invertibility performance on real data.
The aim of this work is to propose a new $3 \times 3$ convolution that can be inverted without resorting to its Fourier transform but directly addressing the inversion of its associated matrix. Fourier transform is commonly used to treat convolutional problems because a convolution becomes a simple multiplication in the frequency domain and it simplifies the process the inversion. Although, when the size of the matrix is large, the calculation of the inverse comes at the cost of an approximation, which generally gets worse with the increase in size, because of tolerance in numerical approximations. In chapter 3.2, we propose the mathematical formulation of an invertible convolutional kernel, its theoretical application and an architecture that can be used as a test. In chapter 3.3, we present an analysis of the data extracted during training and their evolution, followed by our results on image classification, reconstruction, and generation.

## 3.2 Methods for invertible CNN

**Definition 3.2.1.** ***Diagonalization***

*Let $A \in \mathbb{F}^{n\times n}(\mathbb{K})$ a squared matrix with value in a field $\mathbb{K}$ is diagonalizable if $\exists$ an $n \times n$ invertible matrix (i.e. an element of the general linear group $GL_n(F)$) $P$ such that $P^{-1}AP$ is a diagonal matrix.*
*$A \in \mathbb{F}^{n\times n}$ diagonalizable $\iff \exists P \in GL_n(F)$: $P^{-1}AP$ is diagonal.*

**Definition 3.2.2.** ***Vectorization***

*Let $A \in \mathbb{R}^{n\times n}$ a generic matrix of elements $a_{ij}$ and Let $A' \in \mathbb{R}^{n^2}$ a column vector of elements $a'_m$. If the elements of $A'$ are such that:*

$$a'_{j+n(i-1)} = a_{ij}$$

*then, we call $A'$ the vectorization of $A$, or $A$ vectorized.*

**Lemma 3.2.1.** *Vectorization is always invertible, knowing the original size of the matrix, because it is an ordered change of indices.*

### 3.2.3 Reformulation of the convolution

**Definition 3.2.4.** ***Convolution***
*Let $\kappa \in \mathbb{R}^{3\times 3}$ a kernel of elements $k_{ij}$ and $X \in \mathbb{R}^{n\times n}$ a generic matrix. Then, the operation $X$ convoluted $\kappa$, namely $X * \kappa = Y$ can be written as:*

$$y_{ij} = \sum_{m=-1}^{1}\sum_{l=-1}^{1} k_{2+m,2+l} \cdot x_{i+m,j+l}$$

*All the elements with indices zero or higher than the maximum size are treated as zero*[2].

In a more visual format, a convolution can be written as:

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} & \cdots \\ x_{21} & x_{22} & x_{23} & x_{24} & \cdots \\ x_{31} & x_{32} & x_{33} & x_{34} & \cdots \\ x_{41} & x_{42} & x_{43} & x_{44} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} * \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & y_{13} & y_{14} & \cdots \\ y_{21} & y_{22} & y_{23} & y_{24} & \cdots \\ y_{31} & y_{32} & y_{33} & y_{34} & \cdots \\ y_{41} & y_{42} & y_{43} & y_{44} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{3.1}$$

Or alternatively, it can be expressed as a matrix product of a Toeplitz matrix with the vectorized input (Gray *et al.*, [42]), in the form of $KX' = Y'$, such as:

$$\begin{bmatrix} k_{22} & k_{23} & 0 & 0 & 0 & \dots & k_{32} & k_{33} & 0 & \dots \\ k_{21} & k_{22} & k_{23} & 0 & 0 & \dots & k_{31} & k_{32} & k_{33} & \dots \\ 0 & k_{21} & k_{22} & k_{23} & 0 & \dots & 0 & k_{31} & k_{32} & \dots \\ 0 & 0 & k_{21} & k_{22} & k_{23} & \dots & 0 & 0 & k_{31} & \dots \\ 0 & 0 & 0 & k_{21} & k_{22} & \dots & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots \\ k_{12} & k_{13} & 0 & 0 & 0 & \dots & k_{22} & k_{23} & 0 & \dots \\ k_{11} & k_{12} & k_{13} & 0 & 0 & \dots & k_{21} & k_{22} & k_{23} & \dots \\ 0 & k_{11} & k_{12} & k_{13} & 0 & \dots & 0 & k_{21} & k_{22} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \cdot \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ \vdots \\ x_{21} \\ x_{22} \\ x_{23} \\ \vdots \end{bmatrix} = \begin{bmatrix} y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \\ \vdots \\ y_{21} \\ y_{22} \\ y_{23} \\ \vdots \end{bmatrix} \tag{3.2}$$

where $X'$, $Y'$ are the vectorization of the matrices $X$, $Y$ and $K$ is the matrix associated with convolution.
If the determinant of $K$ is different from zero ($\Delta(K) \neq 0$), it means that the vectors of $K$ are linearly independent and, then, $K$ will be invertible.

[2] In image processing, this is known as *zero padding*. The convolution operation, from here until the end of this document, is intended with zero padding and with output matrix dimensions equal to the input.

By construction, $K$ is a $n \times n$ tri-diagonal Toeplitz block matrix with each block being another $n \times n$ matrix. The total dimension of $K$ is $n^2 \times n^2$ with blocks different from zero only in the three main diagonals.

$$K = \begin{bmatrix} W_2 & W_3 & O & O & \dots \\ W_1 & W_2 & W_3 & O & \dots \\ O & W_1 & W_2 & W_3 & \dots \\ O & O & W_1 & W_2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \qquad W_r = \begin{bmatrix} k_{r2} & k_{r3} & 0 & 0 & \dots \\ k_{r1} & k_{r2} & k_{r3} & 0 & \dots \\ 0 & k_{r1} & k_{r2} & k_{r3} & \dots \\ 0 & 0 & k_{r1} & k_{r2} & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{3.3}$$

where $W_r$ with $r = 1,2,3$ are again tri-diagonal Toeplitz matrices and $O$ is the null matrix.
The eigenvalues of a tri-diagonal $n \times n$ Toeplitz matrix are [8] given by:

$$\lambda_i = k_{r2} + 2\sqrt{k_{r1}k_{r3}}\, cos\Big(\frac{i\pi}{n+1}\Big), \qquad i = \{1,\dots,n\}.$$

By definition, the determinant of a matrix equals the product of its eigenvalues. In the case of $W_r$, the determinant is:

$$\Delta(W_r) = \prod_{i=1}^{n} \lambda_i = \prod_{i=1}^{n} \Big[k_{r2} + 2\sqrt{k_{r1}k_{r3}}\, cos\Big(\frac{i\pi}{n+1}\Big)\Big]$$

As we can deduce, under the condition $k_{r1}k_{r3} < 0$, all the eigenvalues may have a complex part. Supposing $n$ to be even, the support of the cosine function is contained in the segment $[0,\pi]$. In this interval, the cosine function is anti-symmetric with respect to $\pi/2$. Therefore, the terms containing the square root will disappear.
This leads to the following simplification steps:

1. Define $\varphi = 2\sqrt{k_{r1}k_{r3}}$ and $k = k_{r2}$;

2. Split the product into two parts, one for the first $n/2$ elements and the second for the remaining $n/2$;

3. Change the indices in the second product;

4. Re-writing the second product using the cosine properties, we obtain:

$$\Delta(W_r) = \prod_i^n \lambda_i = \prod_i^n \left[k_{r2} + 2\sqrt{k_{r1}k_{r3}}\, cos\left(\frac{i\pi}{n+1}\right)\right]$$
$$= \prod_{s=1}^{n/2} \left[k_{r2} + \varphi\, cos\left(\frac{s\pi}{n+1}\right)\right] \cdot \prod_{t=n/2+1}^{n} \left[k_{r2} + \varphi\, cos\left(\frac{t\pi}{n+1}\right)\right]$$
$$\text{with} \quad s = 1 \ldots \frac{n}{2}\,, \quad t = \frac{n}{2} + 1 \ldots n \quad \Rightarrow \quad t = n + 1 - s$$
$$= \prod_{s=1}^{n/2} \left[k_{r2} + \varphi\, cos\left(\frac{s\pi}{n+1}\right)\right] \cdot \prod_{s=1}^{n/2} \left[k_{r2} + \varphi\, cos\left(\frac{(n+1-s)\pi}{n+1}\right)\right]$$
$$\text{Taking into account that:} \quad cos\left(\frac{n+1}{n+1}\pi - \frac{s\pi}{n+1}\right) = cos\left(\pi - \frac{s\pi}{n+1}\right) = -cos\left(\frac{s\pi}{n+1}\right)$$
$$\Delta(W_r) = \prod_{s=1}^{n/2} \left[\left[k_{r2} + \varphi\, cos\left(\frac{s\pi}{n+1}\right)\right] \cdot \left[k_{r2} - \varphi\, cos\left(\frac{s\pi}{n+1}\right)\right]\right]$$
$$= \prod_{s=1}^{n/2} \left[k_{r2}^2 - \varphi^2\, cos^2\left(\frac{s\pi}{n+1}\right)\right]$$
$$= \prod_{s=1}^{n/2} \left[k_{r2}^2 - 4k_{r1}k_{r3}\, cos^2\left(\frac{s\pi}{n+1}\right)\right] \tag{3.4}$$

We have now gotten rid of all the eventual complex terms and removed the square roots in the equation. Thus, $\Delta(W) \in \mathbb{R}$ for all kernel values $k \in \mathbb{R}$. This closed-form stands only for $n$ even.

### 3.2.5 Proof of Invertibility

Let us suppose $\kappa$ is defined by the convolution between two other kernels $\tau$ and $\varrho$ of shape $1 \times 3$ and $3 \times 1$ respectively, in the following way:

$$\tau * \varrho = \kappa\,, \qquad \begin{pmatrix} a & b & c \end{pmatrix} * \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{bmatrix} \alpha a & \alpha b & \alpha c \\ \beta a & \beta b & \beta c \\ \gamma a & \gamma b & \gamma c \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \tag{3.5}$$

It may be noticed that the number of independent parameters is only six instead of nine as in a generic $3 \times 3$ kernel. Nonetheless, the size of the receptive field[3] remains the same (Szegedy *et al.,* [108]). Reformalizing the convolution has been a requirement due to the extreme difficulty encountered in extracting a relatively simple determinant formula for a generic kernel. Using associativity of the convolutional operation and the property described in Equation 3.1 and Equation 3.2 , we reformulate the convolution of $X$ with $\tau$ and $\varrho$. Let us consider the following operation:

$$X * \kappa = X * \tau * \varrho = (X * \tau) * \varrho = P(X * \tau) = P(TX) = PTX$$

[3]The receptive field in a convolution is the portion of the matrix on which we apply the convolutional kernel.

where $T$ and $P$ are matrices associated with $\tau$ and $\varrho$. By construction, $T$ and $P$ are defined as follows:

$$T = \begin{bmatrix} D & 0 & 0 & 0 & \dots \\ 0 & D & 0 & 0 & \dots \\ 0 & 0 & D & 0 & \dots \\ 0 & 0 & 0 & D & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \qquad D = \begin{bmatrix} b & c & 0 & 0 & \dots \\ a & b & c & 0 & \dots \\ 0 & a & b & c & \dots \\ 0 & 0 & a & b & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{3.6}$$

$$P = \begin{bmatrix} B & C & 0 & 0 & \dots \\ A & B & C & 0 & \dots \\ 0 & A & B & C & \dots \\ 0 & 0 & A & B & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \qquad \begin{cases} A = \alpha \cdot \mathbb{I} \\ B = \beta \cdot \mathbb{I} \\ C = \gamma \cdot \mathbb{I} \end{cases} \tag{3.7}$$

where $\mathbb{I}$ is the Identity matrix. This means that:

$$PT = \begin{bmatrix} BD & CD & 0 & 0 & \dots \\ AD & BD & CD & 0 & \dots \\ 0 & AD & BD & CD & \dots \\ 0 & 0 & AD & BD & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} = \begin{bmatrix} \beta b & \beta c & 0 & 0 & 0 & \dots & \gamma b & \gamma c & 0 & \dots \\ \beta a & \beta b & \beta c & 0 & 0 & \dots & \gamma a & \gamma b & \gamma c & \dots \\ 0 & \beta a & \beta b & \beta c & 0 & \dots & 0 & \gamma a & \gamma b & \dots \\ 0 & 0 & \beta a & \beta b & \beta c & \dots & 0 & 0 & \gamma a & \dots \\ 0 & 0 & 0 & \beta a & \beta b & \dots & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots \\ \alpha b & \alpha c & 0 & 0 & 0 & \dots & \beta b & \beta c & 0 & \dots \\ \alpha a & \alpha b & \alpha c & 0 & 0 & \dots & \beta a & \beta b & \beta c & \dots \\ 0 & \alpha a & \alpha b & \alpha c & 0 & \dots & 0 & \beta a & \beta b & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{3.8}$$

It can be shown that $PT$ is equal to the matrix associated with the $\tau * \varrho$ convolution kernel. From that, we can naturally derive that $\tau * \varrho = \kappa \iff PT = K$.

From Equation 3.4, we know that:

$$\Delta(D) = \prod_{s=1}^{n/2} \left[ b^2 - 4ac \cos^2\left(\frac{s\pi}{n+1}\right) \right]$$

and knowing that the determinant of a diagonal block matrix, like $T$, is the product of the determinants of the blocks on the diagonal, we obtain:

$$\Delta(T) = \left[\Delta(D)\right]^n = \left[ \prod_{s=1}^{n/2} \left[ b^2 - 4ac \cos^2\left(\frac{s\pi}{n+1}\right) \right] \right]^n.$$

If the previous equation is non-zero, we guarantee the invertibility of $T$.
Similarly, the determinant of a diagonal matrix is simply the product of the elements on the

diagonal, so $\Delta(A) = \alpha^n$, $\Delta(B) = \beta^n$, and $\Delta(C) = \gamma^n$. With all simultaneously $\alpha, \beta, \gamma \neq 0$, the matrices $A$, $B$, and $C$ will always be invertible.
By induction, it can be demonstrated (see Appendix) that the determinant of $P$ is the same as the one of a tri-diagonal Toeplitz, for the $n$-th power.
Adding up and knowing that the determinant is a multiplicative map, we obtain:

$$\begin{aligned}\Delta(PT) = \Delta(P)\Delta(T) &= \left[\prod_{s=1}^{n/2}\left[b^2 - 4ac\ cos^2\left(\frac{s\pi}{n+1}\right)\right]\right]^n \cdot \left[\prod_{s=1}^{n/2}\left[\beta^2 - 4\alpha\gamma\ cos^2\left(\frac{s\pi}{n+1}\right)\right]\right]^n \\ &= \left[\prod_{s=1}^{n/2}\left[b^2 - 4ac\ cos^2\left(\frac{s\pi}{n+1}\right)\right] \cdot \left[\beta^2 - 4\alpha\gamma\ cos^2\left(\frac{s\pi}{n+1}\right)\right]\right]^n \\ &= \left[\prod_{s=1}^{n/2}\left[b^2 - ac\,\Gamma_{s,n}\right] \cdot \left[\beta^2 - \alpha\gamma\,\Gamma_{s,n}\right]\right]^n . \qquad (3.9)\end{aligned}$$

Now we have a formula that establishes a relationship between the kernel and the determinant of the matrix associated with the kernel. If the determinant of Equation 3.9 is different from zero, then the convolution will be invertible.

Note that $\Gamma_{s,n} = 4cos^2\left(\frac{s\pi}{n+1}\right)$ could be pre-calculated and stored in a look-up table.

### 3.2.6 Applications

The natural application of the method explained before is on CNNs. We can impose the condition of invertibility of the convolutional kernels (see Equation 3.9) directly inside the loss function, and train a network that is invertible from its first steps of training. This method has two significant implications on the neural network:

- The number of free parameters of each kernel passes from 9 to 6 and this may influence the level of abstraction of the network.
- The increase of computational complexity of the loss function makes the training more time consuming for an equal space in memory occupation.

The time spent in feed-forward and back-propagation remains equal as before because the actual conversion of the convolution into matrix multiplication and its inversion are executed only when inverting the model and not in the learning phase.

#### 3.2.6.1 Single channel case

Let us assume an input $X$ of shape $n \times n$, a kernel $\kappa$ of size $3 \times 3$ that respects the aforesaid conditions, its associated matrix $K$, an invertible activation function $\sigma$, and a bias $\Phi$ of dimension 1 or $n \times n$, indifferently. Then, we can formalize a basic convolutional layer of the network as follows:

$$Y = \sigma(X * \kappa + \Phi) = \sigma(KX + \Phi).$$

From this, the inversion of the layer can be written as:

$$X = K^{-1}(\sigma^{-1}Y - \Phi). \tag{3.10}$$

### 3.2.7 Test Architectures

The practical key principle of this work is substituting a generic $3 \times 3$ convolution with a sequence of two convolutions: one $3 \times 1$ followed by one $1 \times 3$. This allows us to control the determinant of the operator and makes the layer invertible. In figure 3.1, we present the two versions of what we will call *convolutional layers*, from now on. In all this work, we refer as *standard networks* to the architectures containing the single $3 \times 3$ convolution, versus our method represented by an architecture containing double convolutions. We use hyperbolic-tangent (tanh) as the activation function because it is invertible. In general, any invertible activation function can be used, but it could introduce higher error in reconstruction, especially, asymptotic limited functions such as softsign function. Only in one experiment, because of the use of a large number of layers, we use a linear activation function to focus only on the convolution. Another source of information loss is the *pooling layer*. In a normal pool with stride 2, the size of the input is halved for each dimension, so 3/4 of the information is discarded. In substitution to those, we use a pooling layer where the input is divided into four outputs following the rule of figure 3.2, i.e. each of the outputs takes the elements of the input, skipping one row and one column for each two, in a way that each single pixel of the input is saved in one of the outputs.

All the architectures used in this work have been tested using both configurations of convolutional kernels. These architectures have been chosen as examples with the aim of showing the possibilities of this method. Our purpose here is to demonstrate there is an alternative method for convolution inversion without significant accuracy loss.

The first architecture is a simple sequence of five convolutional layers, followed by a single fully connected layer. This configuration aims to give an estimation of the error propagation during image reconstruction.

The second one is organized into two levels: the first one is just a single convolutional layer followed by a pooling layer. The four outputs of the pooling go to another four convolutional layers, whose results are then concatenated and sent to a fully connected layer. Graphic examples are given in figure 3.3.

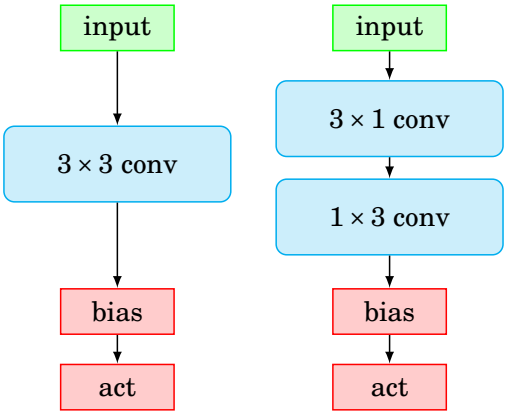



FIGURE 3.1. Flowcharts of the two versions of our convolutional layers. On the right side, is our invertible version. The abbreviations *conv* and *act* stand for convolution and activation.

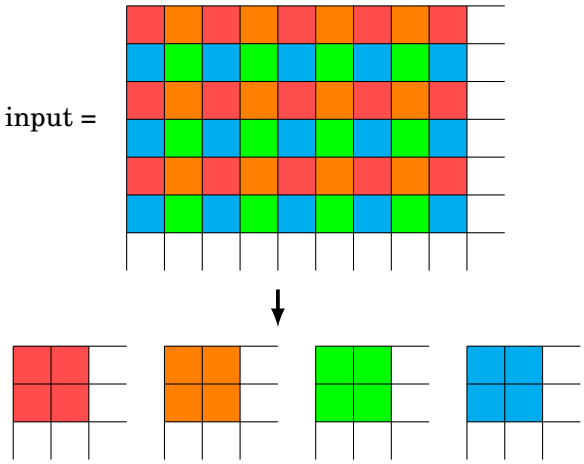



FIGURE 3.2. Example of the pooling layer.

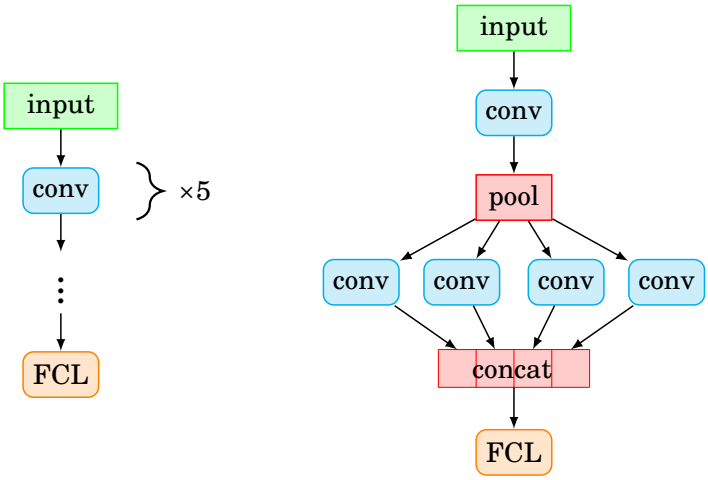



FIGURE 3.3. Flowcharts of the test architectures. In the rest of this work, we refer to these as architecture number 1, and 2, reading from left to right. The *conv* layer can be alternately one of the two versions in Figure 3.1. The abbreviations *conv*, *concat* and *FCL* stand for convolution, concatenation, and Fully Connected Layer.

### 3.2.8 Metrics

The metrics used in this work for image classification is *accuracy* in its most common formulation. In order to calculate the inversion error for a generic square matrix $A$, we propose the following set:

$$\varepsilon_{inv} = |AA^{-1} - \mathbb{I}| \,, \quad \mathscr{E}_{inv} = max\{|AA^{-1} - \mathbb{I}|\} \,, \quad \bar{\varepsilon}_{inv} = mean\{|AA^{-1} - \mathbb{I}|\} \tag{3.11}$$

where $max$ gives back the maximum value inside the matrix, $mean$ is the average of the matrix values, and $\mathbb{I}$ is the identity matrix of the appropriate dimension.
The error of reconstruction is similarly defined as:

$$\varepsilon_{rec} = |X - Y| \tag{3.12}$$

where $X$ is the original input, and $Y$ is the reconstructed image.
We also decided to contextualize the aforesaid formula for an 8-bit image, defining the bit-reconstruction error. This gives the maximum difference between two matrices (as in Equation 3.12) in bits with the following equation:

$$\varepsilon_{bit} = \left\lfloor \frac{|X - Y|}{1/255} \right\rfloor.$$

In this case, $\lfloor x \rfloor$ represents the *integer part* or (floor operation) of a number $x$, instead the factor 1/255 is the distance between two colours in an 8-bit image, normalized from 0 to 1. In the rest of the work, we will refer to this quantity as *1-bit distance* or *1-bit error*. As in Equation 3.11, we always use $\mathscr{E}$ for indicating the maximum error, and $\bar{\varepsilon}$ for the average error.

## 3.3 Results for invertible CNN

All experiments have been written in Python, using the TensorFlow module, and run on a virtual machine with 32 Intel Xeon CPUs and an NVidia Tesla V100 GPU. The categorical cross-entropy function has been used with Adagrad or Adam optimizer. All experiments in this work have been done using MNIST dataset.

### 3.3.1 Classification Results

An important characteristic of this new method is that the network with invertible convolutions must have similar test accuracy results, compared to its non-invertible counterpart. We took our test architectures and, for each of them, we ran five training with different learning rates. We used the test set itself as validation, with the objective of demonstrating the invertibility qualities of the proposed method. In addition to the accuracy on the test set, we provide the maximum and mean reconstruction errors averaged on all the samples inside the test set.

All the aforesaid data are presented in Table 3.1. The reconstruction error has not been computed for the non-invertible architectures (reconstruction was not possible for these architectures).

### 3.3.2 Kernel properties evolution

In order to be able to use Equation 3.10 for inverting the layer, we need the associated matrix $K$ to have a finite determinant different from zero (Definition 3.1.4). This is a necessary condition,

| architecture | method | LR | accuracy | $\mathscr{E}_{inv}$ | $\bar{\varepsilon}_{inv}$ |
|---|---|---|---|---|---|
| 1 | non-invertible | 0.0005 | 95.43% | - | - |
| | | 0.001 | 95.77% | - | - |
| | | 0.002 | 95.53% | - | - |
| | | 0.003 | 95.53% | - | - |
| | | 0.005 | 95.66% | - | - |
| | invertible | 0.0005 | 91.94% | $8.15 \pm 8.40 \cdot 10^{-4}$ | $5.18 \pm 3.59 \cdot 10^{-5}$ |
| | | 0.001 | 94.54% | $5.21 \pm 3.13 \cdot 10^{-4}$ | $7.09 \pm 3.91 \cdot 10^{-5}$ |
| | | 0.002 | 94.11% | $1.51 \pm 1.19 \cdot 10^{-3}$ | $7.82 \pm 9.92 \cdot 10^{-5}$ |
| | | 0.003 | 94.48% | $7.21 \pm 5.07 \cdot 10^{-4}$ | $5.64 \pm 3.45 \cdot 10^{-5}$ |
| | | 0.005 | 95.02% | $1.96 \pm 1.87 \cdot 10^{-3}$ | $1.49 \pm 1.24 \cdot 10^{-4}$ |
| 2 | non-invertible | 0.01 | 97.33% | - | - |
| | | 0.03 | 95.86% | - | - |
| | | 0.05 | 94.31% | - | - |
| | | 0.1 | 91.95% | - | - |
| | | 0.3 | - | - | - |
| | invertible | 0.01 | 94.19% | $3.79 \pm 6.46 \cdot 10^{-3}$ | $1.48 \pm 2.27 \cdot 10^{-4}$ |
| | | 0.03 | 95.28% | $5.57 \pm 5.11 \cdot 10^{-5}$ | $2.71 \pm 2.70 \cdot 10^{-5}$ |
| | | 0.05 | 95.57% | $1.22 \pm 1.64 \cdot 10^{-3}$ | $5.81 \pm 7.86 \cdot 10^{-5}$ |
| | | 0.1 | 95.89% | $1.56 \pm 1.34 \cdot 10^{-5}$ | $1.88 \pm 1.10 \cdot 10^{-6}$ |
| | | 0.3 | 96.81% | $2.12 \pm 3.11 \cdot 10^{-3}$ | $7.97 \pm 11.20 \cdot 10^{-5}$ |

TABLE 3.1. Classification accuracy results and reconstruction errors (maximum and average), for the two versions of architecture 1 and 2. We used Adam optimizer, for the first architecture and Adagrad for the second one. LR is the learning rate.

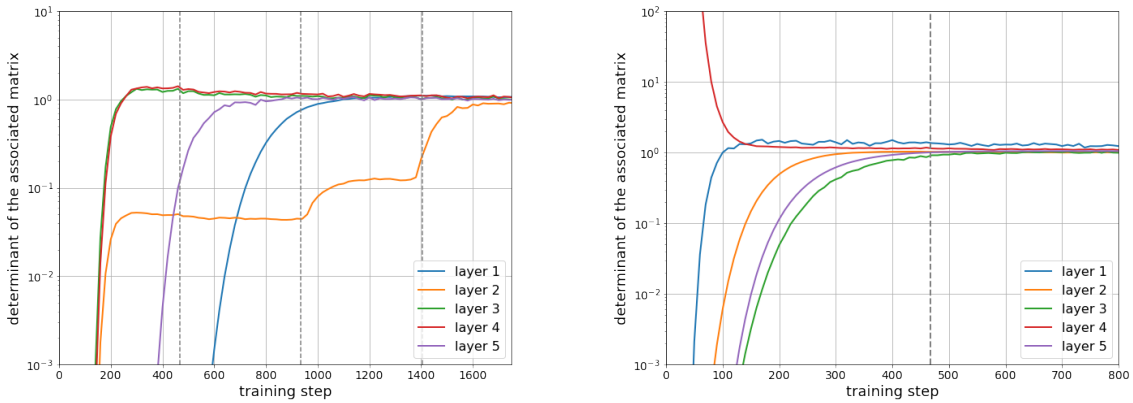



FIGURE 3.4. Evolution of the determinant of the associated matrix at each training step (batch size 128) for the five layers of architecture 1 (left) and 2 (right), with learning rate 0.03. Each epoch has been indicated with a grey dashed line. The classification accuracy of these model is 94.54% and 95.28%.

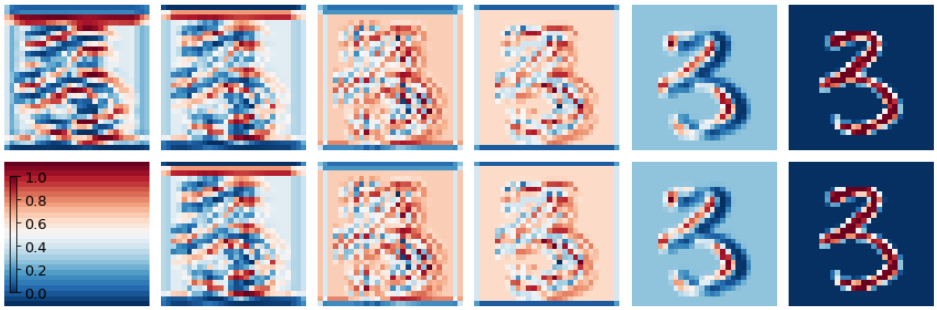



FIGURE 3.5. Example of image reconstruction. In the upper part, in order from right to left, the original image and the outputs of the five convolutional layers. In the bottom, in the inverse order, the reconstructed images. In the bottom left corner, there is a colormap. Images have been normalized in the interval $[0, 1]$.

but it is not yet sufficient to ensure invertibility, with an error *low enough*, in an environment subjected by a limited machine precision, due to rounding in floating point arithmetic.
One possible way of inverting a matrix $A$ is through the calculation of its cofactor matrix $C$, as $A^{-1} = C^T/\Delta(A)$. It is known [28] that the elements of the cofactor matrix of a tri-diagonal Toeplitz (as $W$ in Equation 3.3, without the $r$ index) are:

$$c_{ij} = \begin{cases} \Delta(W_{i-1})\, k_1^{j-i}\, \Delta(W_{n-j}) & \text{with } i \leq j \\ \Delta(W_{j-1})\, k_3^{i-j}\, \Delta(W_{n-i}) & \text{with } i \geq j \end{cases}$$

where, in this case, the index under $W$ is the dimension of the Toeplitz matrix, and $\Delta(W_0) = 1$, for convention. Therefore, it is essential to keep under control the determinant and the values of the elements outside the main diagonal ($k_1$ and $k_3$), during training. Because for $i \ll j$ and for $i \gg j$, the values of $c_{ij}$ can rapidly diverge to both very large and very small values, introducing an unpredictable error in the matrix inversion. As we can see in the example of Figure 3.4, using a proper loss function, we can force the determinants of the associated matrices to converge to values very near to one, in the first few epochs of the training.

### 3.3.3 Reconstruction Results

The objective of this work is to create a CNN capable of reconstructing the original image. We have presented our results in Table 3.1, but it is also interesting to assess the images produced inside the network. In Figure 3.5 we present an example of a reconstruction done by a network with 94.5% accuracy in the classification. Going deeper inside the network, the convolution transforms the original image into another image, unrecognizable by a human, but containing the same amount of information.

As a supplementary test, for addressing the error propagation in subsequent layer reconstructions, we trained a network, similar to the first one in Figure 3.3, but with linear activation

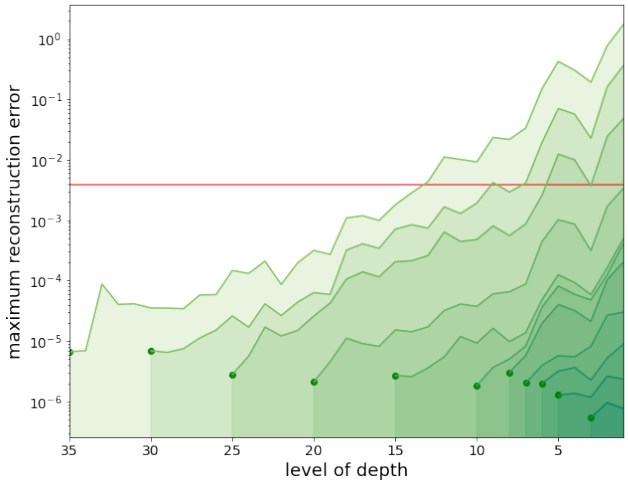



FIGURE 3.6. Example of the maximum image reconstruction error ($\mathscr{E}_{inv}$) propagation as function of the number of reconstructed layers, in logarithmic scale. The starting point of the reconstruction has been marked. 1-bit error has been plotted in red.

functions and with 35 stacked convolutions. Then, starting from the output of a precise layer, we reconstruct the previous outputs until the original image, using only the weights (i.e. the 6 values of the two convolutions and the bias) of the previous layers. The error propagates exponentially, independently of the layer chosen as starting point. Figure 3.6 shows how the maximum error grows with each step of the reconstruction, which can be compared with the 1-bit distance. The error in the first reconstructed layer tends to be lower as we decrease the level of depth.

In general, architecture 2 has shown a wider range of error in reconstruction than architecture 1. Figure 3.7 shows the histograms of the maximum reconstruction error for all the 50.000 images in the dataset, for both architectures. The curve of the cumulative counts provides an estimation of the total counts at each bit error. In this way, we can see that, for architecture 1, 84.9% of the cases have an error lower than the minimum detectable one. Instead, this value is 77.1% for architecture 2. The percentage of the cases over 3-bit error is 0.6% and 1.3% for the first and second case, respectively.

## 3.4 Discussions on invertible CNN

Our results show that invertibility comes at small cost in classification accuracy, but the control of weights is guaranteed throughout the whole training process. The two versions of the networks, in some cases with different learning rate or after a different number of epochs, tend to reach a very similar level of accuracy on the test set. This could be interpreted that regardless of the lower number of free parameters of the kernel (in our case, 6 against 9 for a standard $3 \times 3$ kernel),

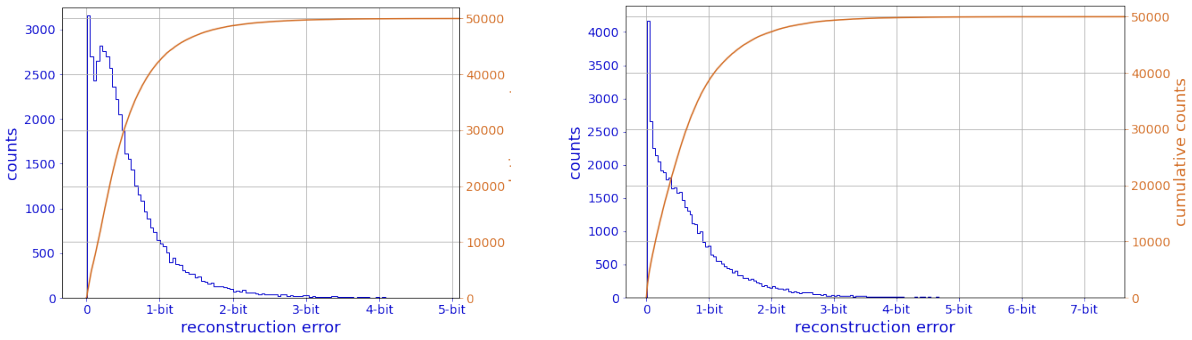



FIGURE 3.7. Histograms of the maximum image reconstruction error ($\mathscr{E}_{inv}$) for all the images in the test set (blue), for architecture 1 (left) and 2 (right). The error is shown in bits of the reconstructed image. Curve of the ordered cumulative counts (orange).

the level of abstraction seems to be preserved. As we can see from table 3.1, the reconstruction error of the full network is always behind the limit of the 1-bit error. This means that in the vast majority of those cases the differences between the originals and the reconstructed images is undetectable. The reasons why architecture 2 has a larger standard deviation on reconstruction errors may come from the pooling layer. Separating the input in four different smaller images, and then processing them separately, can introduce more uncertainty but, at the same time, gives better classification results.

In Figure 3.6, we have seen the error propagation depending from the depth of the network. It is remarkable that, even if there is the presence of a rounding machine-error, it is possible to reconstruct the original image even after it has been processed by 20 convolutional layers, with an error that is lower than the detectable numerical tolerances. All these reconstructions have been done using a 64 bit floating-point precision. Using special libraries with higher precision (e.g. float 128) or with fixed-point precision, it is possible to further reduce this error. The convergence of the determinants of the associated matrices (Figure 3.4) is smooth and very quick. Happening in the first epochs, it allows the network to learn under a regime of invertibility from the very first epochs of the training. We have to point out that the learning process is unpredictably influenced by the initialization values of the kernels. This will surely be a matter of focus for future work. From this initial work, we can hypothesize that the invertible networks may suffer less from the phenomenon of over-fitting. This hypothesis shall be addressed in future work.

## 3.5 Conclusions on invertible CNN

In this work, we presented a method for training, under specified conditions, an invertible convolutional neural network. Our experimental results show that it is possible to achieve very similar classification accuracy to standard CNNs but with the capability of reconstructing the input images starting from the output of the last convolutional layer with a reconstruction error in the range of numerical tolerances. We have seen that, with both our test architecture, around 80% of the images have been reconstructed with a total maximum error lower than 1-bit distance and around 99% of the images of the test set have been reconstructed with a total maximum error lower than 3-bit distance.

This work provides new insights into CNNs, opening the possibility to use them in new ways for image generation. For example, Variational Auto-Encoders (VAEs) and Generative Adversarial Networks (GANs) are the models that could receive the largest profit from applying our method.

CHAPTER 4

# THE LEARNING PROCESS OF VISUAL TRANSFORMERS

## 4.1 Introduction to Transformers

The second work presented in this thesis regards the optimization of transformer learning. Based on the concept of *attention*, introduced in 2014 by Bahdanau et al. [4], transformers have been able quickly to impose themselves on all-natural language processing tasks. It is only since 2020 that transformers [24] started to take the stage on image-related tasks as well. In less than two years, they managed to improve the results of previous neural network models and architectures. The problem of image classification still stands out, as it has historically been used to describe the evolution of Deep Learning and a *thermometer* of the progress made.

Despite the improvement in accuracy that they achieve, transformers, and especially image transformers, are still black boxes. It is not possible to determine exactly why and how they learn or which represents the exact hidden feature that makes one transformer better than another. We are interested to figure out how the learning procedure depends on the position of the blocks to learn and the epoch of the training process. We hypothesize that the key to understanding their properties is hidden in exploring the changes occurring within their attention layers. One possibility to ascertain this is through estimating the evolution and convergence e.g. focusing on the norm of differences between weight matrices during the training process. An alternative would be to use a Singular Value Decomposition (SVD) on the weight matrices of the attention layers with the objective of observing the behavior of the singular values during training. Their distribution, apparently independent of the type of transformer, can provide an inside view on what is their actual focus and what they are actually paying attention to. By doing so, we can determine which layers are crucial for learning and freeze those mostly affected by noise and do not contribute to learning.

Here we introduce a generic framework to explore how transformers learn and explore five of the

| Rank | Model | Accuracy | Transformer |
|---|---|---|---|
| 1 | CoAtNet-7 [29] | 90.88% | Hybrid |
| 2 | ViT-G/14 [124] | 90.45% | ViT |
| 3 | CoAtNet-6/14 [29] | 90.45% | Hybrid |
| 4 | ViT-MoE-15B [98] | 90.35% | ViT |
| 5 | Meta Pseudo Labels[a] [91] | 90.20% | No |
| 6 | SWinV2-G [76] | 90.17% | SWin |
| 7 | Florence-CoSwin-H [127] | 90.05% | SWin |
| 8 | Meta Pseudo Labels[b] [91] | 90% | No |

[a]with EfficientNet-L2.
[b]with EfficientNet-B6-Wide.

TABLE 4.1. Best results achieved on ImageNet [22] in Image Classification: chart. Hybrid means that it is a model that combines transformers with CNN

most commonly employed visual transformers to illustrate the utility of the framework. Without loss of generality, we provide a basis for understanding how they learn in image classification tasks. We also propose a method that significantly reduces the training time with a slight increase in accuracy.

## 4.2 State of the art on Transformer Optimization

If we consider the evolution of the results the best models got on ImageNet [22], a huge dataset with more than 14 million images, we will see that in 2011 AlexNet [64] had a 63.3% accuracy. 8 years later, CoAtNet-7 [29] achieved an accuracy of 90.88% and recently 7 more models have achieved an accuracy larger than 90%. Remarkably, of these top 8 models on ImageNet, more than half are transformers and only two of them do not implement attention (Table 4.1).

### 4.2.1 Works on Transformers Optimization

Because of the novelty of this field of research, the literature on frameworks to explore transformers' learning processes is limited. Shu et al. [104] developed a method that, working on the singular values of the weight matrices, is able to reduce more than 50% of the energy consumption of training, without losing accuracy. This was applied on Deit transformer, trained and tested on CIFAR and ImageNet datasets.
An alternative approach is given by Xie et al. [121] who proposed a Second order Transformer (SoT) that couples simultaneously word tokens and classification tokens. In order to implement that, they also use singular value Power Normalization (svPN) for cross-covariance matrices. This method can be applied to most of the latest transformers and proved to be able to increase

their accuracy by more than 2% on average[1], but with an increase of about 10% in the number of parameters and FLOPs. Although these works focus on improving or transforming weight matrices to achieve a final performance improvement, no one provides a deeper insight into the convergence of weight matrices during the learning process.

The paper of Chavan *et al.,* [15] tries to find an optimal sub-model from a vision transformer and presents the vision transformer slimming (ViT-Slim) framework. This method is based on a learnable $l1$ constraint that allows searching for more efficient architectures of the network, in different dimensions and with a single-shot training scheme. They set a threshold according to the requirements of accuracy-FLOPs trade-off of the user or the device. Their experiments show that ViT-Slim can reduce up to 40% the number of parameters and FLOPs on common vision transformers while slightly increasing the accuracy on ImageNet.

The same authors of DeiT propose also an optimize deeper transformer networks for image classification. Touvron *et al.,* [113] make two architecture changes on DeiT and this significantly improve the accuracy of this transformer, even with a deeper architecture. They called *LayerScale* their main introduction and it consists in a residual connection where the output of every attention layer is multiplied by a diagonal matrix of learning weights and then multiplied with the input of the previous layer. This leads to a model that does not saturate in the early stages of training and obtains high accuracy results on ImageNet, with a lower number of operations and parameters.

## 4.3 Methods for Transformer Optimization

### 4.3.1 Attention

Attention is the key concept of transformers. Given two inputs: $X \in \mathbb{R}^{n_x \times d_x}$ and $Y \in \mathbb{R}^{n_y \times d_y}$, we can define the Query, Key, and Value matrices as follows:

$$Q = XW^Q \ , \quad K = YW^K \ , \quad V = YW^V$$

where $W^Q \in \mathbb{R}^{d_x \times d_k}$, $W^K \in \mathbb{R}^{d_y \times d_k}$ and $W^V \in \mathbb{R}^{d_y \times d_v}$ are weight matrices with $d_k, d_v \in \mathbb{R}$, as in [74].

The attention operator is defined as

$$\text{Attention}(Q,K,V) = \text{Softmax}\Big(\frac{QK^T}{\sqrt{d_k}}\Big)V$$

The standard (unit) Softmax function $: \mathbb{R}^N \to (0,1)^N$, for a vector $x \in \mathbb{R}^N$ of elements $x_i$ and with $N \geq 1$, is defined by the formula:

$$\text{Softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}$$

[1] Calculated from the results of the most accurate models, called "heavyweight models", presented in the article.

### 4.3.2 Self-attention

Transformers use self-attention, which means that the inputs $X$ and $Y$ are equal. This concept is implemented in a Multi-Head Self Attention (MHSA) block, which joins multiple attention blocks.

Given $h > 0$ attentional blocks, each block with its query matrices $Q_i$, key $K_i$ and value $V_i$, obtained from the respective $W^{Q_i}$, $W^{K_i}$ and $W^{V_i}$ with $i \in \{1,2,...h\}$, and given a projection matrix $W0 \in \mathbb{R}^{hd_v \times d_x}$, then

$$\text{MHSA}(Q,K,V) = \text{Concat}(Z_1,...,Z_h)W0$$

where $Z_i = \text{Attention}(Q_i,K_i,V_i)$ and $Q,K,V$ are the matrices obtained by respectively concatenating matrices $Q_i$, $K_i$, $V_i$.

## 4.4 Criteria for transformers evolution analysis

To explore the learning process of visual transformers, we propose analyzing how each weight block of the Visual transformer evolves and how the position of each block affects the learning process, measured at the end of each epoch. Let us consider the *n-th block* referring to the $n$-th attention layer, in order from input to output. Firstly, we want to know whether every single block is needed during the learning process or, in contrast, some blocks could be frozen to reduce training time without reducing accuracy. Secondly, we want to know if the variations of the blocks during fine-tuning produce a tangible impact on the accuracy of the transformer. To evaluate this, we propose two different criteria:

1. The first criterion is based on the sum of all the changes in the weights between two consecutive epochs, as follows:

$$C_1^k = \frac{1}{n^2} \sum_{i,j=0}^{n} \left| w_{i,j}^{k+1} - w_{i,j}^{k} \right| \tag{4.1}$$

   where $w_{i,j}^k$ is the element in row $i$ and column $j$ in epoch $k$, of the weight matrix $W$.

2. The second criterion is based on the evolution of the sum of all the elements in the matrix and the standard deviation of all the changes in the matrix elements:

$$C_2^k = max\left( \frac{1}{n^2} \left| \sum_{i,j=0}^{n} (w_{i,j}^{k+1} - w_{i,j}^{k}) \right|, \right.$$
$$\left. \frac{c}{n^2} \sqrt{\sum_{i,j=0}^{n} \left( (w_{i,j}^{k+1} - w_{i,j}^{k}) - \bar{W} \right)^2} \right) \tag{4.2}$$

   where $w_{i,j}^k$ is the element at row $i$ and column $j$ in the epoch $k$ and $\bar{W}$ is the mean of the values $w_{i,j}^{k+1} - w_{i,j}^{k}$ $\forall i,j$, and $c$ is a constant to determine the weight between both terms (in our case, we used $c = 0.05$).

We use these two criteria to decide when a weight matrix of a transformer stops learning and, thus, can be frozen. We compute the respective criterion for each matrix at the end of each epoch and if the value of the criterion is under a predetermined threshold (defined in Section 4.5.2), then the correspondent weight matrix is frozen.

## 4.5 Validation of Transformer Optimization

### 4.5.1 Dataset

To test our hypothesis and analyze the learning process, we will fine-tune each transformer with the Food-101 dataset [10]. This dataset consists of 101 different food categories and contains 1000 images for each category. Then, we will employ 750 images for training and 250 for testing. We choose this dataset as it represents a complex domain where transformers can illustrate their potential to achieve excellent recognition results.

### 4.5.2 Implementation setting

In our experiments, we fine-tuned the transformers during 50 epochs, we used a batch size of 32 and a learning rate of $10^{-3}$ with Stochastic Gradient Descent (SGD) as optimizer. Models pre-trained with images of size $224 \times 224$ have been employed as well. Finally, for criterion $C_1$ (Equation (4.1)) we set the threshold at $5 \cdot 10^{-4}$ and for the criterion $C_2$ (Equation (4.2)) we set the thresholds at $5 \cdot 10^{-3}$.

### 4.5.3 Results and analysis

| | Accuracy | | | Training time | | |
|---|---|---|---|---|---|---|
| Model | No | C1 | C2 | No | C1 | C2 |
| ViT | 88.58 | 88.78 | **89.70** | 71 632 | **44 932 (37%)** | 49 545 (31%) |
| BEiT | **90.57** | 90.17 | 90.13 | 96 995 | **60 843 (37%)** | 66 267 (32%) |
| DeiT | **87.59** | 86.35 | 86.54 | 74 730 | 48 873 (35%) | **43 617 (42%)** |
| SWin | 91.10 | **91.62** | 90.56 | 114 172 | **74 059 (35%)** | 84 121 (26%) |
| CSWin | 89.51 | **89.60** | 89.41 | 126 716 | 114 998 (9%) | **81 982 (25%)** |

TABLE 4.2. Results obtained on dataset Food-101 for transformers: without weight blocks freezing (No), and with C1 and C2 criteria. Accuracy is given in percentage and training time in seconds. Best results are given in bold.

We run the experiments on five of the most popular and recent transformers (ViT, DeiT, BEiT, SWin, and CSWin) and applied both criteria to freeze their weight blocks. The results are given in Table 4.2. As one can see, freezing the blocks, in 3 out of 5 cases, noticeably reduces training time up to 42% compared to the standard transformer training process. We also see an improvement

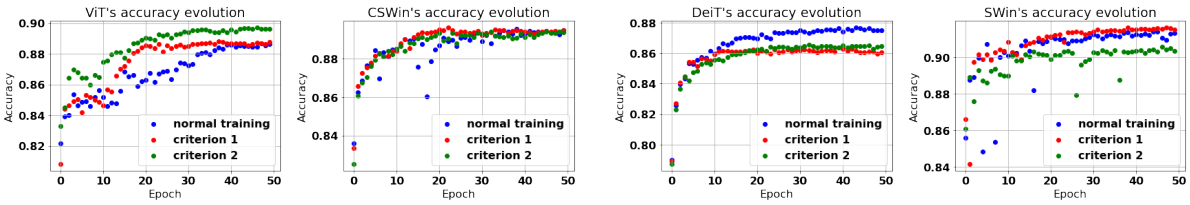



FIGURE 4.1. Accuracy evolution of ViT, CSWin, Deit and SWin during fine-tuning on Food-101 with standard training and applying $C1$ and $C2$.

in accuracy only on three of the transformers, namely ViT, SWin, and CSWin. As can be seen in (Figure 4.1) and Table 4.2, SWin remains the most accurate on the Food101 dataset, while ViT is still the less accurate, but with our criteria having a strongly positive effect on the training process improving its performance and results continuity (Figure 4.1 left). $C_1$ turns out to be especially useful for the SWin and CSWin achieving better performance (up to 0.52%), results continuity, and time reduction up to 35%. In the BEiT, the $C_1$ criterion reduced by 37% the time spent sacrificing the performance just by 0.4. For ViT, both criteria led to increased performance with up to 1.12 (Figure 4.1 left) and reduced time up to 37%. For DeiT, we achieved the highest time reduction (42%) although the performance was reduced by 1. Another benefit of applying our freezing criteria is that transformers learn in a more stable way as we can observe in Fig.4.1, according to the shape of the learning curves for standard training and with the application of our two criteria. Following, we illustrate the performance of the transformer showing plots for DeiT and SWin, achieving the best and the worst of our results on the Food101.

Both criteria ($C_1$ and $C_2$) are very useful because they give valuable insight into how the transformers learn over time and what is the behavior of the different blocks, in particular, how the position of the block influences the training process. The criteria indicate for all transformers clearly decrease toward zero during the training phase which shows good weight block convergence (Figure 4.2 and Figure 4.3). We can see that the behavior is not the same for all the blocks since their position in the pipeline matters. In most of the transformers (e.g. DeiT), the first blocks are less prone to change than the latter blocks. The only transformer where the first blocks change more than the last ones was the SWin (Figure 4.2 and Figure 4.3, right), with an exception for block 0. We can make some consideration also on the smoothness of these curves corresponding to the different blocks. SWin presents the largest fluctuations and DeiT the smallest ones during the optimization process. This fact can explain why the SWin transformer better adapts to the data and achieves the highest performance. For the other transformers, we observed that the matrices presented a very similar behavior to the DeiT during epochs showing moderate and continuous convergence and data adaptation capacity.

The next question is if our criteria are optimal with respect to possible alternatives. We considered alternatives: the Frobenius matrix norm, the matrix rank, and the Singular Value Decomposition (SVD). On all 5 transformers, we observed that the rank did not change during

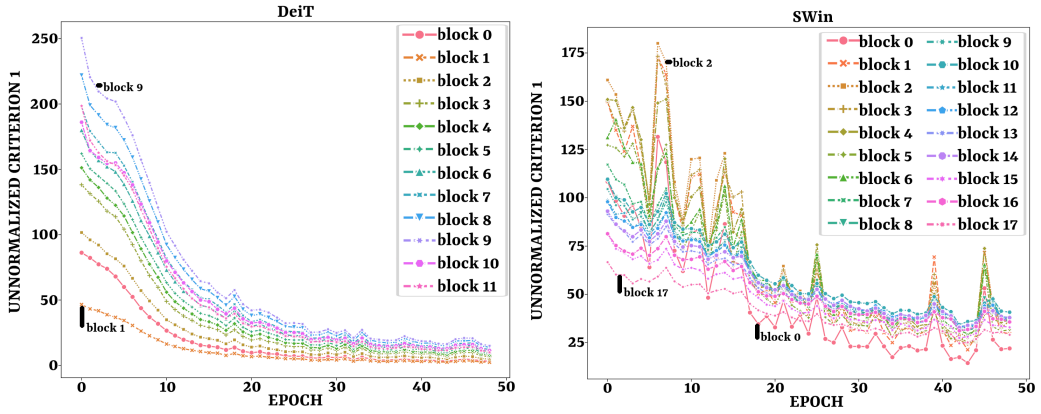



FIGURE 4.2. $C_1$ criterion on $W_v$ matrices for Deit and layer 2 of SWin transformer during fine-tuning on the Food-101 dataset.

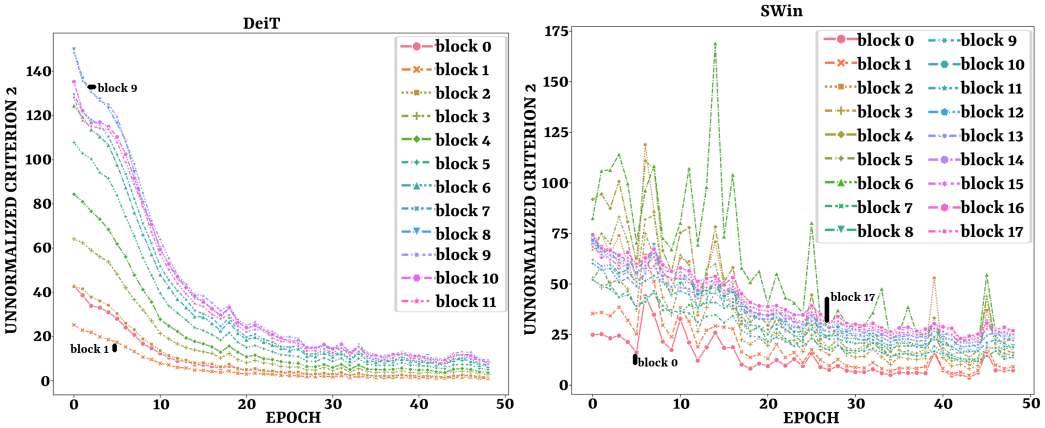



FIGURE 4.3. Maximum value of the $C_2$ criterion on $W_q$ matrices for Deit and layer 2 of SWin transformer during fine-tuning on the Food-101 dataset

the learning process. Moreover, there is a very slight change in the Frobenius norm of the weight matrices (Figure 4.5), so they cannot be used as a criterion to monitor the learning process.

We applied the SVD, to study the properties of the weight matrices. We noticed that the singular values almost remain constant during the training process for the different matrices (Figure 4.4 left). The biggest change on the singular values was observed in SWIN on layer 0 where the biggest first 50 singular values slightly increased during the training. For this reason, we believe the evolution of the singular values is not a good measure if there is a need to freeze the blocks. On the other hand, we see that the last third of the less important singular values have a really small magnitude that makes us think that they could be disregarded. This matrix transformation could have a good effect on improving the overfitting of the transformer but add additional cost to apply the SVD on the weight matrices.

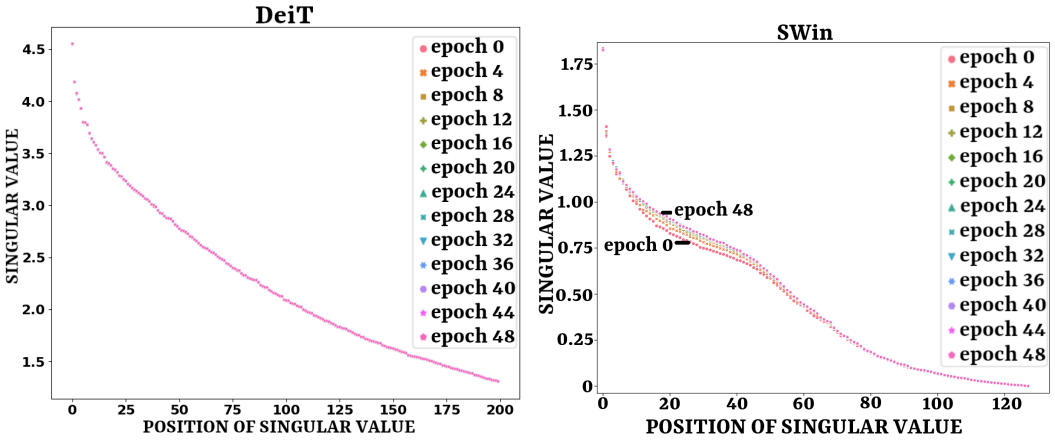



FIGURE 4.4. Evolution of the singular values of $W_v$ matrices of DeiT and layer 0 of SWin transformer on the Food-101 dataset. On the x-axis, the index of the singular values, and on the y-axis its corresponding value.

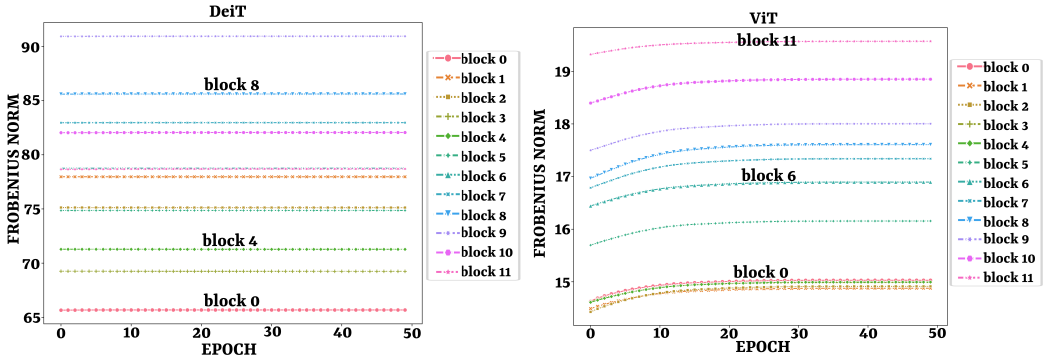



FIGURE 4.5. Frobenius norm of the weight matrices as a function of the epoch for the MLP of DeiT (left) and $W0$ of VIT (right).

## 4.6 Conclusions on Transformer Optimization

In this work, we have shown it is worth exploring if a single layer is still learning by studying the convergence of its weight matrices, in transformers. This can be used to reduce time execution and even improve final performance by using *stopping* criteria. We have proposed two different criteria and we noticed that freezing non-learning layers, in all five transformers, has reduced over-fitting and noise, and has saved time during training. The proposed approach has been able to reduce training time by up to 42%, on average, for all the five transformers under analysis. The position of the blocks also matters where in most transformers the first blocks learn less. There was no substantial difference in the behavior of different types of weight matrices ($W^Q$, $W^K$, $W^V$, MLP, and $W_0$). Also, our criteria allow to illustrate the convergence of the learning process as well as give insight into its continuity and speed.

# PART II

## Deep Learning in Digital Health

CHAPTER 5

# DIGITAL HEALTH

*This idea that we're going to just scale up [...]*
*and eventually* ***human-level AI*** *will emerge...*
*I don't believe this at all, not for one second.*

YANN LECUN

In the current millennium, Artificial Intelligence (AI) is helping humans automatize and optimize complex tasks. We have seen it employed in industry, as well as transport and information. However, AI closely regards the field of health as well, nowadays commonly known as *Digital Health* (DH) (Figure 5.1). In the last decade, we have seen machine learning applications popping up all around the medical processes, sometimes even outperforming humans [26], applied in an extensive range of healthcare domains, such as cardiology and general internal medicine, mental health, general surgery (especially oncology), but also prevention[1].


FIGURE 5.1. Healthcare transformation, according to the University of Sydney. Source: Youtube.

Machine learning finds fertile ground in cardiology [18] because it is one of the branches of medicine that produces the largest amounts of data. In fact, many different kinds of diagnosis machines are used for studying the heart. We can see applications for visualizing congestive heart failure in chest radiographs, discriminating hypertrophic cardiomyopathy from physiological hypertrophy in echocardiography of athletes, or detecting and classifying myocardial delayed enhancement patterns in Magnetic Resonance Images (MRI). Researchers and psychiatrists are success-

[1] Source: Forbes.

fully identifying mental health issues, using the power of AI[2]. For example, they apply neural networks to electroencephalograms, MRIs, voice registrations, etc. for detecting brain syndromes, and identifying the best treatment for the patient [106]. There are tools for automatically diagnosing dementia, Alzheimer, and other psychiatric illnesses. But also neural networks that can help treat psychological problems like depression and help prevent suicide and violent behavior. These new technologies become extremely impressive when they are applied to surgery [87]. Machine learning is able to optimize preoperative planning and intraoperative performance in various surgical specialties. Leading hospitals are already installing AI tools in their daily practice, showing improvements in surgical outcomes and patient safety. There is a technology in the course of development that, coupling AI with Virtual Reality (VR), allows a surgeon to view in real-time a 3D hologram of a solid part (such as a bone) of the patient's body from a CT and superimposed the hologram on the real body of the patient, in order to fully visualize relevant anatomy during surgery [3].

In the last years, AI in cancer detection and treatment captured the attention of health institutions and governments. The range of potential applications seems endless and, at this moment, a lot of that investigation is focusing on tools for cancer imaging. Machine learning algorithms are spotting early cases of cancer and, in various fields, are reaching human-like levels [14]. From three-dimensional images of the whole body to microscopical pictures of cancer cells, doctors apply neural networks in many ways. The main goals are finding cancer at its earliest stages and determining the stage of the tumor. But the same tools can be used to check if treatment is working, or to monitor whether the area affected by cancer has reduced or increased after treatment. In radiology, these recent advantages are having a strong impact on early diagnosis as much as on the reduction of misdiagnosis and therapy-related complications. There are benefits both in the short and long term, including saving the patients from overtreatment and extending their life expectancy. Apart from the detection of nodules and lesions from medical images, there are also machine learning algorithms able to detect cancer from biomarkers [128]. Their usage in fields like liquid biopsy is rapidly increasing [72]. Pharma companies are already using these technologies in drug discovery [21]. The aim is to accelerate the research process and reduce the risks to patients during clinical trials. Compared with other fields, digital health is the slowest one in installing new technologies in clinical practice. The first thing slowing down digital health is related to certifications. This is (and must be) a very thorough evaluation process because the safety of the patients may depend on this, but it is overly slow. In Europe, there are only 32 notified bodies under Medical Device Regulation (MDR), overflowed by work, and they need from about eight to eighteen months to release a CE mark for medical devices. This comes at an overall cost of the order of a hundred thousand euros, for a single certification of a device. Another reason is the difficulty to access medical data, which are crucial for training neural networks. Hospitals have the duty of protecting personal patient data and keeping them

[2]Source: IEEE innovation at work.
[3]Source: Frontiers.

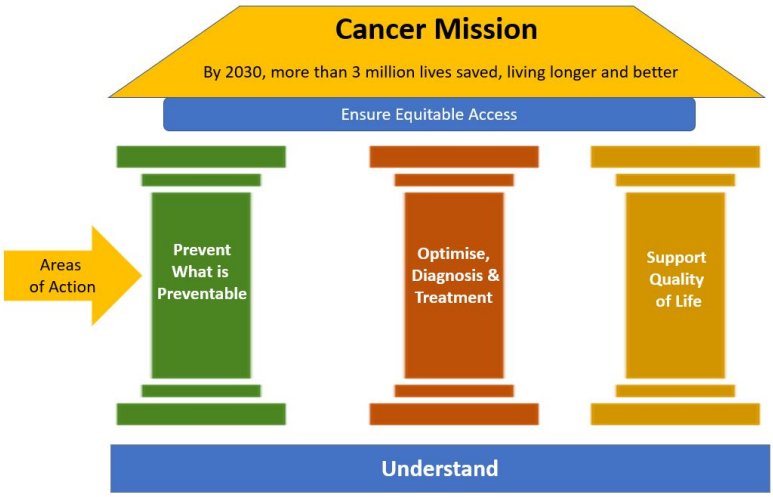



FIGURE 5.2. Illustration outlining the areas of intervention for action in Cancer Mission by Horizon Europe.

secured, in respect of the ISO[4] norms and the GDPR[5]. Therefore, being able to access these data, even for research use only, is very difficult and time-consuming. Hospitals and clinics give much value to patients' personal data. In fact, they sell them for very large amounts of money[6]. Nevertheless, these are not complete stoppers. European Union is now implementing a central database of medical records called European Health Data Space that will, among other things, provide universities and research centers with the data they need for developing new and better technologies in the medical field. Furthermore, European Union is supporting an initiative called *Cancer Mission*[7] that aims at saving lives and improving the wellness of 3 million people, by 2030 (Figure 5.2). Among the recommendations given by the board's draft, a good part of them are related to cancer prevention, screening, and early detection. Topics on which we tried to give our contribution, and explained in this part of the thesis.

At the present time, we are seeing, not only in the youngest ranks of doctors and clinicians, a strong willingness to bring new technologies to the standard of care. It is a common belief that AI technologies in healthcare can reduce mortality. Some guess it could reach a level sufficient to reduce early deaths by a percentage from 10 to 30%[8]. A 20% reduction in the 8 million total annual deaths in Europe in 2019, would have meant saving 1.6 million people. In other words, the population of the city of Barcelona.

[4] International Organization for Standardization
[5] General Data Protection Regulation
[6] An insight from Scientific American.
[7] Cancer Mission is part of Horizon Europe
[8] Sources: Forbes, Towards Data Science.

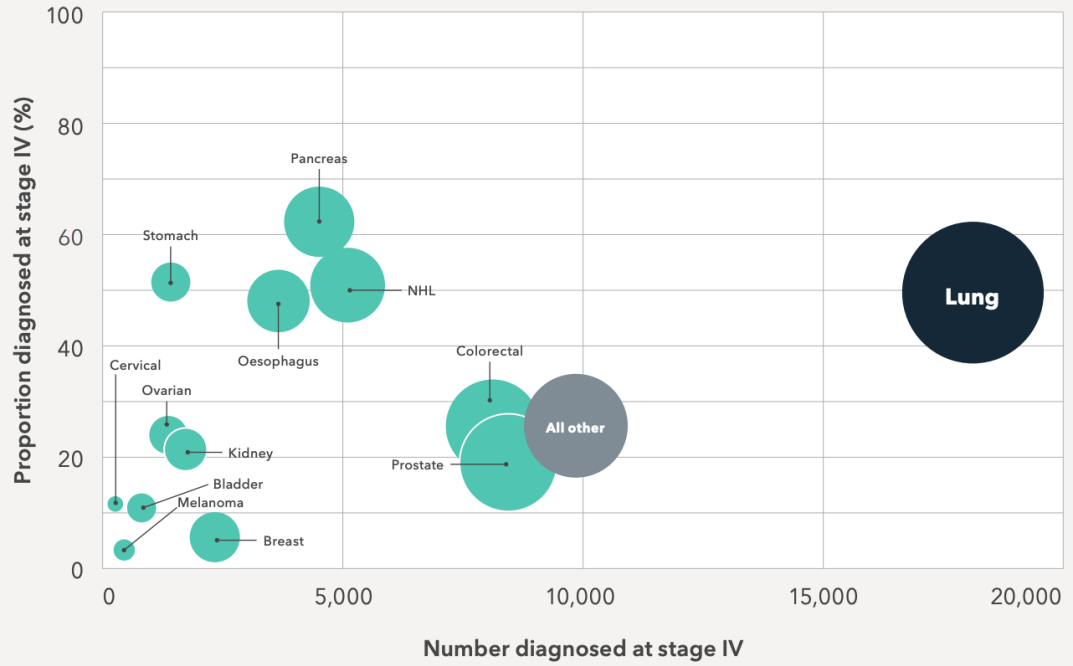



FIGURE 5.3. Plot of the number of diagnoses at stage IV in relation to its occurrence percentage. The size of the circles shows the relative weight of each cancer type in terms of its contribution to the total number of cancers detected at an advanced stage. Update of the original figure produced by the United Kingdom Lung Cancer Coalition (UKLCC). Data from Public Health England, 2018. From Health Policy Partnership

## 5.1 AI for Lung Diseases

In 2020, more than 2.2 million people were diagnosed with lung cancer, and nearly half a million were in Europe. It is the second most commonly diagnosed form of cancer, after breast cancer, and the most lethal one, in absolute numbers (Figure 5.3). Lung cancer is the leading cause of cancer deaths worldwide and approximately one in five cancer deaths globally are due to that. In the United States, the average of people still alive after five years after the diagnosis of lung cancer is just 22.6%[9]. The toll of this plague on the human side is huge but it is worth noticing also its economic cost. In the European Union in the year 2009, the cost of lung cancer, including care and productivity loss, was 18.8 billion euros. Equivalent to 15% of the sum of all the costs due to every other form of cancer[10]. The main reason is that the largest majority (around 75%) of the cases of lung cancer are detected in the late stages, when the possibilities of treatment or intervention are sensibly reduced.

Reading a lung CT is a tedious and very time expensive task. In assessing the nature of an eventual suspicious mass, radiologists follow special guidelines [41] [80] that suggest using

[9] Source: Lung.com.
[10] Source: Health Policy Partnership.

measure the volume of the nodule for a more precise diagnosis. However, the precise three-dimensional segmentation[11] of all the eventual lung nodules present in a CT is a task that requires around one hour in average [115]. Consequently, radiologists rely very often only on the diameter of the nodules, measured on the axial plane. This is a kind of approximation that carries an error up to 50%. Artificial Intelligence can help radiologists reduce the reading time of a CT from 20 to 33% [51], and add value with volumetric segmentation and automatic characterization with no other time loss. This would allow the clinicians to report more images or dedicate to critical tasks.

Our goal in this field is to apply the knowledge we collected in fundamental research and develop new architectures especially designed for lung image applications. We aim at releasing new tools for detecting and characterizing lung cancer, focusing in particular on early detection. When COVID-19 struck globally, given its impact on the world's society, we decided to use our knowledge to contribute to the research also on other kinds of lung lesions.

In chapter 6, we present our works on lung nodule segmentation, explaining how crucial this task is in chest CT reading and giving a perspective about the applications of our tool.

In chapter 7, we show the evolution of the previous work, namely a series of tools for lung segmentation, and COVID-19 lesion detection and segmentation, all organized in a workflow from the raw CT to the resulting images. This work has been on COVID-19 but has been designed to be applied to other common pulmonary diseases if needed.

[11] Segmentation means the exact definition of the area (or volume) of interest.

CHAPTER 6

*What I cannot create,*
*I do not* ***understand***

RICHARD FEYNMAN

# LUNG NODULE SEGMENTATION

## 6.1 Introduction to Lung Nodule Segmentation

Recent researches about cancer prevalence and incidence of mortality show that lung tumors represent so far the leading cause of cancer death. According to the data of GLOBOCAN that have been collected in 185 countries of the world, there were an estimated 18.1 million cancer cases and 9.6 million cancer deaths in 2018 and the 18.4% of these have been caused by lung tumors [11]. With an early diagnosis, faster treatment planning, and cleaner air in our cities, most of these deaths could be avoided. Nowadays, the most effective analysis that allows for spotting pulmonary nodules is Computed Tomography (CT). A 3D CT scan may contain up to 1000 slices and, therefore, identifying the nodules manually is often time-consuming and tedious. However, precise nodule segmentation is compulsory for the assessment of the characteristics that determine the malignancy of the tumor, such as shape, volume, and change rate. In addition, a fast segmentation will deliver a strong speed up to the procedures between diagnosis and radiotherapy, opening a scenario where the patients can go from diagnosis to irradiation on the same day.

Starting from these premises, we can infer that seeking a fast, fully automatized, and reliable nodule segmentation algorithm is of utmost importance. Because of their ability to learn complex patterns, Neural Networks (NNs) such as Convolutional Neural Networks (CNNs) appear to be highly promising instruments to achieve these goals. In the past years, many researchers handled this problem by using a CNN with several methods and gained different levels of accuracy, but there is still room for improvement. The modern approaches are based on networks such as AlexNet [64], VGG [101], and GoogleNet [107] that have demonstrated remarkable success not only in many different computer vision fields but also in medical image analysis tasks. The models most frequently used are inspired by UNet [99] and VNet [84].

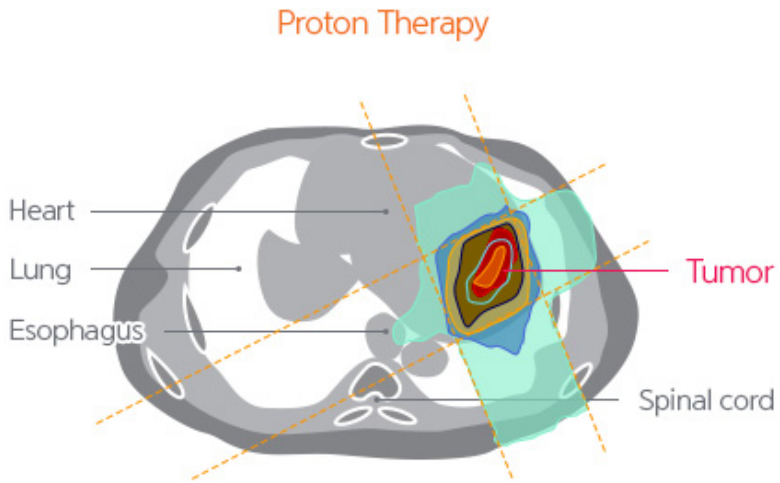



FIGURE 6.1. Proton therapy estimated dose deposition for lung tumor treatment. The red contour area is the segmentation of the tumor currently made by radiologists (Source: Samsung Medical Center).

### 6.1.1 Aim of Lung Nodule Segmentation

The final purpose of this research is to provide medical personnel with a reliable tool for automatic segmentation that can be applied to all possible cases and manage the task without the supervision of the final user. Our eventual goal is to build a full routine that can be used by specialists and clinicians for lung cancer detection and characterization. In this project, segmentation is a vitally important tool for the rapid extraction of all the physical properties of the nodule that can be used to estimate the malignancy of the tumor. An accurate segmentation can prevent a healthy patient from undergoing a useless biopsy and can detect nodule growth, even in cases where this is difficult to be detected by eye. Non-invasive therapies such as radiotherapy and hadrontherapy (Figure 6.1) need a millimetric accurate three-dimensional segmentation, in order to be implemented. With the current methods, this process can take up to several hours [115]. Instead, with the help of Convolutional Neural Networks, this could be done just in seconds.

This chapter is organized as follows. In chapter 6.1.2, we start with showing the actual state of the art concerning the most relevant methods of image segmentation in the case of lung tumor, categorized into automatic and semiautomatic nodule segmentation. In chapter 6.2, we present the new method of nodule segmentation developed in this work, showing the model architecture in chapter 6.2.1. In chapters 6.2.2 and 6.2.3, we describe the loss function and the fit to the segmentation mask used, respectively. In chapters 6.2.4 and 6.2.6, we show the dataset and the experimental methods used. Then, we present and discuss the results in chapters 6.3 and 6.4, respectively. In particular, we show the overall performance of the method in the bi-dimensional case in chapter 6.4.1, the evaluation of the database consistency in chapter 6.4.3, and the comparison of the method with the state of the art in chapter 6.4.4. Conclusions

and outlook are shown in chapter 6.5 and chapter **??**, respectively.

### 6.1.2 Relevant works in Lung Nodule Segmentation

Research in the field of lung nodule segmentation started long before the introduction of artificial intelligence and deep learning. During the previous generation, the existing methods were several, and they can be divided into two main classes: *region-based segmentation* and *edge detection*. The first ones are achieved by studying the homogeneity and similarity between pixels, instead edge segmentation is obtained by enhancing the edges with the use of differential operations or kernel convolutions. These methods are still used in current medical softwares and in research, in order to reinforce the results of deep learning algorithms. A deeper description of these methods is presented in Zheng *et* Lei [131].

With the introduction of deep learning, and in particular the structure encoder-decoder, the research flow in segmentation divided into three main branches: the methods based on *Generative Adversarial Networks* (GANs), resumed accurately in Kazeminia *et al.* [61]; the ones using reinforcement learning, of which Tian *et al.* [111] gives a modern overview; and the convolutional neural networks using residual connections, whose milestone is UNet [99]. In this chapter, we will briefly present some of the best current methods, mostly belonging to this last branch.

Nowadays, UNet is the benchmark for basic image segmentation. In most of the cases, it has shown both good properties of reliability and coherence. UNet is a CNN composed of an encoder and a decoder with four levels of depth. The strength of the UNet model lies in the connection between the respective layers of the encoder and decoder. In other words, the outcomes of every upsampling layer on the decryption side and every convolutional layer on the encryption side are concatenated and processed together. This allows to reduce the loss of resolution that one can encounter in a deep learning architecture due to the repeated convolution and the pooling layers, responsible for the generalization of the output.

#### 6.1.2.1 Automatic Nodule Segmentation

In recent years, the amount of literature on automatic nodule segmentation in CT had an appreciable increase. Wu *et al.* [120] presented one of the simplest structures that we can find for automatic nodule segmentation. They proposed a segmentation algorithm and a malignancy predictor utilizing an UNet structure with just half of the convolutional layers normally used. It has multiple window widths and window centers enriching the nodule information. They show an improvement over the standards of UNet of more than 2% in Dice Score Coefficient (DSC).

Aresta *et al.* [2] developed a model named iW-Net consisting of a concatenation of two UNet networks. It can be utilized with and without user interaction. In the first case, the user draws the nodule's diameter and the respective end-point extraction in order to generate a weight map,

which is then used for altering the prediction of the network. The algorithm has been designed by taking into account the expected spherical shape of the nodules. The weight map is then incorporated as a feature of the model and as a component of the loss function.

A lightly modified version of UNet is the one presented by Keetha *et al.* art:keetha. Their method consists in a UNet structure where the residuals are filtered by more hidden layers situated on the connections between the encoder and the decoder. These hidden layers are further connected with one another. They provide only results regarding Dice score and sensitivity with a good average. They apply the Mish activation function, demonstrating its value with an ablation study.

Zhou *et al.* [133] use another modified version of UNet, similar to the one proposed in the previously discussed article. Here the authors propose many arrangements of the hidden layers, using different shapes of the residual connections and levels of depth. They use a loss function that is a mixture between cross-entropy and soft dice coefficient and do an extensive analysis on six different datasets, including LIDC-IDRI. Due to our interest in this work, we replicated it and compared it with ours in section 6.4.4.

Tang *et al.* [110] proposed an end-to-end 3D Deep-CNN called NoduleNet for solving pulmonary nodule detection, false positive reduction, and segmentation jointly. They employed an UNet-like model that firstly calculates the bounding box of the nodule and then runs a segmentation refinement only on the Volume of Interest (VoI) surrounding the bounding box, progressively up-sampling the cropped volumes and concatenating them with low-level semantic features. This method attempts to solve the loss of resolution inside the VoI due to repeated convolutions and the pooling layers of the image.

Hancock and Magnan [46] proposed a method based on the vanilla level set image segmentation method but, instead of designing the velocity function manually, they use machine learning regression methods in order to learn these parameters. The Central focused Convolutional Neural Network proposed by Wang *et al.* [117] is a network formed by two different parallel branches receiving the input image in 2D and 2.5D[1], respectively. After the elaboration, they concatenate the resulting images for the last layer, where the features are mixed and the nodule mask is provided as a probability map. Their idea of using a central pooling layer is appreciable because it pools the image without reducing the resolution around the VoI, thus mitigating one of the most relevant problems of deep learning with CNNs.

In a similar work, Cao *et al.* [13] obtained almost the same results as Wang *et al.*. The main differences in Cao *et al.* are the use of residuals, which assume more importance, the use of two slightly different pooling layers, and the ResNet convolutional blocks that make this network a deep learning network for all the intents. Another added value consists of a strong post-processing that makes the algorithm able to gain another 0.2% in dice score accuracy.

Huang *et al.* [54] provide a fully automatic routine for lung nodule detection and segmentation.

[1] In this case, it receives an input of only three slides on the third axis and not all the CT or a cubic tensor.

Their architecture for segmentation is a composition of a Faster Regional-CNN (for encoding) and a Fully Connected CNN (for decoding) with a *VGG16* ([101]) backbone, that generally proved to be a reliable method for reconstruction. Especially, if used in fine-tuning, as done in this work. The loss function is a composition between cross-entropy and L1 distance. The input image size is $64 \times 64$. Similarly, Qian *et al.* [93] use the *VGG16* structure for encoding but introduce a new kind of decoder, called pyramid deconvolutional neural network. This takes the outputs of each level of depth of the encoder and composes them, with the use of deconvolution, in order to obtain the nodule mask. In this way, they reduce the resolution loss coming from a series of stacked convolutions. The stride used for deconvolution has an impact on the balance between sensitivity and precision, allowing the user to tune it as required. They work exclusively on lung nodule segmentation.

Finally, we found remarkable the work done by Jiang *et al.* [57], who developed a Multiple Resolution Residual Network as an extension of the ResNet [48] likewise based on a model similar to UNet.

#### 6.1.2.2 Semi-Automatic Nodule Segmentation

Part of the literature is dedicated also to hybrid systems combining image segmentation and the supervision of the user in order to optimize the results. Messay *et al.* [83] presented a hybrid system, based on a regression Neural Network requiring the user to input eight precise points in order to create the mask.

We also want to highlight the research of Liu *et al.* [73], who developed a network for Juxta-Pleural Lung Nodules but only tested it on 50 manually chosen images, making any comparison unfeasible.

Roy *et al.* [100] developed a shape-driven lung nodule segmentation. During the pre-processing, they applied a mask[2] erasing all the non-soft tissues around the lungs. This makes the Juxta-Pleura nodules way easier to be segmented, but difficult to be applied in real clinical conditions due to the unavailability of these masks during the analysis of the patient's data. Furthermore, they chose a subset of the test set, making the comparison with their results impaired.

## 6.2 Methods for Lung Nodule Segmentation

### 6.2.1 Model Architecture

The method we are proposing is inspired by the UNet model, but with some substantial modifications. It has one encoder and two decoders with only two levels of depth. In this new structure,

[2]Lung masks computed by using an automatic segmentation algorithm and provided by LUNA16 challenge.

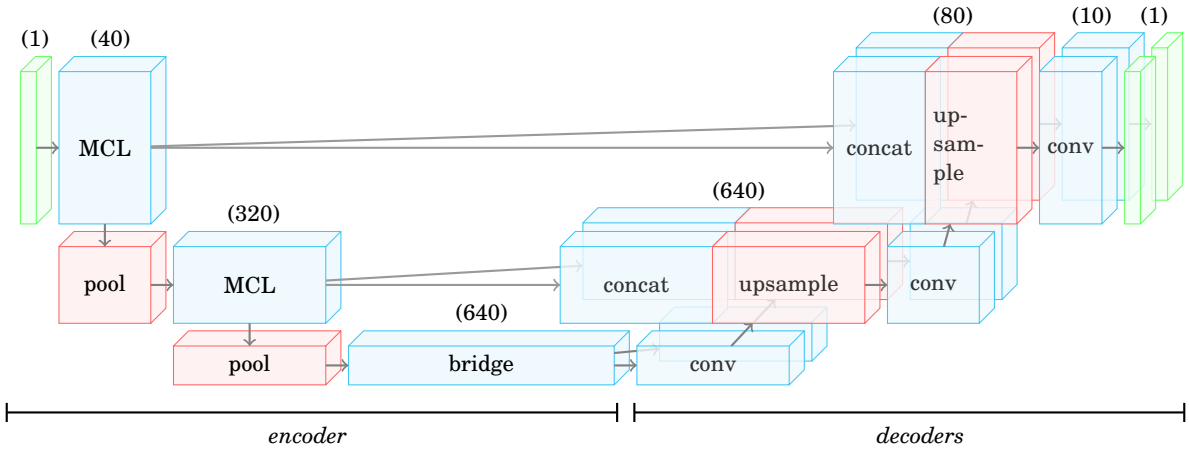



FIGURE 6.2. Structure of our NN, with one encoder and two decoders. The name on each sample corresponds to the operation applied in order to obtain that sample. The last step is always a convolution. In parentheses, the number of filters appears. Input and output, re-scaling and convolutional layers are indicated in green, blue and red, respectively.

the first two layers on the encryption side are substituted by two Multiple Convolutional Layers (MCL), inspired by the Inception-v4 architecture. Inside those, the input tensor is given to the four different branches and convoluted a different number of times with $3 \times 3$ kernels. Only in one case over four, the tensor is not convoluted but just replicated and average pooled.

| | size | input dim | filters |
|---|---|---|---|
| CT image | $64 \times 64$ | - | 1 |
| 1st MCL | $64 \times 64$ | 1 | 40 |
| 1st pool | $32 \times 32$ | 40 | 40 |
| 2nd MCL | $32 \times 32$ | 40 | 320 |
| 2nd pool | $16 \times 16$ | 320 | 320 |
| bridge | $16 \times 16$ | 320 | 640 |
| 1st conv | $16 \times 16$ | 640 | 320 |
| 1st upsample | $32 \times 32$ | 320 | 320 |
| 2nd conv | $32 \times 32$ | 640 | 40 |
| 2nd upsample | $64 \times 64$ | 40 | 40 |
| 3rd conv (1) | $64 \times 64$ | 80 | 10 |
| 3rd conv (2) | $64 \times 64$ | 10 | 1 |

TABLE 6.1. Scheme of the layers and the respective input and output number of filters.

Thereafter, all the results are concatenated such that every branch of the MCL contributes to 1/4 of the total number of filters. The structure of the multiple convolutional layer is shown in Figure 6.3. Moreover, the depth of the network has been chosen to be made by just two levels because of the limited dimensions of the input images. As shown in Figure 6.2 and with the details in Table 6.1, the network possesses a first encryption part (also referred to as *encoder*), where the CT image scaled to the interval $[0,1]$ is convoluted inside the MCLs and pooled. This operation is repeated twice with a decrease in the size and an increase in the filters' dimensionality. The bridge operation is essentially a convolution. All the convolu-

tions are carried out with a $3 \times 3$ kernel and followed by batch normalization and an activation function.

The second part of the network consists in the decryption part (or *decoder*). It receives the output of the encoder coming from the bridge and produces an image with the same dimension as the input.
As we will explain in paragraph 6.2.3, it can be useful to obtain two or more different outputs. In our case, inside the network, we employ two decoders, each one with different activation functions. One decoder has ReLU rectifiers, while the other one uses softsign[3] functions. The last convolution of every decoder has been rectified with a ReLU layer, in order to ensure a positive result.
The two masks obtained are the $\alpha$ and the $\delta$ of Equation 6.2, respectively. In the last step, the network output is converted into a binary mask with a fixed threshold of $T = 0.5$. The gradient of this operation is not calculated and then not included in back-propagation because the derivative of the step function is always zero and singular in one point.

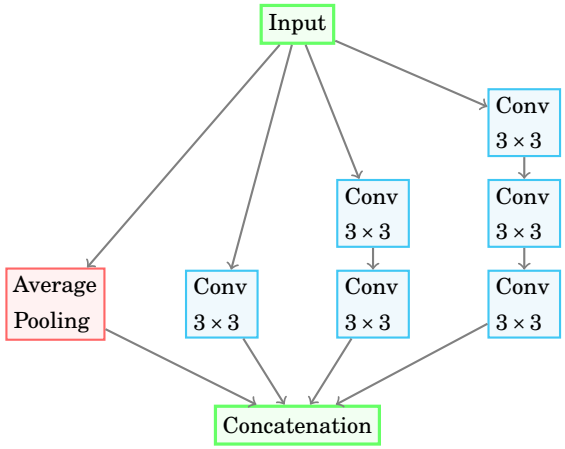



FIGURE 6.3. Structure of the MCL inspired by Inception-v4.

### 6.2.2 Loss function

In the nodule segmentation literature, we can frequently find custom loss functions created so as to take into account the characteristics of the nodule, focusing on special features considered the most interesting, such as the borders, the volume, and the shape. A widespread problem of segmentation networks is the lack of precision at the borders. This is caused not only by the loss function itself but also by the ground truth masks provided by the training datasets that, in many cases, are not very precise. Forcing the network to strictly learn from an imprecise ground truth often brings to over-fitting or over-learning. Therefore, we propose using a loss function inspired by the *Mean Square Error* (MSE) function with tolerance on the pixels near the borders of the nodule. Our function does not simply calculate the loss element-wise between the produced image and the ground truth. It measures the loss between one pixel of our image and all the pixels around the corresponding one in the ground truth, inside a specific area. Then, it takes the minimum of those values. The radius of that area is $\Delta \leq f(d_{nod})$, where $d_{nod}$ is the diameter of

[3] Very similar to $\tanh(x)$, it proved to be more effective for our purposes.

the nodule and $f$ is a function chosen by the user. We adopted a function

$$f(d_{nod}) = \begin{cases} 0 & \text{if } d_{nod}/n < 64 \\ 1 & \text{if } d_{nod}/n < 16 \\ 2 & \text{elsewhere} \end{cases}$$

where $n$ is the total number of elements of the image and the unit of distance is in pixels. Moreover, it is coupled with an exponential preventing under-segmentation. Namely, our loss function is:

$$L(z,\hat{z}) = \frac{1}{n}\sum_{i}^{n}(\hat{z}_i - z_i)^2 \beta^{(\hat{z}_i - z_i)}$$

where $z_i$ and $\hat{z}_i$ are the values of the $i$-th pixel of the mask and the ground truth, $\beta$ is a hyper-parameter decided by the user and $(\hat{z}_i - z_i)^2$ is the term that ensures the convergence. The values of $z_i$ and $\hat{z}_i$ are always in the interval $[0,1]$. Thus, if $f(d_{nod}) = 1$, the calculated loss on pixel $z_{i,j}$ is

$$\begin{aligned} L(z_{i,j},\hat{z}_{i,j}) = min\{&L(z_{i,j},\hat{z}_{i,j+1}), L(z_{i,j},\hat{z}_{i+1,j}), \\ &L(z_{i,j},\hat{z}_{i,j-1}), L(z_{i,j},\hat{z}_{i-1,j}), \\ &L(z_{i,j},\hat{z}_{i,j})\} \end{aligned}$$

Because of the fact that our loss function has compact support on $\mathbb{N}$ and it is continuous and derivable, we can conclude that it is also Lipschitz continuous. From Goodfellow *et al.* [38]: *In the context of deep learning, we sometimes gain guarantees by restricting to functions that are either Lipschitz continuous or have Lipschitz continuous derivatives*. Considering that we must take into account the sign of the exponent and taking $\beta > 1$, the result of the exponent for the single pixel will be larger than 1 in the case of under-segmentation and smaller in the case of over-segmentation. In this way, over-segmentation is encouraged during the convergence to zero. Without this idea, there would be the risk of not segmenting the small nodules or losing some details.

### 6.2.3 Fit to the Segmentation Mask

The new loss function helps to improve the results, but it still does not reach the necessary accuracy. Until now, most of the CNNs for nodule segmentation try to reconstruct the mask directly, some of them with good results but still with some limitations. Our purpose is to reconstruct the exact shape of the cancer nodule and visualize it as a binary mask. Instead of letting the network do it end-to-end, we decided to use the image itself as a starting point of the segmentation algorithm and the network only for creating a mask of all the elements that we want to exclude, namely a way to fit the mask image. To formalize, our mask $M$ (before thresholding) is of the form:

$$M = I - \alpha \cdot I \tag{6.1}$$

where $\alpha$ is the output of the CNN (with the same shape of $I$), $I$ is the input image, and '$\cdot$' is the element-wise product. By implementing this procedure, we do not need to consider the precision at the borders, because the morphology of the mask comes directly from the image. Ideally, the network should solely identify the matter that is not part of the nodule. Since air and very soft tissues are already under the threshold limit, we expect significantly good results, especially on isolated nodules. In addition, we have to deal with nodules laying on the pleura or that are partially enclosed in other objects. We propose to use the following mask:

$$M = I - \delta - \alpha \cdot I \tag{6.2}$$

where $\alpha$ and $\delta$ represent two different outputs of the same network. We obtained very high indices of F1 score and IoU with this type of mask, more than 2% on average with respect to the ones obtained with the mask in Equation 6.1. The parameters $\alpha$ and $\delta$ can be calculated with the decoder using ReLU and softsign activation functions, respectively.
Our interpretation is that $\delta$ gives a larger contribution in the detection of the nodule in the image, deleting all the eventual other masses that have been segmented.

### 6.2.4 Dataset

For this work, we used a public dataset to show the performance of our proposal. We chose the *LIDC-IDRI*[4] dataset, described in Armato *et al.* [3]. It consists of 1018 CTs done on 1010 patients from seven distinct institutions. Each of these tomographies has been reviewed by four radiologists who independently detected and manually segmented all the nodules with a diameter larger than three millimeters. This makes a total number of 2669 segmentations (or masks) but, following Jiang *et al.* [57], we consider just the nodules which have been confirmed and delineated by all of the radiologists and that present an average malignancy index greater than 3. According to the aforesaid article, the total number of these nodules should be 529. Looking closer at the dataset, we discovered that there were totally different interpretations on some nodules and that the mean of these masks was producing unrealistic results. We settled a threshold regarding the overlapping at $IoU = 40\%$ and another one on the diameter. In this way, if one mask has a diameter at least 40% larger than another one or if their overlap is smaller than 40% of their total area, they cannot be considered as the same nodule.

The diameter threshold has been applied exclusively on the nodules with a diameter larger than $20\,mm$. This last correction regards also nodules with more than four masks. There have been some ambiguous cases[5] of large masses that led to an erroneous result, where some radiologists diagnosed the presence of a single big nodule and others recognized multiple separated nodules apparently attached to healthy tissue. This causes ambiguity and casts reasonable doubts on the nature of the mass. These are just a few cases that could jeopardize the attempts of the network

[4] Website of the public LIDC-IDRI dataset.
[5] These are six cases in particular.

to learn the right pattern. This is the reason why our database consists of only 493 nodules whose mask is decided with a 75% consensus over the four different segmentations. This means that a pixel is included in the ground-truth image only if three or more radiologists have selected it. This will be discussed further in paragraph 6.4.3. This constraint ensures also a reduction of over-segmentation because we noticed that a consistent part of the mask had been drawn to enclose the whole nodule in a way that includes more than the actual shape of the nodule. An example of a misleading segmentation is in Figure 6.4. As you can notice, the four radiologists segment the nodule with different criteria. All the 3D tensors representing the CTs have been pre-elaborated by operating methods such as clipping, normalization, and tri-linear interpolation. As a result, we obtained a homogeneous resolution on every axis and a clearer definition of the objects. The validity of the results has been verified visually for every single nodule in three dimensions.

### 6.2.5 Evaluation Statistics

The output of the Neural Network is a tensor with values between zero and one. After post-processing, using a threshold set to 0.5, this is transformed into a binary mask where 1 represents the nodule and 0 is the background. At that point, it should be compared with the ground-truth pixel by pixel. The most used criterion in this type of problems is the *Sensitivity* ($S$). It measures the percentage of pixels where the ground truth is 1 and it is correctly predicted (or true positive divided by the relevant elements). High sensitivity is one of the goals, but it also means over-segmentation. That is why it is usually counteracted by its counterpart counting the percentage of the true positives over the total predicted positives and, therefore, it estimates the under-segmentation, namely the *Precision* ($P$). Their *reduced mass* is proportional to the *F1* (or Dice similarity coefficient - *DSC*). In general, in order to evaluate the goodness of these algorithms, we can use the triple:

$$S = \frac{G_+ \cap M_+}{G_+} \qquad P = \frac{G_+ \cap M_+}{M_+}$$
$$F1 = 2\frac{S \cdot P}{S + P},$$

where $G$ is the ground truth, $M$ is our mask and $_+$ is the index indicating only the pixels where the values of the tensors are one. Another widespread evaluation criterion is the *Intersection over Union*. It is described by:

$$IoU = \frac{G_+ \cap M_+}{G_+ \cup M_+}.$$

It does not provide more information than F1, but it is extensively used. The good quality of Intersection-over-Union (IoU) is that it is invariant if we swap the entries: $IoU(G,M) = IoU(M,G)$. We applied this criterion also during the pre-processing in order to exclude all the masks with a low agreement between the four radiologists. The last evaluation criterion that we propose is called $\gamma$-index test by Low *et al.* [78]. It is very commonly used in medical fields like

radiotherapy. We utilized a simplified version consisting in a comparison of our result with the ground-truth mask with one-pixel tolerance. The data are binary images where 1 indicates the nodule and 0 the background. Since we impose a one-pixel tolerance, the procedure is repeated for all the adjacent pixels of the corresponding one. At this point, the $\gamma$-index pass rate is:

$$\gamma = \frac{\#passed}{\#pixels},$$

or, in words, the number of pixels that passed the test divided by the total number of pixels. This test provides a quantitative estimation of how many pixels have an error larger than the resolution. The aforesaid methods do not supply this information. For example, in one of our cases with a $\gamma$-index pass rate of 99.5%, we clearly derive that 20 pixels have an error larger than $R = 0.5 \div 1\,mm$.

As last, we used the *Kolmogorov-Smirnov* test in order to check the correlation between the data produced by the network and the ones produced by the radiologists and used for creating the ground-truth. Practically, this test is conducted to check if two sets of discrete data come from the same distribution. The two sets are plotted in a *Cumulative Fraction Plot* and then the maximum vertical deviation is measured between the two curves as statistics, $D$. Based on the number, $N$ of data points of each set, we establish a critical value. According to the tables:

$$D^* = \frac{1.36}{\sqrt{N}},$$

with significance level $\alpha_{ks} = 0.05$. If the statistics $D < D^*$, the test is passed and the two datasets are indistinguishable. Our idea is to reproduce a sort of Turing's test for seeing if one could discern the human-made and the software-made masks, looking only at the results. That limit could be considered the gold standard in this field.

### 6.2.6 Implementation Details

At the beginning of every single run, we randomly partitioned the dataset into training and test sets with a ratio of one to eight. Because of the limited number of data, we operated a data augmentation. At the beginning of each epoch, all the CT slices of the training test have been randomly rotated by an angle in the interval $]-180°, 180°]$ and randomly zoomed with a magnification included in $[\times 0.95, \times 1.05]$. The eventual padding has been made with black pixels (equivalent to zero).

In our experiment, the total number of learnable weights is 13 109 472 for a total of $\sim 50MB$. We trained our CNN 10 times for 300 epochs each. In order to support our expensive calculations, we used a machine owing an Intel Xeon E7-4830 v3 (i7) processor with 64GB of RAM. The GPUs are two Nvidia GeForce GTX 1080 Ti with 12GB of CUDA memory each. Everything was implemented in Python 3.7 with the usage of the Pytorch module. We decided to use a batch size of 1 so that we can give a more significant impact on the loss function at the expense of the

| run # | IoU (%) | Sensitivity (%) | Precision (%) | F1 (%) | $\gamma$-index (%) |
|---|---|---|---|---|---|
| 1 | $74.2 \pm 16.7$ | $80.4 \pm 17.8$ | $\mathbf{90.8 \pm 16}$ | $83.8 \pm 14.7$ | $99.20 \pm 1.03$ |
| 2 | $67.4 \pm 21.3$ | $78.9 \pm 20.1$ | $81.8 \pm 24$ | $78 \pm 19.9$ | $98.87 \pm 1.13$ |
| 3 | $\mathbf{76.6 \pm 12.3}$ | $\mathbf{90 \pm 9.46}$ | $85.2 \pm 14.8$ | $\mathbf{86.1 \pm 8.59}$ | $99.07 \pm 1.00$ |
| 4 | $72.6 \pm 15.9$ | $85 \pm 12.3$ | $84.8 \pm 17.8$ | $83 \pm 12.3$ | $99.14 \pm 0.84$ |
| 5 | $74.9 \pm 12.4$ | $82.3 \pm 11.9$ | $90.3 \pm 13.2$ | $85 \pm 9.26$ | $99.07 \pm 1.07$ |
| 6 | $68.2 \pm 19.7$ | $77.8 \pm 20.6$ | $84.1 \pm 21.4$ | $79 \pm 18.4$ | $99.0 \pm 1.32$ |
| 7 | $72.3 \pm 19.9$ | $78 \pm 21$ | $90.7 \pm 15.9$ | $81.8 \pm 18$ | $99.08 \pm 0.89$ |
| 8 | $74.3 \pm 17$ | $84.8 \pm 16$ | $86.3 \pm 16.7$ | $83.9 \pm 14.4$ | $99.31 \pm 0.66$ |
| 9 | $72.8 \pm 18$ | $81.5 \pm 17.8$ | $88.2 \pm 12.7$ | $82.7 \pm 15.6$ | $98.97 \pm 1.24$ |
| 10 | $72.9 \pm 19.5$ | $84.1 \pm 20.3$ | $83.9 \pm 21.1$ | $82.3 \pm 18.7$ | $\mathbf{99.34 \pm 0.69}$ |
| Average | $72.6 \pm 2.7$ | $82.3 \pm 3.6$ | $86.6 \pm 3.0$ | $82.5 \pm 2.4$ | $99.11 \pm 0.14$ |

TABLE 6.2. Results of the evaluation criteria for the ten runs of CoLe-CNN, calculated for each case of the test set and then averaged. In bold, we indicate the best results in the column. The row "Average" indicates the total mean of the criteria resulting from the runs, whose STD has been calculated on the average of each run.

computation time. The learning rate has been fixed at 0.0001. The average time consumption for a single computation was 200 minutes. We settled the input images at a fixed dimension of $64 \times 64$ pixels. We maintained all the images with nodule diameter smaller than 32 pixels at the original resolution[6] and interpolated the CTs with a bigger nodule at a pixel density:

$$R^* = R\frac{d_{nod}}{32} \qquad \text{if} \quad d_{nod} > 32,$$

where $R$ is the original pixel density and $d_{nod}$ is the diameter of the nodule. Considering that the pixel length of the elaborated images is $P_L = 0.5 \div 1\,mm$, we encountered a slight loss of resolution only in the nodules having a diameter larger than $20\,mm$ on average. In these few cases, the loss is anyway negligible. We used the Adam optimizer for back propagation, since it ensures stability in non-stationary problems even in the presence of sparse gradients.

## 6.3 Results for Lung Nodule Segmentation

The overall CT slice number used for testing is 61 and it is insufficient to cover all the possible species of nodules in the dataset. In order to verify the accuracy of the method with a larger number of configurations, we trained and tested the network ten times. The results of all ten runs are displayed in Table 6.2. Before each of the runs, we shuffled the dataset and randomly partitioned it in training and testing sets with a ratio of 1/8. We also reported the mean value for intersection over union, sensitivity, precision, F1 score, and $\gamma$-index pass-rate, with the respective errors calculated as the standard deviation on each produced mask.

[6] By original resolution we refer to the resolution of the CT after the interpolation that sets the pixel length on every axis equal to the minimum pixel length of the CT.

Four interesting cases are shown in Figure 6.7. Excluding the simplest situations, we want to demonstrate the good behavior of our method in the most complex cases. Generally, these cases correspond to situations where the nodule is surrounded by filaments or other non-tumor masses, in the extremely noisy CTs[7], in presence of low contrast or when the nodule is not so clearly visible, when the nodule is nestled on the pleural surface, and when the shape of the tumor is unusual or irregular. For all the aforementioned cases, we show some results produced by our network in Figure B.1, B.2, B.3 and B.4 in appendix B.2.

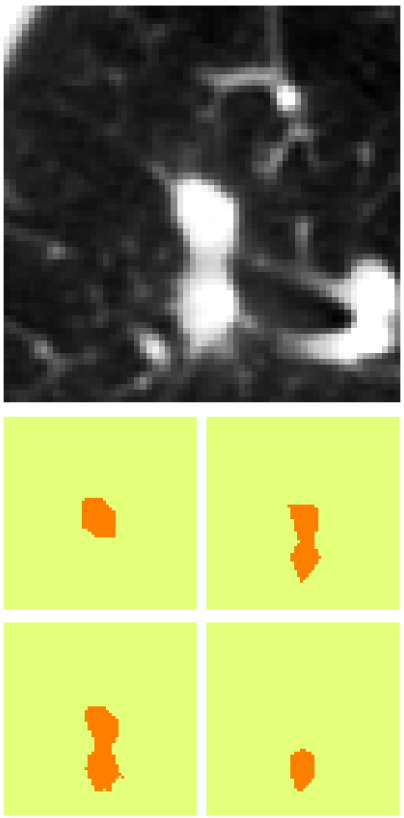


FIGURE 6.4. Example of high expert inter-variability of nodule labeling. In the upper part, the original CT image is shown. Down, we can observe the segmentations made by the four radiologists.

## 6.4 Discussions on Lung Nodule Segmentation

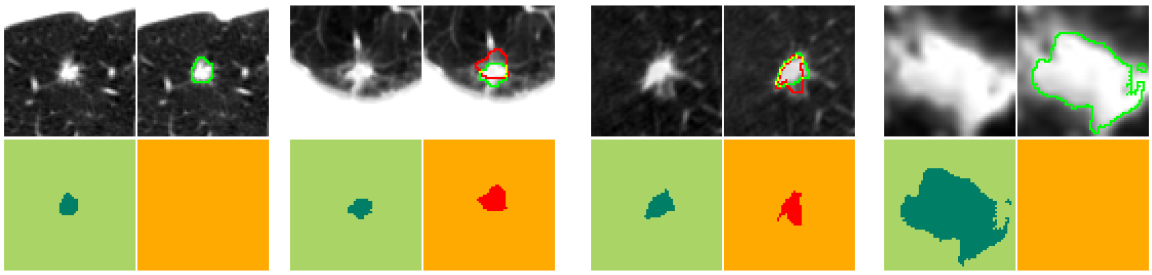
In this chapter, we present an extensive analysis of our results compared with the most recent articles in this field that used the same database.

### 6.4.1 Overall Performance

The first result we can notice is that the deviation from the average is quite significant for all the criteria. The reason for that can be undoubtedly addressed to the combined effect of the broad variety of cases in the dataset and the presence of labels that are incorrect or absent. In Figure 6.6, there are four excellent segmentations with low accuracy, since they are compared to wrong labels. For example, on a test set composed of 61 cases, the absence of the label and a consequent zero in the validation criteria can influence the average of a quantity up to 1.6%. Therefore, it is difficult to discuss the reliability and reproducibility of the results.

In terms of sensitivity and precision, the results show to be unbalanced. In run number 7, an F1 score of 81.8% corresponds to a precision = 90.7% and a sensitivity = 78%. At a first glance, it could appear as an encouraging result but, in practice, it means that we are under-segmenting the nodule with the risk of losing fundamental parts of it. In order to reach this balance, we should fine-tune the parameter $\beta$ in Equation 6.2.2. It may not afflict significantly the values of

[7] The strong noise does not belong only to old CT scanners, but it can be determined by several reasons also in new machinery.



(A)
$\gamma$-index = $N/A$
IoU = 0%
Sensitivity = 0%
Precision = 0%
F1 = 0%

(B)
$\gamma$-index = 98.19%
IoU = 34.84%
Sensitivity = 43.2%
Precision = 64.29%
F1 = 51.67%

(C)
$\gamma$-index = 98.71%
IoU = 55.41%
Sensitivity = 72.5%
Precision = 70.16%
F1 = 71.31%

(D)
$\gamma$-index = $N/A$
IoU = 0%
Sensitivity = 0%
Precision = 0%
F1 = 0%

FIGURE 6.6. Examples of detected and segmented nodules in cases of wrong labeling. The output of our method is in green, the ground truth in red, and their superposition in yellow.

IoU and F1, but it will produce more accurate masks, especially in nodules with peculiar shapes. The $\gamma$-index test shows that, on average, the produced masks differ from the ground-truth only of $\sim 36$ pixels with a 1-pixel tolerance. In this exact case and considering the resolution of the CT scans, we can infer that every result over 99% can be defined as good.

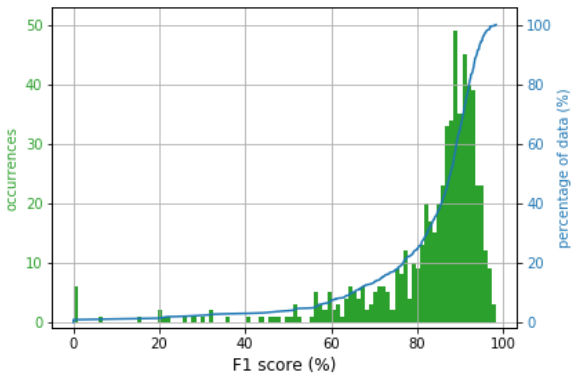
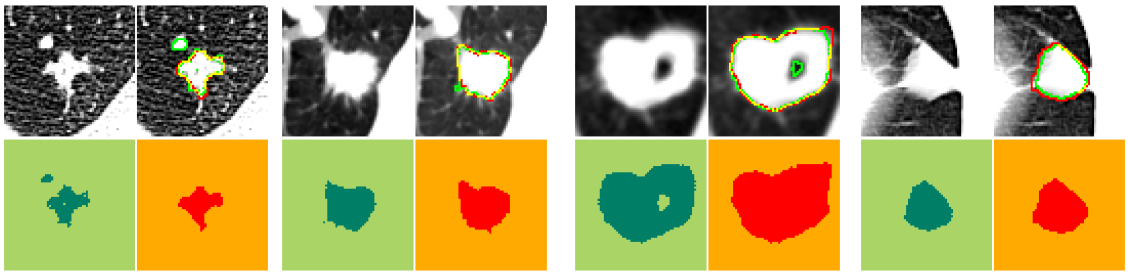



FIGURE 6.5. In green, the histogram of the F1 score results for all the 10 runs. The Cumulative plot of the same data is shown in blue.
Mean = $82.56 \pm 15.62$.

At this point, it is important to observe that, because of the fact that our binary mask is actually the image itself filtered and thresholded, noisy CT scans produce irregular masks, especially in conditions of low contrast. This problem could be fixed with dilation in post-processing.

As last, we show a graph (Figure 6.5) containing the distribution of the F1 score of each test image for all ten runs. The first notable thing is that the histogram has a peak around 90% and that almost the 80% of the cases stand over the 80% of the F1 score. Instead, the mean = $82.56 \pm 15.62$ has been conditioned sensibly from the presence of cases of a very low score.



(A)
$\gamma$-index = 98.95%
IoU = 79.27%
Sensitivity = 89.86%
Precision = 87.05%
F1 = 88.44%

(B)
$\gamma$-index = 99.17%
IoU = 89.61%
Sensitivity = 93.45%
Precision = 95.61%
F1 = 94.52%

(C)
$\gamma$-index = 97.46%
IoU = 90.99%
Sensitivity = 92.6%
Precision = 98.13%
F1 = 95.28%

(D)
$\gamma$-index = 98.51%
IoU = 80.57%
Sensitivity = 80.57%
Precision = 100%
F1 = 89.24%

FIGURE 6.7. Examples of detected and segmented nodules in other interesting cases. The output of our method is in green, the ground truth in red, and their superposition in yellow.

| depth | # runs | IoU | Sens | Prec | F1 |
|---|---|---|---|---|---|
| 3 | 3 | 72.3 | 83.4 | 85.4 | 82.0 |
| 2 (ours) | 10 | 72.6 | 82.3 | 86.6 | 82.5 |

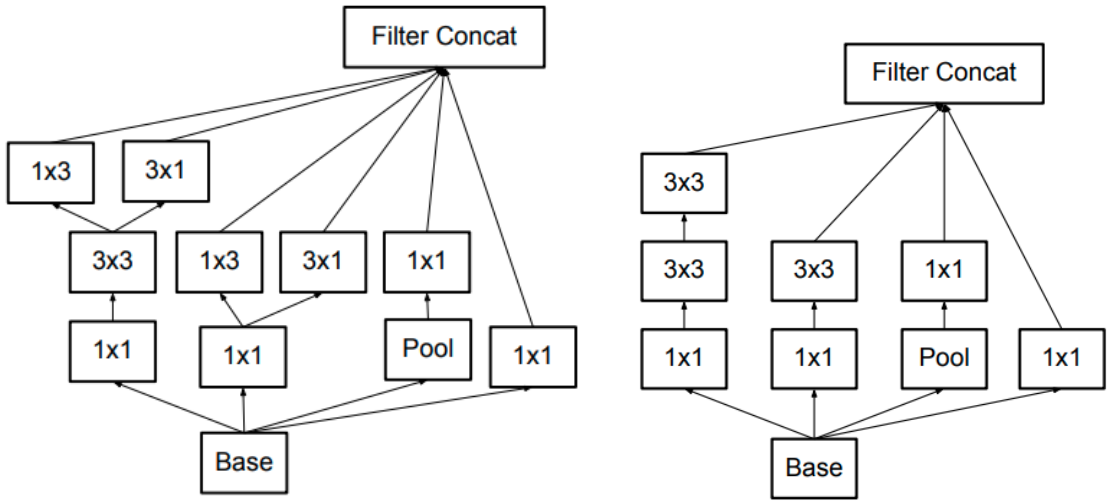
TABLE 6.3. Test of different depths for our NN. All the results have been averaged on the respective number of runs

### 6.4.2 Ablation studies

In order to numerically demonstrate the impact of our introductions on the network, we run three main ablation studies over: the depth[8] of the network, the loss function and the Multiple Convolutional Layers. In all these studies, the hyperparameters and the main structure of the network have not been modified if it was not strictly necessary.

In section 6.2.1, we said that we consider as sufficient a network with depth 2. We pragmatically justified it from the fact that an input image of size $64 \times 64$, at depth 3, is reduced to an $8 \times 8$ matrix (with our fixed pooling kernel size equal to 2) whose amount of transmitted information is negligible for our purpose, even with a very large number of filters. The results in Table 6.3 compare the score of a depth-3-version of our network with our original one of depth 2. As one can notice, the two architectures are almost equivalent in performance but with a significant difference in memory consumption and training time, which are more than doubled in the case of depth 3. Therefore, there is no advantage in using a deeper network.

[8] in our case, depth $k$ means that the input tensor is pooled $k$ times inside the network.




(A) Structure of the MCL from Inception 3

(B) Structure of the MCL from Inception 4 block A

FIGURE 6.8. Representation of the Multiple Convolutional Layers used for this study.

| MCL | # runs | IoU | Sens | Prec | F1 |
|---|---|---|---|---|---|
| no | 4 | 67.9 | 77.6 | 83.8 | 77.8 |
| Inc 3 | 8 | 67.3 | 79.5 | 82.8 | 78.0 |
| Inc 4 | 8 | 69.0 | 83.3 | 81.3 | 79.3 |
| ours | 10 | 72.6 | 82.3 | 86.6 | 82.5 |

TABLE 6.4. Test of different MCLs for our NN. All the results have been averaged on the respective number of runs

| loss func | # runs | IoU | Sens | Prec | F1 |
|---|---|---|---|---|---|
| dice | 2 | 0 | 0 | 0 | 0 |
| KLDiv | 2 | 21.3 | 95.8 | 21.9 | 32.2 |
| L1 | 2 | 46.3 | 48.4 | 83.2 | 56.1 |
| MSE | 3 | 67.0 | 79.7 | 82.8 | 73.3 |
| BCE | 3 | 66.9 | 77.6 | 84.7 | 77.6 |
| ours | 10 | 72.6 | 82.3 | 86.6 | 82.5 |

TABLE 6.5. Test of different loss functions on our NN, compared with ours. All the results have been averaged on the respective number of runs

For the Multiple Convolutional Layers, we took inspiration from the Inception modules of Szegedy *et al.* [108] and [109]. In Table 6.4, we compare the results of our MCL with the ones obtained from implementing the Inception modules in Figure 6.8 and the one obtained without MCL, where a standard $3 \times 3$ convolution has been used as a replacement.

Our method proved to be the best one for this configuration, with a margin of more than three percentage points in IoU and F1. The solution without MCL pays mostly in terms of sensitivity. Clearly, there is also a difference in terms of memory consumption. The weights of our MCL need more GPU memory but this is affordable because of the depth of our network. Instead, a very deep network such as Inception-ResNet ([109]), should look for different solutions in order to be lighter.

| | R1 | R2 | R3 | R4 | Average |
|---|---|---|---|---|---|
| R1 | - | 82.9 | 84.0 | 83.8 | |
| R2 | 82.9 | - | 83.9 | 83.5 | |
| R3 | 84.0 | 83.9 | - | 83.9 | |
| R4 | 83.8 | 83.6 | 83.9 | - | **83.7** |
| ours | 79.5 | 80.5 | 80.3 | 80.2 | **80.1** |

TABLE 6.6. Average F1 score in percentage, calculated for each radiologist with respect to the others. All the uncertainty values for the radiologists go from ±13.0 to ±14.0, while from ±14.8 to ±15.6 for ours.

| | R1 | R2 | R3 | R4 | Average |
|---|---|---|---|---|---|
| R1 | - | 83.3 | 84.9 | 86.6 | |
| R2 | 83.3 | - | 86.6 | 85.1 | |
| R3 | 84.9 | 86.6 | - | 84.6 | |
| R4 | 86.6 | 85.1 | 84.6 | - | **85.2** |
| ours | 87.5 | 87.3 | 87.7 | 86.4 | **87.2** |

TABLE 6.7. Average Precision in percentage calculated for each radiologist. All the uncertainty values for the radiologists range from ±15.1 to ±17.3. From ±17.5 to ±17.8, for ours.

The last of our ablation studies regards the loss function. In order to have a scale of comparison, we tested five of the most commonly used loss functions in segmentation. All these ones are already implemented in PyTorch "nn" module. KLDiv-loss and Dice are producing unstable losses, with a relatively fast divergence of the results. This might also be related to the batch size equal to 1. For memory reasons, we did not deepen the subject. L1-loss tends to under-segmentation, maybe due to the absolute value in the loss function. Mean Square Error and Binary Cross Entropy (BCE) are producing appreciable results but are still less accurate than our loss function, on average. These results, shown in Table 6.5, also demonstrate that our loss function provides a significant improvement in sensitivity without losing precision.

For all the aforesaid reasons, we still consider our configuration as the best among the tested ones.

### 6.4.3 Database Consistency

Before comparing our results with the other published data, we check the internal consistency of the database and observe where our results are positioned with respect to it. We performed a pairwise comparison between the radiologist masks. As we can see in Table 6.6, the disparity between the score of the radiologists within themselves and our network is on average of 3.6 points, in favor of the radiologists. This means that there is a stronger correlation between the manual segmentations than when compared with our method. We can better understand by looking at Table 6.7. The results of the precision are higher for our Neural Network in comparison with the manual masks. This means that our masks are, in general, smaller than the manual ones and are more often contained inside the radiologist masks' borders. This is in perfect matching with the expectations. As a matter of fact, we took as ground-truth the superposition of the single masks at 75%, usually obtaining a segmentation that is often slightly smaller than what the radiologists have suggested and, consequently, a Neural Network in accordance with it. More

data can be seen in the appendix B.1.

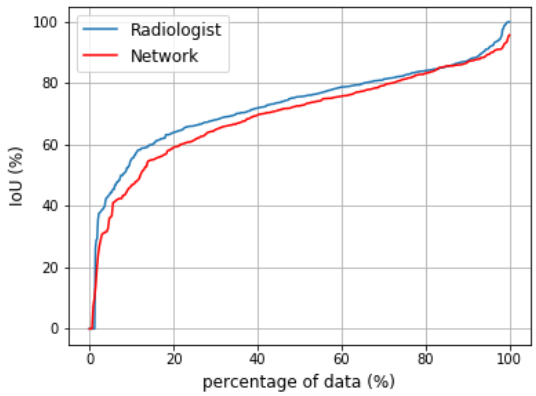



FIGURE 6.9. Cumulative Fraction Plot for IoU between two datasets of the radiologists (R1-R2) and a set of data radiologist-network (R2-NN). With reference at Table 6.10.

The following step was the Kolmogorov-Smirnov test. It helped us to understand if the masks produced from the Neural Network could be distinguished or not from the ground-truth. Thus, we took the masks of the radiologists and the one of the network, we calculated their intersection-over-union and then we run the test on them in pairs. We refer to each of these pairs as a set. As it appears from the data in Table 6.10, there is more accordance between the sets of the radiologists and a more significant discrepancy with the data of the network. According to Kolmogorov-Smirnov's tables (with a significance level $\alpha_{ks} = 0.05$) the critical value is $D^* = 0.061$. Therefore, almost all of the sets are distinguishable by a software, radiologist-radiologist sets included. However, if the distance between the statistics $D$ and $D^*$ is small, it may be impossible to find this pattern by eye. In many cases, it would be impossible to distinguish between the labels and the hand-made segmentations, even for an expert radiologist. In Figure 6.9, we can observe an example of a cumulative fraction plot, where the average difference between the curves is 0.032 in IoU(%). We then can state that we are very near to human performance.

### 6.4.4 Comparison to the State of The Art

In paragraph 6.1.2.1, we exposed the latest fully automated nodule segmentation methods. The purpose of this is to compare them with our results. We have listed the values of Intersection-over-Union and F1 score (whenever possible) both for the aforesaid methods and for our best and average results. Regarding Unet++ [133], we replicated their results for comparison using the code from the GitHub page provided in the article and run it using our images. The summary of the results after the twenty runs is in Table 6.9.

As one can notice, on average, our method gets 2.5% more in IoU, 12.8% more in Precision, and 3.9% more in F1-score (or Dice). This addresses the higher reliability of our method, together with the results of the standard deviation. On average, UNet++ obtains larger results in sensitivity that, without a balanced counterpart in precision, determines a general tendency towards over-segmentation.

From Table 6.8, we can see that our best result overcomes the best results obtained so far in

| Author | year | IoU | F1 |
|---|---|---|---|
| Aresta et al. [2] | 2019 | 55% | - |
| Cao et al. [13] | 2019 | - | 82.74% |
| Hancock et al. [46] | 2019 | 71.85% | - |
| Huang et al. [54] | 2019 | 70.24% | - |
| Jiang et al. [57] | 2019 | - | 68% |
| Keetha et al. [62] | 2020 | - | 82.82% |
| Qian et al. [93] | 2019 | 62.8% | 71.93%[9] |
| Tang et al. [110][10] | 2019 | 69.98% | 81.80% |
| Wang et al. [117] | 2017 | 71.16% | 82.15% |
| Wu et al. [120] | 2018 | 58%[11] | 74.05% |
| Our Average | 2020 | 72.6% | 82.5% |
| Our Best | 2020 | **76.6%** | **86.1%** |

TABLE 6.8. Comparison of our method to the fully automatic nodule segmentation state-of-the-art methods on the LIDC-IDRI database.

literature by 3.3% in F1 and 4.7% in IoU. The method used by Wang *et al.* still stands among the best methods. As in the case of our network, their results have just two pooling layers, but their decoder is simpler and it is constituted by a single layer. This feature, together with the loss function, could really make a difference. The second place in the chart of F1 score is owned by Cao *et al.*. Being the only one using real deep learning, it is heretofore the most expensive in terms of computing time and memory.

| Method | run | IoU | Sens | Prec | F1 |
|---|---|---|---|---|---|
| UNet++ | best | **77.8** | **93.7** | 83.0 | 85.4 |
| ours | best | 76.6 | 90.0 | **85.2** | **86.1** |
| UNet++ | aver | 70.1 | **94.2** | 73.8 | 78.6 |
| ours | aver | **72.6** | 82.3 | **86.6** | **82.5** |
| UNet++ | std | 4.22 | 1.76 | 4.49 | 3.97 |
| ours | std | 2.7 | 3.6 | 3.0 | 2.4 |

TABLE 6.9. Comparison between the results of Unet++ [133] and our method, both applied to the LIDC-IDRI dataset. All statistics in percentage. The result published by Zhou *et al.* [133] for UNET++ was IoU = 77.0%, which is in line with the best results that we obtained in our experiments with this network.

| | R1-R2 | R1-R3 | R1-R4 | R2-R3 | R2-R4 | R3-R4 | R1-NN | R2-NN | R3-NN | R4-NN |
|---|---|---|---|---|---|---|---|---|---|---|
| R1-R2 | - | 0.077 | 0.067 | 0.085 | 0.046 | 0.083 | 0.13 | 0.1 | 0.096 | 0.1 |
| R1-R3 | 0.077 | - | 0.073 | 0.032 | 0.079 | 0.067 | 0.19 | 0.15 | 0.16 | 0.17 |
| R1-R4 | 0.067 | 0.073 | - | 0.071 | 0.036 | 0.038 | 0.19 | 0.16 | 0.15 | 0.13 |
| R2-R3 | 0.085 | 0.032 | 0.071 | - | 0.075 | 0.073 | 0.19 | 0.14 | 0.15 | 0.17 |
| R2-R4 | 0.046 | 0.079 | 0.036 | 0.075 | - | 0.042 | 0.16 | 0.13 | 0.13 | 0.11 |
| R3-R4 | 0.083 | 0.067 | 0.038 | 0.073 | 0.042 | - | 0.18 | 0.16 | 0.16 | 0.13 |
| R1-NN | 0.13 | 0.19 | 0.19 | 0.19 | 0.16 | 0.18 | - | 0.079 | 0.059 | 0.076 |
| R2-NN | 0.1 | 0.15 | 0.16 | 0.14 | 0.13 | 0.16 | 0.079 | - | 0.041 | 0.068 |
| R3-NN | 0.096 | 0.16 | 0.15 | 0.15 | 0.13 | 0.16 | 0.059 | 0.041 | - | 0.047 |
| R4-NN | 0.1 | 0.17 | 0.13 | 0.17 | 0.11 | 0.13 | 0.076 | 0.068 | 0.047 | - |

TABLE 6.10. Statistics $D$ of the Kolmogorov-Smirnov test on the relative intersection over union of the four radiologists and the Neural Network. With a significance level $\alpha_{ks} = 0.05$, the critical value is $D^* = 0.061$.

## 6.5 Conclusions on Lung Nodule Segmentation

In this study, we proposed a new method for pulmonary nodule segmentation that introduces several novel features, from the network architecture to the loss function. The main structure of the network ensures all the properties of the UNet architecture, while the Multi Convolutional Layers give a more accurate pattern recognition. The new solutions adopted in order to reduce the resolution loss also increase the details on the border of the nodule. Compared to the state of the art, this method proved to be the most accurate. Indeed, our maximum average F1 score and IoU are 3.3% and 4.7%, respectively, larger than the best results obtained so far in the literature. The Kolmogorov-Smirnov test and the database consistency show that this method is very near to human precision and that, in most of the cases, it could be impossible to discern by eye between the mask made by our software and a hand-made one.

CHAPTER 7

*You **must** stay at home.*

BORIS JOHNSON

# COVID-19 LESION SEGMENTATION

## 7.1 Introduction to COVID-19 Lesion Segmentation

At the end of 2019, we observed the first signs of the worldwide spread of the Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2), which continued throughout 2020 and has been protracted to 2021 with more than 170 million confirmed cases and more than 3.5 million deaths[1] (updated on May 28th 2021). Initially, the World Health Organization (WHO) declared the Coronavirus Disease 2019 (COVID-19) as a public health emergency and recognized it as a pandemic only on March 11th, 2020.

The current most widespread technique for COVID-19 identification is the Reverse Transcription-Polymerase Chain Reaction (RT-PCR) test, which detects the presence of the virus through the throat (or sputum) in swab samples. The PCR test does not provide exact information about the severity of the disease, e.g. the spread of the pulmonary lesion caused by COVID-19 [65]. Moreover, once the virus is eliminated by the immune system, the PCR will likely be negative even if the patient is still affected by pneumonia induced by the original COVID-19 infection. It has been shown that this condition can last for weeks [34] [69]. At this point, it has become necessary to develop robust tools based on medical imaging techniques that can provide clinicians with the clinical information they need for properly assessing the progression of the COVID-19 disease.

Two-dimensional (2D) chest X-ray and three-dimensional (3D) thoracic computed tomography (CT) demonstrated to be able to provide a clear picture of the presence, the spread, and the severity of COVID-19 disease [19]. For example, Zheng *et al.,* [125] accurately described how to detect the most common manifestations of this disease, as depicted in Figure 7.1. Ai *et al.,* [1]

[1] Data from the World Health Organization.

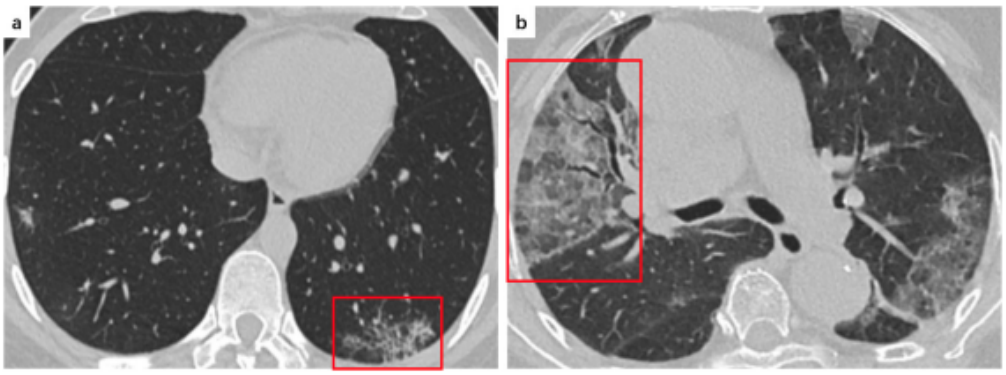



FIGURE 7.1. a) A 34-year-old female COVID-19 patient presenting a fever with a dry cough for 2 days. CT scan shows a slight reticular pattern in the left lower lobe and subpleural area (red frame). b) An 81-year-old female COVID-19 patient presenting fever with cough for 7 days. CT scan shows a reticular pattern superimposed on the background of GGO, resembling the sign of crazy paving stones in the right middle lobe (red frame). Extracted from Zheng *et al.,* [125].

observed that, on more than 1,000 suspected cases, only 59% of the patients resulted positive to the RT-PCR test, whereas 88% of them have shown visible COVID-19 traces through the use of CT images. This study has found that the sensitivity of the diagnosis via CT is 97% on the cases predicted with the swab test and that only 3.5% of the patients showed progression on follow-up chest CT scans after RT-PCR test results turned negative. This demonstrates the importance of CT scans, not only in COVID-19 detection but also in the follow-up of the disease and for assessing the damages after the treatment.
Similarly, Fang *et al.,* [32] stated that 98% of the 51 patients studied showed compatible symptoms with COVID-19 during the CT scan reading, but only 36 of them (71%) resulted positive to the RT-PCR test.

### 7.1.1 Aim of COVID-19 Lesion Segmentation

Lesion segmentation is not useful to diagnose COVID-19, but it is fundamental for assessing the current status of the illness, its severity, and for future treatment planning. In fact, the area covered by the lesion approximately corresponds to the area where the pulmonary alveoli are not working as normal (Liang et al.,[69]). Despite the increase in COVID-19's detection accuracy through the use of CT images, the reading time necessary to interpret 3D CT volumes and to extract the morphological properties of the lesion can greatly increase the workload of radiologists. However, the use of Artificial Intelligence (AI) tools can help to sensibly reduce the interpretation time, as attested by Hosny *et al.,* [50]. Indeed, manual segmentation represents an extremely time-consuming task. Based on a study involving 10 radiologists, Ma *et al.,* [79] found that, on average, one radiologist needs about $400 \pm 45$ minutes to accurately delineate the lesion in a CT scan with 250 slices. However, deep learning strategies can reduce this time to less than one minute.

The work presented here proposes an automated method for COVID-19 detection using chest CT images, together with the segmentation of the Ground-Glass-Opacities (GGO) (Figure 7.1) and other solidifications/fibrosis present inside the lungs. Thus, a unique Convolutional Neural Network (CNN)-based workflow is built, where the following distinct steps are included:

- Lung segmentation,
- COVID-19 detection, and
- COVID-19 lesion segmentation.

The lungs are first segmented from the input CT image to reduce the searching area. Afterward, the detection algorithm is used to analyze the lungs' area in order to detect the presence of COVID-19. In the case of a positive finding, the CT image is processed by the last network (COVID-19 lesion segmentation) to identify the areas affected by the disease.

This chapter is arranged as follows: Section 7.1.2 shows the current state-of-the-art concerning the most relevant methods of COVID-19 detection and lesion segmentation using CT scans. In the following Section (7.2), the methods of COVID-19 detection and segmentation developed in this work, including the model architecture (7.2.1), are presented. Then, Section 7.2.2 and 7.2.4 describe the experimental methods and the dataset used. Next, we present and discuss the results in Section 7.3 and 7.4, respectively. In particular, we present the overall performance of the method in Section 7.4.1 and the comparison of the proposed method with the state-of-the-art in Section 7.4.2. Conclusions and future work are finally presented in Section 7.5 and **??**, respectively.

### 7.1.2 Relevant Work on COVID-19 Detection and Segmentation

COVID-19 has hit the world with such an unbridled force that completely transformed our lives and directed most of the scientific research to counteract its effect. Therefore, the literature on this topic is rapidly increasing, especially in the field of AI, where automated tools can support healthcare professionals in their diagnosis.

We hereby investigated and included the most relevant works about COVID-19 detection and lesion segmentation algorithms using AI using chest CT images. First, we briefly introduce the work of Harrison *et al.,* [47], who tested the expertise of 7 radiologists in discerning between cases of COVID-19 and pneumonia. They did a blinded test using more than 400 CT images, in which the ground-truth was set by RT-PCR test results. The authors gave also an accurate list of differences between COVID-19 and non-COVID-19 viral pneumonia appearance in chest CTs. The results are presented in Section 7.3.

#### 7.1.2.1 COVID-19 detection

With the term *COVID-19 classifier*, we refer to one software that classifies the patients in two main classes: positive or negative to COVID-19, depending on the presence or the absence of the disease. Shi *et al.,* [103] carried out one of the first review articles which suggests the use of neural networks for COVID-19 classification purposes. Among all analyzed works, we highlight the work of Zheng *et al.,* [130] that developed a two-stage neural network. The first stage for lung segmentation, using UNet architecture [99], and the second one for COVID-19 classification, where only the lung area was considered as the input to the system. For this final step, they provided a mixture between residual blocks and standard convolutions with adaptive max-pooling layers. They trained and tested their algorithm by using a private dataset with remarkable results in terms of accuracy and sensitivity, but low specificity.

Wang *et al.,* [116] adopted a modified Inception network [109] [108] to classify the patients into COVID-19 or pneumonia classes. They also made use of a private CT image dataset of nearly 100 patients, in order to fine-tune and then test the network. They compared the predictions of their neural network with those provided by two radiologists, on a total of 745 images. Moreover, they reported a study according to which 75% of patients with negative RT-PCR results were positive to COVID-19 from findings in their CT images.

Wu *et al.,* [119] provided an explicable neural network, based on VGG-16 [105] that classifies the patients into COVID-19 and non-COVID-19, and returns a heat map to highlight the areas, where the lesion is present. With the introduction of a large dataset (810 images) and under deep supervision, they registered a maximum of 96% for sensitivity, but specificity was below 94% for all the cases. They additionally provided a COVID-19 segmentation algorithm that will be presented in Section 7.1.2.2. Hu *et al.,* [52] used a 5-layer convolutional neural network for infection detection and classification. The encoded information was extracted from the last layer of the three last levels of depth, then this was concatenated and used to predict the patient status. With the aid of a well-constructed data augmentation strategy, they presented their results for classification between COVID-19, no-COVID-19, and pneumonia cases. They also included a weight that is dependent on the class frequency of occurrence, as well as on the loss function.

#### 7.1.2.2 COVID-19 segmentation

In this work, COVID-19 segmentation refers to the delineation of the lesion induced by the presence of coronavirus into the lungs. As described by Kong and Agarwalhey [63], the lesion appears in two different shapes: (1) as a GGO and (2) as a solidification of the tissue. Although they are classified independently one from the other, we do not distinguish between them in this work because they both are COVID-19 manifestations.

Compared to our proposed pipeline, the most similar work we found in the literature is from Fan *et al.,* [31]. They used the same open-access image databases and, in the same way, they employed a lung segmentation algorithm as a first step. The Inf-Net that they proposed is a

combination of Reverse Attention modules for edge learning and parallel decoders with deep supervision. In order to provide a more comprehensive picture of their segmentation performance, they also compared their results with the most widespread versions of UNet.

Wu *et al.,* [119] used a large dataset of 810 CT images, which is partially available online. Their network for predicting the lesion masks for COVID-19 consists of a VGG-16 encoder, similar to their classification network, and a 5-levels-of-depth decoder.

The COVID-19 Pneumonia lesion (COPLE-Net) segmentation algorithm, designed by Wang *et al.,* [118], was also inspired by UNet, but with the addition of extra residual connection and a reduced number of filters carried from the encoder. Their method was further enforced with noise-robust features and loss function. They trained and tested their method on a large private dataset.

Chen *et al.,* [16] provided a residual attention UNet, where the convolutions have been substituted with *ResNeXt Blocks*. They used a relatively small dataset of 110 CT images, but with intensive work on data augmentation. A similar work was conducted by Zhou *et al.,* [132], using the same dataset and the same network structure.

Yan *et al.,* [123] proposed a Feature Variation (FV) block that enhanced the capability of feature representation adapting to diverse cases. They also applied Progressive Atrous Spatial Pyramid Pooling for handling various infection areas with diverse appearances and shapes. Their dataset included more than 800 CT scans from five different hospitals in China, whose manual annotations and segmentation (i.e. ground truth) were performed by six expert radiologists.

Elharrouss *et al.,* [30] used two encoders and one decoder for lung and COVID-19 lesion segmentation. The inputs of the lung segmentation model correspond to the CT slice image and its texture. In addition, the COVID-19 lesion segmentation model required the CT image and the segmented lungs. They used a subset of a publicly available dataset of 100 CT images.

The experimental part of Ma *et al.,* [79] is likewise similar to the one proposed here. They first trained their network for lung segmentation (with an average of 86.91% in Dice-score) and then, after filtering out the part of the CT external to the lungs, they performed COVID-19 lesion segmentation. This network represents a revisited version of UNet with minor changes, trained and tested five times on a publicly available dataset that has been prepared by the authors specifically for this task.

Yu *et al.,* [126] proposed a multi-class COVID-19 segmentation network in the shape of a classic encoder-decoder structure, with a pyramid attention mechanism and a loss function based on wavelet decomposition. The pyramid attention module combines a pyramid multi-scaling and channel attention mechanism to highlight salient features at each stage, in addition to a wavelet edge loss function, which uses wavelet decomposition to extract multi-directional edge information of the lesion area to improve the accuracy of the segmentation. They use two datasets: one public, used also in our work, and one private. We report in Section 7.3 the statistics only for the public dataset.

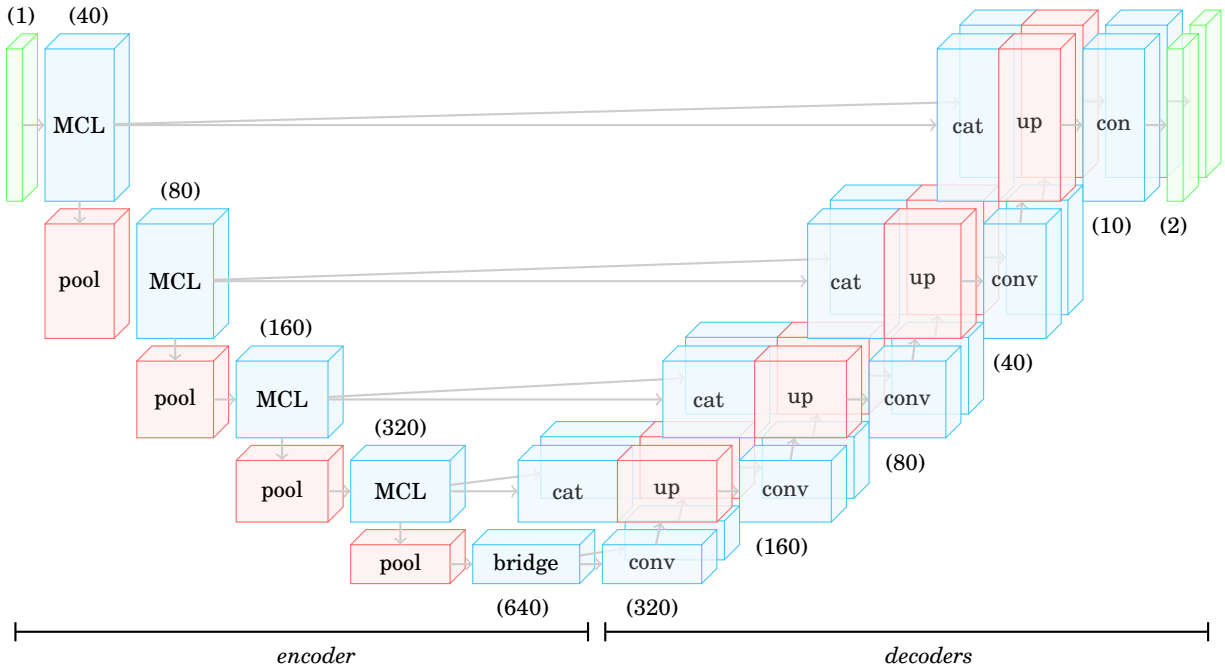



FIGURE 7.2. Structure of our CoLe-CNN+, with one encoder and two decoders. The name on each sample corresponds to the operation applied. The last step is always a convolution. The number of filters is provided between brackets. Input and output, convolutional and re-scaling layers are indicated in green, blue, and red, respectively. The abbreviations *conv*, *cat* and *up* correspond to convolution, concatenation, and up-sampling, respectively.

Despite the publication of similar AI strategies in the literature to detect and segment COVID-19 from CT images, we observed a general lack of open-access datasets, which would facilitate a fair comparison by members of the community. To train and validate the methods proposed in this work in a robust and reproducible way, we created a multi-center and multi-vendor CT image dataset from publicly available datasets, as described below.

## 7.2 Methods for COVID-19 Pipeline

Before explaining the architecture used in this work, we want to underline that all the input images of the network are CT axial images normalized in the interval $[0,1]$ (where a lower value corresponds to a medium with a lower electron density) and re-sampled at a resolution of $256 \times 256$ pixels.

### 7.2.1 Model Architecture

In Figure 7.2, We present the model architecture used for lung and COVID-19 lesion segmentation. It corresponds to a *U-shaped* network with 4 levels of depth. At each level of the encoder, there

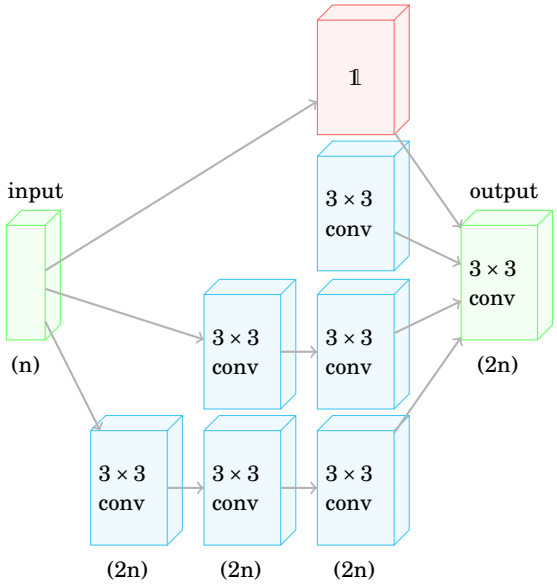



FIGURE 7.3. Structure of our MCL, with four branches. The name on each sample corresponds to the operation applied in order to obtain it. The symbol $\mathbb{1}$ represents the Identity function repeated the corresponding number of times. The relative number of filters is provided between brackets. With $n$ we indicate the input number of filters.

is a Multiple Convolutional Layer (MCL) and a pooling layer. The MCL, shown in Figure 7.3, has four parallel branches. In three of these, the input tensor is convoluted from one to three times with a series of $3 \times 3$ convolutions. In the fourth branch, it is simply copied with an identity function (marked as a red cube). The four results are then concatenated and convoluted one more time. The number of filters is doubled at each MCL, except for the first layer where the number of filters goes from 1 to 40. After this block, the tensor is pooled with a $2 \times 2$ kernel and then given to the following level. The last stage is a $3 \times 3$ convolution, commonly called *bridge*. Afterward, the tensor passes to the decoders. These decoders are two almost identical parallel branches providing two different outputs. In each decoder, the input is up-sampled and concatenated with the output from the MCL of the correspondent level of depth, then it is convoluted again. This step is repeated four times in order to restore the original shape of the image. The only difference between the two decoders lies in the size of the kernels of the convolutions, which is $3 \times 3$ in one branch and $5 \times 5$ in the other one.

The architecture of the segmentation algorithms is inspired by the CNN used in Pezzano *et al.,* [89] and provides some major differences developed specifically for the purposes of COVID-19 lesion segmentation. The key differences from the architecture used in our previous work are: (i) the use of a loss function with a parameter for maximizing sensitivity, (ii) a new architecture of the MCL, (iii) a new mask calculation formula (explained in the final part of Section 7.2.1), (iv) an additional post-processing procedure to reduce false positives and increase specificity, (v)

two additional levels of depth of the network, which then go from two to four levels, and (vi) an extensive validation with an accurate selection of hyper-parameters derived specifically for the purposes of COVID-19 lesion segmentation, i.e. learning rate, batch size, thresholds, etc.

In our architecture, each decoder $j$ produces two masks ($M_{j,1}$ and $M_{j,2}$) that are then used to construct the COVID-19 lesion segmentation mask $M_{cov}$ with the following formula:

$$\begin{aligned} M_{cov} =& I \cdot M_{1,1} + M_{2,1} \\ & - I \cdot M_{1,2} - M_{2,2} \\ =& I\ (M_{1,1} - M_{1,2}) + M_{2,1} - M_{2,2}, \end{aligned} \tag{7.1}$$

where $I$ is the input CT image, and the operators $\cdot$, $-$ and $+$ are the element-wise product, subtraction, and addition, respectively. The mask is then converted into a binary image using a threshold set at 0.1. In the case of lung segmentation, equation (7.1) is slightly different, because we want to segment an area whose values are near zero intensity, since the volume inside the lung, consisting mainly of air, is characterized by a very low density. Thus, we calculate the mask starting from 1 (a tensor of unitary values only, with the same shape as the input) minus the input image, as shown below:

$$\begin{aligned} M_{cov} =& (1 - I) M_{1,1} + M_{2,1} \\ & - (1 - I) M_{1,2} - M_{2,2} \\ =& I\ (M_{1,2} - M_{1,1}) + M_{2,1} - M_{2,2} \\ & + M_{1,1} - M_{1,2} \end{aligned}$$

The structure of the decoder changes for COVID-19 classification purposes (Figure 7.4), due to the need of using Fully Connected Layers (FCLs). This decoder takes the result of the bridge operation and convolves it three times. The outcoming tensor is then vectorized (or flattened) and given to the FCLs. The information is processed for three more layers until the probability of both classes is predicted, with a *softmax* activation layer as the last operation.

### 7.2.2 Implementation Details

The lung and COVID-19 segmentation networks have been coupled with a composite loss ($L$) that minimizes the Mean Square Error (MSE) and the sensitivity ($S$):

$$L = \sum_i \left[ MSE(z_i, \hat{z_i}) + 0.5\big(1 - S(z_i, \hat{z_i})\big) \right], \tag{7.2}$$

where $i$ is the index that runs over all the pixels of the images, $z$ and $\hat{z}$ are the elements of the predicted mask and the ground truth.

The classification network uses the Cross Entropy (CE) loss, which is defined as follows:

$$CE(x, class) = -x[class] + \log\left( \sum_j e^{x[j]} \right),$$

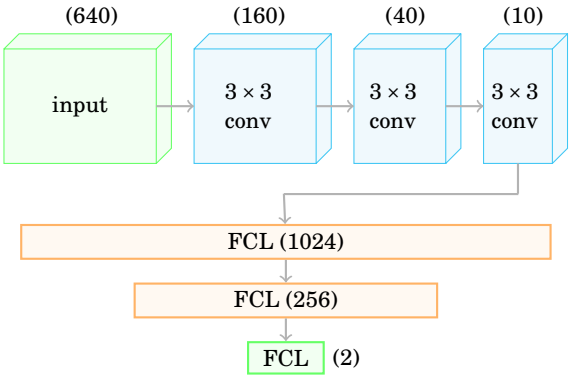



FIGURE 7.4. Structure of the decoder for COVID-19 detection. The name on each sample corresponds to the operation applied in order to obtain that sample. In parentheses, the relative number of filters appears.

where the classes are two, namely COVID-19 and no-COVID-19 .

At the beginning of each training, the dataset has been randomly divided into training and test sets with a ratio of $1 \div 8$. Due to the limited number of data, we operated a data augmentation strategy. At the beginning of each epoch, all the CT slices of the training test have been randomly rotated within the interval $[-180°, 180°]$ and randomly zoomed with a magnification included in the range $[\times 0.95, \times 1.05]$. The eventual padding has been made with black pixels (equivalent to zero). As a next step, we operated another transformation using the *torchvision.transforms.ColorJitter* function. This function randomly changes the brightness, saturation and contrast of a factor of $\pm 0.2$. This helps the network to learn how to work with multi-vendor, multi-center input data with different exposure times, dose, and reconstruction methods.

In our experiment, the total number of learnable weights for COVID-19 detection is $15,427,042$ for a total of $\sim 62MB$ of memory occupation. For segmentation, the number of weights is $5,504,074$ for $\sim 22MB$. We trained our CNNs during 100 epochs in the classification case and 60 epochs in the segmentation case. In order to support our expensive calculations, we used a machine equipped with an Intel Xeon E7-4830 v3 (i7) processor with 64GB of RAM. The GPUs are two Nvidia GeForce GTX 1080 Ti, each one with 12GB of CUDA memory. All the libraries employed were implemented in Python 3.7 with the usage of the PyTorch module. We decided to use a batch size of 2. This allowed us to gain better results at the expense of the computation time. The learning rate has been fixed at 0.00001 for classification and 0.0005 for both segmentation neural networks. The average time consumption for a single segmentation training was 180 minutes, and 30 minutes for classification. The time spent for testing on a single image (once the model has been loaded[2]) is less than a second for all the networks. We used Adam optimizer for

[2]Loading the model is an operation that can take up to one minute on our hardware and needs to be done only once for every test set.

back-propagation, since it ensures stability in non-stationary problems even in the presence of sparse gradients.

### 7.2.3 Post-Processing

In order to increase the accuracy of our methods, the resulting segmentation masks were further processed. This step allows us to reduce the image noise, which is present in the form of sparse points with no relevance to the actual segmentation. Using the *morphological transformations* module of OpenCV (v. 4.3.0), we apply a first softer operation of opening (first erosion and then dilation) with the $3 \times 3$ kernel $K_1$ (Equation 7.3). Then, in order to obtain smoother contours and to avoid holes within the segmented area, we operated a stiffer transformation of closing, using the $K_2$ kernel (Equation 7.3):

$$K_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad K_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{7.3}$$

### 7.2.4 Image Dataset

For this study, we created a multi-vendor, multi-centre COVID-19 CT image dataset after combining data from several public databases. Due to the novelty of the research topic, open-access COVID-19 CT image repositories are difficult to find. We obtained access to a total of 79 COVID-19 CT volumes and 110 CT slices that have sufficient annotation for our purposes. Among these, 20 volumes come from Cohen *et al.,* [17] and all of them have COVID-19 lesion and lung segmentation annotations. From *MosMedData*[3] we downloaded 1110 CT images but only 50 of them were useful for our purpose, having COVID-19 lesion segmentation. The remaining 9 CT volumes and 110 slices were collected from *SIRMI*[4] and include lung and lesion masks. In order to have a pool of CT scans of patients not suffering from COVID-19 or pneumonia (healthy cohort), we included in our dataset 884 CT images from *LIDC-IDRI*[5] dataset, described in Armato *et al.,* [3]. These CTs have also complete lung masks, thus we employed them for both classification and lung segmentation.

Since our model is for 2D image processing, it allows us to take different slices from the same CT volume, but with the constraint that 2D slices belonging to the same CT, i.e. the same patient, cannot be found both in the training and the test set. The datasets have been split at patient level for cross-validation. In all the cases, the images extracted from one CT, i.e. from one patient, can belong strictly to the train set **or** to the test set only. There are no repeated CTs in our datasets, nor the intersection of patient data in both training and test datasets. In this case, we automatically selected 558 slices from 79 CT volumes, using the criteria that the area of the annotated COVID-19 lesion must be larger than 20 pixels at a resolution of $256 \times 256$, and the

[3] Moscow Medical Data accessed on August 21st 2020
[4] Società Italiana di Radiologia Medica e Interventistica accessed on August 21st 2020
[5] LIDC-IDRI dataset accessed on August 21st 2020

distance between the centers of the slices must be larger than $30mm$. In order to get a larger set of images for lung segmentation, we extracted three slices from each CT of the LIDC-IDRI dataset. Therefore, the total number of 2D CT images used in this work is: 663 with COVID-19 and 2,652 without COVID-19 . All of the aforesaid slices have been taken on the axial plane.

It has recently become a standard procedure in medical image segmentation to compare the results with those obtained with UNet. Indeed, UNet is a reliable network that often represents the skeleton of more modern architectures. Due to the absence of unified COVID-19 -CT databases as well, an objective method to make a comparison between different works would be to use UNet[6] as a benchmark.

### 7.2.5 Evaluation Statistics

After the post-processing stage, a binary image[7] is created using a threshold set at 0.5 for lung segmentation and at 0.1 for COVID-19 lesion segmentation. Such binary image is then compared to the ground truth (G) in a pixel-by-pixel manner. The evaluation criterion most commonly used in this type of problem is the *sensitivity* ($Sens$). High sensitivity is desired, although this could lead to over-segmentation. For this reason, this must be balanced with *precision* ($Prec$). Their *reduced mass* is proportional to the *F1* (or Dice Similarity Coefficient - *DSC*). In addition to these evaluation metrics, we calculated the accuracy ($Acc$) and the specificity ($Spec$) of the prediction. In terms of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) predictions[8], we used the following statistics:

$$
\begin{aligned}
Acc &= \frac{TP+TN}{TP+TN+FP+FN} \\
Spec &= \frac{TN}{TN+FP} \qquad Prec = \frac{TP}{TP+FP} \\
Sens &= \frac{TP}{TP+FN} \qquad F1 = 2\frac{Sens \cdot Prec}{Sens+Prec}.
\end{aligned}
$$

Another widespread evaluation criterion is the *Intersection over Union (IoU)* described by:

$$
IoU = \frac{G_+ \cap M_+}{G_+ \cup M_+},
$$

where $G$ is the ground truth, $M$ is the mask and $_+$ is the index indicating only the pixels where the values of the tensors are equal to 1. Although IoU does not provide more information than F1, it is extensively used since it is also invariant to changes in the entries: $IoU(G,M) = IoU(M,G)$.

[6] UNet [99] is the most common architecture for image segmentation in the medical field, at the moment
[7] The values of our binary images are 0 and 1, where 0 is a negative and 1 is a positive prediction.
[8] The prediction for segmentation is pixel-wise calculated

| Cases | IoU | Acc | Sens | Prec | F1 |
|---|---|---|---|---|---|
| 392 | 94.7 | 99.4 | 98.9 | 95.4 | 96.9 |

TABLE 7.1. Average results of the algorithm used for lung segmentation, over the number of cases. Statistics are given in percentages.

| run | IoU | Acc | Spec | Sens | Prec | F1 |
|---|---|---|---|---|---|---|
| 1 | 64.0 | 98.9 | 99.2 | 73.7 | **85.5** | 76.7 |
| 2 | 58.1 | 99.0 | 99.6 | 77.8 | 72.4 | 71.5 |
| 3 | 59.2 | 99.2 | 99.6 | 70.4 | 79.8 | 72.1 |
| 4 | 59.4 | 98.8 | 99.3 | 69.4 | 83.6 | 73.0 |
| 5 | 60.9 | 98.7 | 99.4 | 77.5 | 74.4 | 73.4 |
| 6 | 54.0 | 98.9 | 99.3 | 68.9 | 75.2 | 67.4 |
| 7 | 64.7 | **99.6** | **99.8** | 79.4 | 78.2 | 76.7 |
| 8 | 62.3 | 98.4 | 99.5 | **82.8** | 72.9 | 73.3 |
| 9 | **65.2** | 99.1 | 99.6 | 80.4 | 80.5 | **78.3** |
| 10 | 65.0 | 98.8 | 99.6 | 88.1 | 72.4 | 77.1 |
| aver | 61.3 | 98.9 | 99.5 | 76.8 | 77.5 | 73.9 |
| std | 3.47 | 0.30 | 0.17 | 5.94 | 4.52 | 3.13 |

TABLE 7.2. Result of the 10 runs of our COVID-19 lesion segmentation algorithm. Statistics are given in percentages.

## 7.3 Results for COVID-19 Pipeline

### 7.3.1 Lung Segmentation

Since COVID-19 mainly affects the inside part of the lungs, the pleura and all the background have been filtered out from the CT images before training the classification and the segmentation networks. In order to provide a fully automated method, the lung masks have been calculated with a neural network identical to the one used for lesion segmentation. The lung masks have been generated and our results are shown in Table 7.1.
The training was performed with a special focus on the sensitivity parameter. Indeed, the sensitivity was kept as high as possible in order to minimize the risk of filtering out a relevant area inside the lungs.

### 7.3.2 COVID-19 lesion Segmentation

Table 7.2 shows the results of ten runs of our COVID-19 lesion segmentation network, measured with the evaluation metrics presented in Section 7.2.5. For each run, the dataset was randomly divided into training and test sets with a proportion of $1 \div 8$. In addition, the slices belonging

| run | Acc | Spec | Sens | Prec | PPV | NPV |
|---|---|---|---|---|---|---|
| 1 | 98.5 | 100 | 86.7 | 100 | 100 | 98.3 |
| 2 | 95.5 | 100 | 73.9 | 100 | 100 | 94.8 |
| 3 | 97.0 | 100 | 78.9 | 100 | 100 | 96.6 |
| 4 | **98.5** | 100 | **87.5** | 100 | 100 | **98.3** |
| 5 | 96.2 | 100 | 64.3 | 100 | 100 | 95.9 |
| 6 | 96.2 | 100 | 76.2 | 100 | 100 | 95.7 |
| 7 | 98.5 | 100 | 84.6 | 100 | 100 | 98.3 |
| 8 | 97.0 | 100 | 71.4 | 100 | 100 | 96.7 |
| 9 | 97.0 | 100 | 77.8 | 100 | 100 | 96.6 |
| 10 | 97.0 | 100 | 78.9 | 100 | 100 | 96.6 |
| aver | 97.1 | 100 | 78.0 | 100 | 100 | 96.8 |
| std | 1.00 | 0 | 6.80 | 0 | 0 | 1.14 |

TABLE 7.3. Results of the 10 runs of our COVID-19 detection algorithm. Statistics are given in percentages.

to the same CT were not used in both training and test set. Then, for each epoch, we randomly operated data augmentation as explained in Section 7.2.2. Figures 7.5 and 7.6 present sample results for lesion segmentation.

### 7.3.3 COVID-19 detection

The results after ten runs of training of COVID-19 detection are depicted in Table 7.3. In this case, the ratio between the number of CT slices with and without COVID-19 is $1 \div 6$.

## 7.4 Discussions on COVID-19 Pipeline

In this section, we present an extensive analysis of our results compared with the most recent related articles in the literature.

### 7.4.1 Overall Performance

As previously described, the input of our CoLe-CNN+ was filtered using a prior lung segmentation step. Through the use of a sensitivity parameter in the loss function, in Equation 7.2, we obtained a sensitivity of almost 99%, on average. This result, coupled with high accuracy, shows that the loss of information within the inside part of the lungs is negligible in the majority of the cases. This lung segmentation step has two main advantages for the COVID-19 lesion segmentation: (i) an improvement in accuracy and specificity (Table 7.2), because false positives in the background are minimized; and (ii) a reduction of the probability to have under-segmentation of the lesions,

| Paper | CTs | Acc | Sens | Prec | Spec | PPV | NPV |
|---|---|---|---|---|---|---|---|
| Harrison *et al.,* * [47] | 424 | 74.3 | 79.3 | - | 68.7 | 78.7 | 76.3 |
| | 58 | 82.1 | 80.4 | - | 83.7 | 89.8 | 78.6 |
| Wang *et al.,* [116] | 453 | 73.1 | 67.1 | - | 76.4 | 61.0 | 81.0 |
| Zheng *et al.,* [130] | 630 | 90.8 | - | - | - | 86.7 | 96.4 |
| Wu *et al.,* [119] | 810 | - | **96.0** | - | 91.5 | - | - |
| Hu *et al.,* [52] | 450 | 87.4 | 88.5 | 87.5 | 87.1 | - | - |
| our average | 189 | 97.1 | 78.0 | 100 | 100 | 100 | 96.8 |
| our best | 189 | **98.5** | 87.5 | **100** | **100** | **100** | **98.3** |

TABLE 7.4. Average results in classification from radiologists and other major articles in this field. Results are given in percentages. * In Harrison *et al.,* [47], the classification has been made between COVID-19 and pneumonia cases.

e.g. if an incorrect lung mask considers a COVID-19 lesion as part of the pleura and filter it out, that lesion would not be detected. The values of sensitivity and precision are balanced, on average. This means that our algorithm has generally found the optimum equilibrium for a reliable prediction.

Table 7.3 shows the COVID-19 detection results of ten runs. Notice that, in all the cases, the network produces zero FPs. Setting specificity, precision, and Positive Predicted Value (PPV) to a fixed 100% is a fundamental requirement in this kind of tool. Our reported accuracy in COVID-19 detection is higher than 95% in all the cases, with a very small standard deviation stating the robustness of our method. Also, the average sensitivity is 78%. One of the reasons for the difference between accuracy and sensitivity lies in the large disparity in the CT numbers (i.e. pixel intensities) for COVID-19 and no-COVID-19 subjects.

### 7.4.2 Comparison to the State of the Art

The definitive proof of reliability of our method is given by the comparison of our results with the current state of the art. In Table 7.4, we compare our average and the best results to four articles, described in Section 7.1.2.1, and a work from Harrison *et al.,* [47], where ten radiologists detected COVID-19 during CT reading sessions. We can infer that our results outperform the state of the art in almost all the evaluation metrics and ensure very strong reliability with solid results in precision and PPV. Particular relevance should be given to the accuracy level which is almost 8% better than the maximum value obtained in the other works from the literature. The results of sensitivity are, however, in line with the other works. The highest results in this column have been achieved by Hu *et al.,* [52] and Wu *et al.,* [119]. With regard to this last one, we cannot discuss the general goodness of the work because of the absence of information regarding the other parameters. About the latter, we can attest that it is balanced in sensitivity, specificity,

and precision and this suggests that it can be considered reliable. The same assertion can be done for Zheng *et al.,* [130]. Despite all, considering the results of the radiologists and all the information available to us, our work has shown remarkable results.
As mentioned in Section 7.2.4, we trained and tested a UNet architecture using the same data augmentation that we used for our method. The comparison between the methods described in Section 7.1.2.2 and UNet is shown in Table 7.5. As expected, the results obtained by UNet are very different depending on the dataset used. We can see that the F1-score floats between 40.1% and 82.0%. A difference of almost 42%, using the same method, does not only depend on data augmentation or hyper-parameter setting but also it is influenced by data annotation, ground truth creation, and the variety of cases. Our choice to use a multi-vendor and multi-center image dataset has been made also in consideration of this aspect. Therefore, our purpose is to develop a reliable method that can work with data acquired from several and different sources. In fact, from Figures 7.5 and 7.6, an expert eye can notice that, although all the images have been pre-processed in an identical way, there are significant differences between them in terms of image properties, acquisition methods, resolution and, most importantly, annotation methods. These big differences between the image datasets used can affect the inferences deducted from the bare comparison between the performances of the proposed networks. The direct comparison with the literature tells us that our method is, on average, the most sensitive, specific, and accurate one. Particular emphasis should be given to the balance between sensitivity and precision, and the level of specificity which ensure the almost complete absence of gross errors of segmentation, especially in areas external to the lungs. This is already a distinctive accomplishment, but the data that really prove the goodness of the proposed method are shown in Table 7.6. Excluding accuracy and specificity that were already well over 96%, the improvement over the results of UNet is significant in each field and greater than the ones obtained by all the other methods compared. Indeed, we gain an improvement of 38.2% in F1-score and 28.8% in sensitivity. That means 3.7% and 11.1% more (in F1-score and sensitivity, respectively), with respect to the best method that we have compared.

The strongest limitation of our work is the quantity and quality of the datasets used. This not only reduces the number of examples that our network can study but also does not allow us to perform deeper studies, such as the inter-observer variability or the accuracy of our method based on the area of the lung afflicted by COVID-19 lesions. Furthermore, with a unique and publicly available dataset, we could have done a direct and fairer comparison with other works. Another limitation is given by the hardware used for the experimental part. Implementing a similar method directly on 3D CT volumes could theoretically give more accurate results. However, our available hardware did not allow us to train a network of such complexity on a whole 3D CT image, or at full resolution of $512 \times 512 \times 512$ voxels.

FIGURE 7.5. Examples of the results obtained with CoLe-CNN+ (in green) compared with the ground truth (in red). In yellow, is the superposition of the two segmentations.

FIGURE 7.6. Other examples of the results obtained with CoLe-CNN+ (in green) compared with the ground truth (in red). In yellow, is the superposition of the two segmentations.

| Paper | Method | CTs | IoU | Acc | Spec | Sens | Prec | F1-score |
|---|---|---|---|---|---|---|---|---|
| Fan *et al.*, [31] | UNet | 129 | - | - | 85.8 | 53.4 | - | 43.9 |
| | Semi-Inf-Net | 129 | - | - | 96.0 | 72.5 | - | 73.9 |
| Wu *et al.*, [119] | UNet | 810 | 54.1 | - | - | - | - | 65.1 |
| | JCS | 810 | **66.5** | - | - | - | - | 78.3 |
| Wang *et al.*, [118] | UNet | 558 | - | - | - | - | - | 70.3 |
| | COPLE-Net | 558 | - | - | - | - | - | 80.3 |
| Chen *et al.*, [16]† | UNet | 110 | - | 83.0 | - | - | 79.0 | 82.0 |
| | theirs | 110 | - | 95.0 | - | - | **89.0** | **94.0** |
| Yan *et al.*, [123] | UNet | 861 | - | - | - | 73.6 | 66.2 | 68.8 |
| | theirs | 861 | - | - | - | 75.1 | 72.6 | 72.6 |
| Elharrouss *et al.*, [30] | UNet | 98 | - | - | 85.8 | 53.4 | - | 43.9 |
| | theirs | 98 | - | - | 99.3 | 71.1 | 85.6 | 78.4 |
| Ma *et al.*, [79] | mod UNet | 161 | - | - | - | - | - | 67.3 |
| Yu *et al.*, [126] | theirs | 100 | - | - | 98.3 | 79.1 | - | 77.9 |
| ours | UNet | 179 | 29.1 | 96.6 | 98.3 | 50.6 | 49.7 | 40.1 |
| | our average | 179 | 61.3 | 98.9 | 99.5 | 76.8 | 77.5 | 73.9 |
| | our best | 179 | 65.2 | **99.1** | **99.6** | **80.4** | 80.5 | 78.3 |

TABLE 7.5. Average results for our segmentation algorithm compared with the SoA. Results are given in percentage. †Chen *et al.*, [16] calculated F1-score with a different formula that takes into account not only the positively predicted pixels but also the negative ones, which makes difficult the direct comparison. This explains also the unexpected closeness of F1 with the accuracy.

| Paper | IoU | Acc | Spec | Sens | Prec | F1-score |
|---|---|---|---|---|---|---|
| Fan *et al.*, [31] | - | - | +10.2% | +19.1% | - | +30.0% |
| Wu *et al.*, [119] | +12.4% | - | - | - | - | +13.2% |
| Wang *et al.*, [118] | - | - | - | - | - | +10.0% |
| Chen *et al.*, [16] | - | +**13.0**% | - | - | +10.0% | +12.0% |
| Yan *et al.*, [123] | - | - | - | +1.5% | +6.4% | +3.8% |
| Elharrouss *et al.*, [30]† | - | - | +**13.5**% | +17.7% | - | +34.5% |
| ours aver | +32.2% | +2.3% | +1.2% | +26.2% | +27.8% | +33.8% |
| ours best | +**36.1**% | +2.5% | +1.3% | +**28.8**% | +**30.8**% | +**38.2**% |

TABLE 7.6. Average increment for each segmentation algorithm compared with UNet, tested on the same dataset.

## 7.5 Conclusions on COVID-19 Pipeline

In this work, we propose a novel routine for lung segmentation, COVID-19 detection, and lesion segmentation. Thanks to an accurate lung segmentation and a sensitivity near 99%, we have been able to reach a PPV value of 100%. We did not observe any FP on the ten different runs of the network, with an average accuracy of $97.1 \pm 1.0\%$. Our sensitivity was therefore in concordance with the one achieved in average by a pool of 10 radiologist[9]. Regarding lesion segmentation, we proved that our method is, in absolute numbers, competitive with the best current methods in the state of the art, with an average accuracy of $98.9 \pm 0.3\%$. In addition, we showed that our method improved the results obtained by UNet on the same dataset, by the largest quantity. In fact, we gained 38.2% on F1-score and 36.1% on IoU, over UNet.

[9]The two tests have been done on different data because the dataset used by the radiologist was not accessible.

CHAPTER 8

*We must state **relationships**, not procedures.*

GRACE HOPPER

# OPTIMAL LUNG PROJECT

The line of research we carried on in these years is just a part of a larger internal research project called *DeepLung*. After about 5 years of research, the group of Data Analytics in Medicine (DAM) at Eurecat has been able to produce several neural networks for the detection, segmentation, and characterization of lung nodules in chest CTs. All are witnessed by several publications [9] [89] [90] [94] [95] [96] [97] and conference talks.

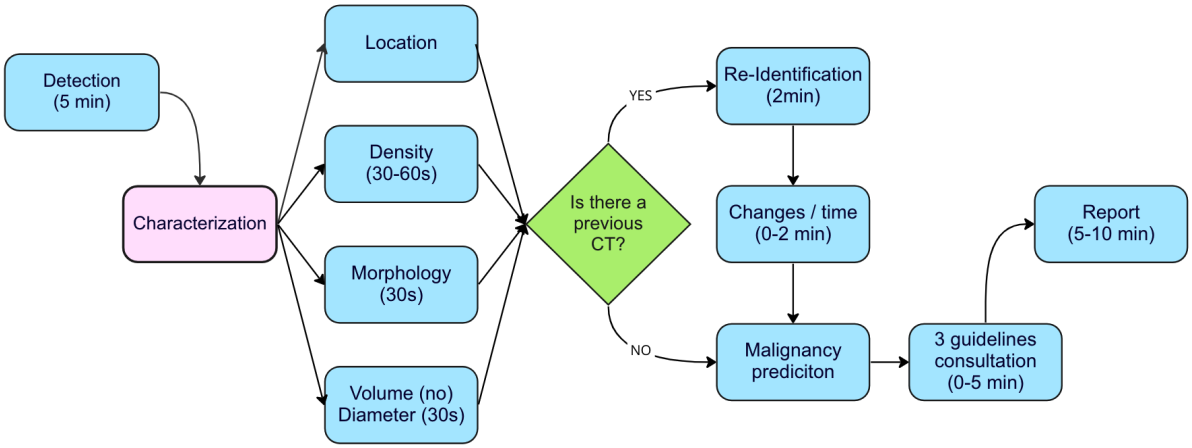



FIGURE 8.1. Resume of the current workflow of radiologists in lung cancer on CT images.

## 8.1 Current Practice

Radiologist's workflow for the identification and classification (benign, uncertain, malignant) of lung nodules in chest CTs can be summarized in the following steps (Figure 8.1):

1. Detection of lung nodules: radiologists scroll the chest CT looking for nodules in the parenchyma of both lungs, discarding those that are generally too small (< 6mm) since most of the time are considered benign. The sensitivity of radiologists in detecting pulmonary nodules ranges from 64% to 82%, with specificity around 87% [82].

2. Characterization: once the most important nodules have been identified the radiologists evaluate specific characteristics to determine their nature. The main characteristics evaluated are:

    a Location: nodules located in the upper segments of the lung have a higher risk of being malignant.

    b Density: nodules can be solid, sub-solid, or non-solid (ground-glass). Non-solid nodules are hard to assess, and in fact, 91% of missed lung nodules are subsolid or non-solid [68].

    c Morphology: nodules can have different shapes and margins, and some of them tend to indicate a higher risk such as spiculated or irregular margins while polygonal morphology is usually benign.

    d Size: it is generally evaluated using the maximum diameter in an axial plane. However, this method of measuring the nodule can be misleading since often the maximum axial diameter is not the largest one. The determination of the volume is recommended since it is a better estimation, but it is not calculated in current clinical practice because it is too time-consuming.

3. Re-identification and changes over time: previous CTs are checked for determining if the nodule was present or not in previous scans. A growing nodule is generally considered suspicious.

4. Malignancy and guidelines: based on the previous characteristics the radiologist decides if the nodule is benign, uncertain, or malignant based on the recommendations from the 3 main clinical guidelines (Fleischner Society, Lung-RADS, and British Thoracic Society guidelines).

5. Report: The radiologist finally writes a report including all the findings and the suggestions for the follow-up diagnosis or treatment.

## 8.2 Our proposal

The *Optimal Lung* project, based on the research developed in the DeepLung project, was born in March of 2022 and its target is to envelop all the technologies developed in DeepLung in one unique service working inside the Hospitals. Our service will automatically collect chest CTs from the PACS[1], as soon as they are acquired from the machine, analyze them in real-time and send them back to the PACS with an additional report saved in the HIS[2]. In this way, during the CT reading, radiologists will find extra information on-screen in their native viewer.

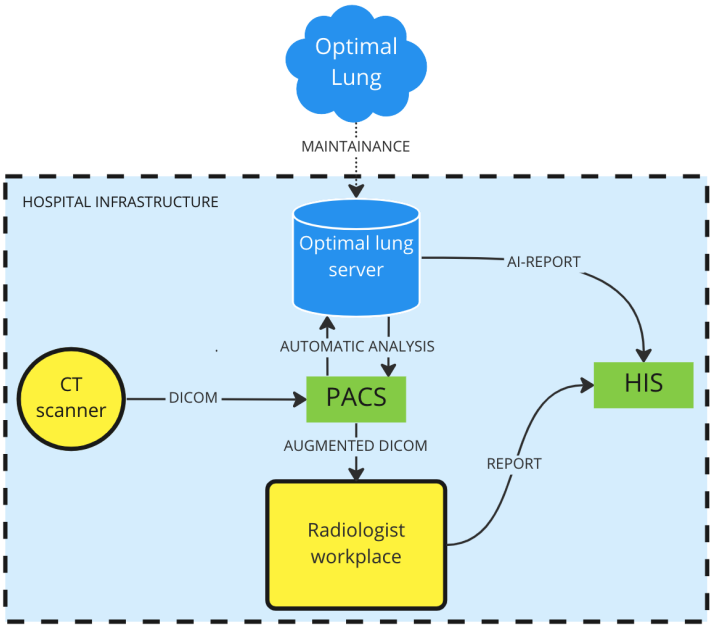



FIGURE 8.2. Basic level graph of the workflow of the CT through our service and back to PACS and HIS systems.

The philosophical idea on which we base our whole work is that is more efficient to have several specialized neural networks than to have a mighty one that tries to do everything at once.

Our detection algorithm is divided into three parts: i) one for detecting all the possible candidates to be nodules; ii) one for checking each candidate and excluding all the non-nodules (e.g. blood vessels); iii) one for reducing the false positive (e.g. lymph nodes). Our segmentation algorithms explained before, predict a series of different masks, before combining them and extracting the most probable one. We use it to segment the volume of lungs, nodules, and COVID-19 lesions, each one with a different neural network. Our cancer prediction algorithm is composed of three neural networks, predicting: i) the expected malignancy of the nodule on a scale from 1 to 5, based on radiologists' opinion; ii) the probability of being malignant of the nodule, in percentage; iii) the future growth per unit of time, presented directly on the image. We also have algorithms for morphology prediction, currently under development.

Optimal Lung operates in every single one of the previously enumerated steps. It detects

[1]The *Picture Archiving and Communication System* is a database technology to securely store medical images and clinically-relevant information. It is also used to transmit information digitally to the authorized medical personnel.

[2]The *Hospital Information System* is a comprehensive, integrated information system conceived to store all the information related to all sort of a hospital's operation, from medical to administrative. It includes all the clinical history of every patient who entered the hospital and the links to their medical images.

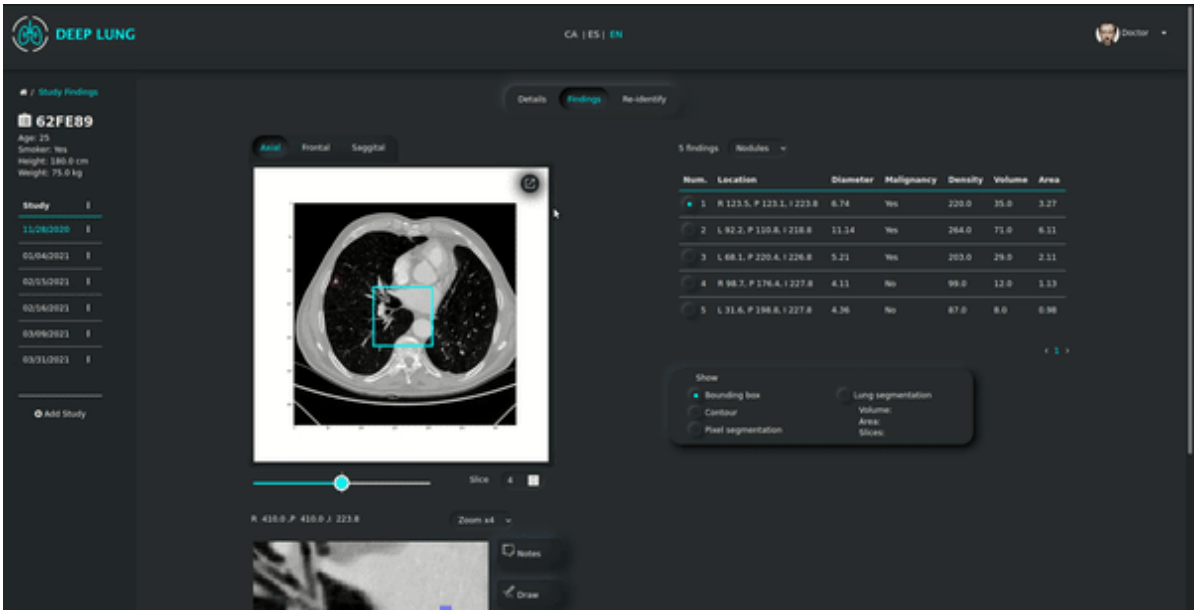



FIGURE 8.3. This is an example of the appearance of the results, in an internally developed visor.

and characterizes suspicious nodules, presenting on-screen all their physical details and the probability of being malignant. It can compare two CTs of the same patient and write a report, pending the approval of the radiologist. All this in a few minutes and before the radiologist starts analyzing the study. Optimal Lung is a service intended to support radiologists, emulating their exact workflow, following the same order. Intuitively, if a practice is well-established and working on for decades, it is wiser not to change it. The goal of Optimal Lung is to automatize all the repetitive and time-wasting operations, with the main aim of reducing the reading time of a CT and giving precise results, in line with the official radiological guidelines. To the best of our knowledge, we are the first ones bringing the "clinical pull" approach to this level.

## 8.3 Results and Statistics

All our algorithms obtained State of the Art (SoA) results on public datasets, including multi-center and multi-vendor data. In this short report, we will present and comment on the most crucial results for the purposes of our tool, summarizing them in the following three categories.

### 8.3.1 Detection

One of the biggest issues and adoption stoppers for radiologists is a high number of False Positives (FP) in the phase of detection. Although, the maximum acceptable number of FPs is not the same for every radiologist.

We developed two neural networks to separate the tasks of detection. The first one detects and saves the coordinates of what we call the *candidates*, i.e. all the suspicious masses[3]. The second one inspects all the candidates, one by one, and selects only the ones that are considered nodules, based on a system of grades. The threshold for selection can be adjusted based on the needs of the user. Therefore, our algorithm has a method for further reducing the number of FP, depending on the radiologist's requirements, but influencing the sensitivity. In Figure 8.4 we present how the sensitivity changes as function of the required False Positives. The overall statistics that our nodule (including benign and possible malignant ones) detection algorithm [94] [95] reaches on the LIDC-IDRI dataset [3], with an average number of 2 FP per CT, corresponds to a sensitivity of 89.6% and a specificity of 99.9%.

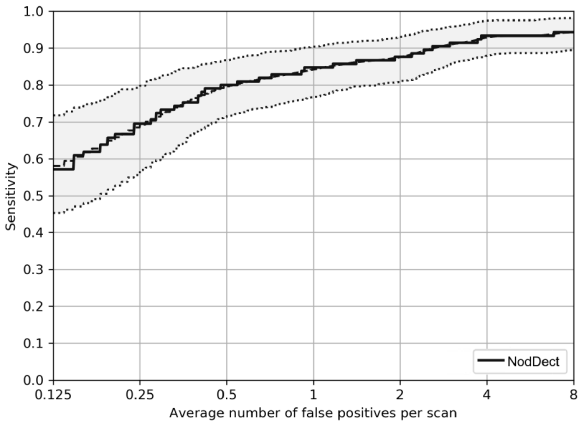



FIGURE 8.4. Sensitivity variation with False Positives

### 8.3.2 Segmentation

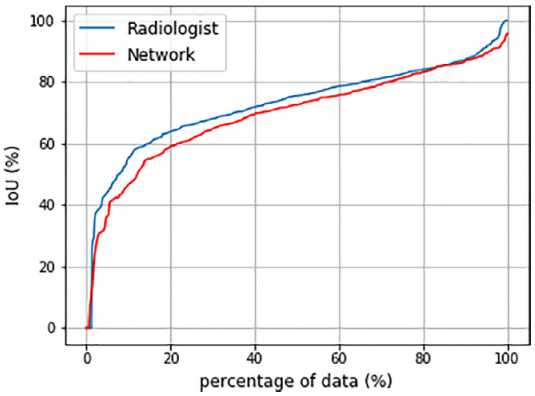



FIGURE 8.5. Cumulative plots of Intersection over Union for Neural Network and Radiologist compared with ground-truth on 61 images.

The main scientific contribution given by this Ph.D. programme regards the topic of segmentation and feature extraction. Segmentation is a critical feature for helping radiologists in their current routines. In fact, all the guidelines for radiologists suggest calculating the exact volume of the nodule to accessing its growth in time and its malignancy. Performing a precise segmentation, it is a very time-expensive procedure and can take up to one hour for a single 3D CT [115]. Instead, because of the overflow of work, radiologists commonly use the largest axial diameter of the nodule, which is not always a good approximation. Our NN [89] (explained in detail in Chapter 6.1) is able to segment the nodule and calculate its volume with sensitivity and precision up to 90% and overall performance

[3]The suspicious masses include all those objects that should not appear in fully health lungs. Not only nodules but, for example: lymphnodes, infections, aneurysms, or even traces of a past pneumonia, and more.

at human-like level. As shown in Figure 8.5, the cumulative IoU of our algorithm is exactly at the same level as the manual segmentation of a radiologist, after the first 80 to 90% of the test cases. As explained in Chapter 6.4.4, our algorithm places itself among the best in the current state of the art. We also developed tools for segmenting COVID lesions and lung volume. In addition, the first one can be fine-tuned to detect and segment other visible lung pathologies, giving an estimation of the volume and portion of the lung afflicted. Instead, lung segmentation is currently used to improve accuracy, but it can also provide experts with the morphological location of the object of interest or help the 3D reconstruction of the image. Our average accuracy in segmenting COVID lesions [90] (explained in detail in Chapter 7.1) is 97.1%, and our accuracy in lung segmentation [90] is over 99.5%.

### 8.3.3 Prediction

| | Positive predicted | Negative predicted |
|---|---|---|
| Positive | 42 | 8 |
| Negative | 6 | 26 |

TABLE 8.1. Confusion matrix results for biopsy outcome prediction, from CT images of 82 patients.

The part of prediction [96] [97] is the most sensitive one because the treatment, and maybe the survival of the patient, may depend on this. For understanding the nature of a suspicious mass, the biopsy is the current standard because imaging is not always sufficient. The average number of lung cancer biopsies per patient before a diagnosis is 1.7 in the United States [129]. From an independent study [53], the levels of sensitivity and specificity of a radiologist predicting the outcome of a lung tissue biopsy, by only reading one or more CT images, are around 70% and 69%, respectively. In a retrospective internal study on 82 patients with CT scan and biopsy results [9], our algorithm has been able the predict correctly 68 of the outcomes of the biopsy, as shown in Table 8.1. The overall sensitivity and specificity in this experiment resulted to be 85% and 81%, respectively. In that same work, our neural network has been compared against a radiologist who achieved a sensitivity of 79% and a specificity of 71%.

## 8.4 Evidence

Currently, our image databases are composed of over 30 000 CT scans coming from public and private datasets, from a large variety of individuals, from different scanners, and with a variable dose of radiation. For our publications and internal validations, we used parts of this database, depending on the annotations we have on each case.
Our future plan is to start two parallel retrospective studies in partner hospitals (Parc Taulí and

Vall d'Hebron) with more than 1500 participants in total, for validation and the application for the CE mark.

On the side of research, our gaze is projected on the newest technologies and we are already in the process of developing new AI solutions. In particular, we are focusing on Transformers for the part of the detection and prediction, trying to overcome the difficulties of adapting a two-dimensional tool to three-dimensional images. For the part of characterization, we are exploring the possibility to use Diffusion methods, which are surprising the whole world with their ability in creating new images never seen before.

Our second objective is to determine if the diagnostic accuracy of radiologists with AI Computer-Aided Diagnosis (CAD) assistance is superior to the diagnostic accuracy of radiologists without AI-CAD in localizing and classifying lung nodules.

We designed our study to be an observational retrospective case-control with two readers in a blinded multi-case environment. The subjects will be more than 1500 asymptomatic adults (> 18 years old) with CT scans acquired during routine CT examination, 50% of them without lung nodules and 50% with one or more lung nodules. Data will come from 2 public hospitals, under GDPR regulations and authorization from the respective Ethical Committees. Data from different CT providers will be necessary. The ground truth for nodule detection would be set by the annotations done by 2 radiologists with at least 5 years of experience in thoracic radiology. The ground truth for nodule classification would be biopsy-proven or by consensus of 2 or more radiologists. For each CT scan, the reading process will take place in two steps: i) a reader's stand-alone reading without Optimal Lung assistance; and ii) a second reader's reading with the support of Optimal Lung. The time spent doing all those actions will be measured live.

## 8.5 Impact of the solution

Our solution would have the highest impact under a lung cancer screening program. This is a method of prevention in finding lung cancer in the late stages when the means of intervention are limited. The people invited to the screening are usually smokers or people who have a predisposition because of other conditions, even genetical. They should have one yearly CT scan and, talking about several thousand every million people, the number of images to be analyzed would increase sensibly.

The United States, China, and the United Kingdom have already started to screen large groups of people with impressive results in terms of early detection. Italy, Spain, Poland, and other countries, to date, are running trials of feasibility of the screenings. European Union released in September 2022 a guideline for lung screening programs and has already placed over a hundred million euros for their implementation[4]

[4]Source: European Commission.

From an independent cost-effectiveness analysis run by Fundació HITT (Health Innovation Technology Transfer), came out that, under a lung cancer screening program, our product could:

- Reduce by 23% the number of patients diagnosed with late-stage lung cancer
- Reduce by 22% the number of uncertain diagnoses[5]
- Increase by 7% the 5-year survival rate of patients diagnosed with lung cancer
- Reduce by 6.5% the implementation costs of the screening program
- Reduce by 25% the time needed to read a single CT or the number of radiologists needed to implement the screening program
- Reduce the costs up to 122.57€ per CT scan analyzed

In fact, the cost for the treatment of a patient diagnosed in a late stage (including everything from analysis to discharge) is 110000€, on average. While for an early detection case it is 36000€, on average.

## 8.6 Wild Card

Together with Dr. Andreu Antolín, radiologist at the hospital Vall d'Hebrón, and Eduard Solér, director of the innovation department at hospital Parc Taulí, we participated in the Wild Card[6] program (Figure 8.6) organized by EIT Health of the European Union, for promoting our project of lung nodules detection and characterization in CTs.

We started this experience in March 2022, when we met for the first time and started defining our idea for the project. We passed in April the first two selections, one determined by our written proposal and one by an interview with one of the organizers of the program. In June, we have been invited to a three-day session in Wien (Austria) where we have been taught how to effectively communicate with collaborators, partners, and investors but also within the team. From July 4th until 8th, we had a full-immersion week of mentoring. We have been followed and advised by Wild Card's experts, connecting from all over Europe. Their expertise ranged from clinical to technical and financial. They put under stress our project, focusing on the need for our solution and the possible applications and outcomes. On the last day, we presented a pitch, going into the details of our idea, and we have been selected for the final phase. This last phase started on August 16th with only the best eight projects participating, four in early detection of cancer and four in mental health. The program consisted of two weekly meetings, spread over nine weeks, with mentors equally divided between the United States and Europe. They found all the pitfalls in the

[5] An uncertain diagnosis needs another CT scan (or another exam), leading to an increase in the length of the waiting lists, extra stress for the patient, and extra costs.

[6] Website of Wild Card.

implementation of our solution and helped us re-thinking the project by applying a well-known method called GAITS[7]. During these weeks we conducted ninety interviews with experts from the field, such as radiologists, start-up owners, investors, notified bodies, and even CEOs and CTOs of some of the competitors or the most impactful companies in the digital health market. They gave us an insight into how this world works and what is needed to arrive on the market. The last event of Wild Card was held in Münich at the headquarter of a massive pharmaceutical company named Amgen. There, the eight finalists pitched for 45 minutes in front of a jury of four investors and, in the end, only two startups have been financed. Optimal Lung did not make it to the funding, but the Wild Card program left us with a suitcase full of experience and knowledge in the field, in addition to the awareness of having classified among the best four startup projects in the early detection of cancer, among hundreds of participants, this year in Europe, with only eight months of work!

We still firmly believe in our idea and we want to bring it to the market. With the support received from our three institutions, and that we will receive in the future, we are confident that it is just a matter of time before that Optimal Lung will become a reality.

[7]Website of GAITS.


FIGURE 8.6. The three components of the team participating in the Wild Card at the final in Münich. From left to right: Andreu Antolín, Eduard Solér and Giuseppe Pezzano

CHAPTER 9

# CONCLUSIONS

*Success in creating AI would be*
*the biggest event in human history.*
*Unfortunately, it might also be the* ***last****...*

STEPHEN HAWKING

In this new millennium, artificial intelligence is helping humans in automatizing and optimizing complex tasks, especially in the field of computer vision. We are seeing a radical change in the way we process images. In fact, we are able to extract several very different features from a single image, often at a super-human level, thanks to deep learning. The time when this new technology will be largely spread in all the fields and constantly present in our everyday life is just around the corner.

We have seen AI expanding in all healthcare fields. Imaging is the most fashionable one, but deep learning is producing good results in all of them. For example, surgery is one of those practices in which even the slightest mistake can produce severe or fatal results. In my humble opinion, if humanity is accepting this new technology in such a delicate environment, it means we are quickly moving toward a world where AI will work side-by-side with clinicians in all the medical fields. Though the medical field will follow with a bit of delay, AI will inevitably fully enter this world and give benefits to it. From my personal perspective, having talked about this topic with many experts, the question is not "if" AI will ever be used in healthcare, but "when"!

Regarding the fundamental research in machine learning, we are looking for new methods to improve the already existing technology. In our work on transformers, introducing two different criteria that establish which layers should be frozen and at which point of the training, we successfully trained the most common transformers saving more than one-third of the time on average and without decreasing their accuracy, neither increasing the number of parameters. Working on convolutions, we proposed a reformulation of the convolutional operator and the necessary conditions to guarantee the invertibility of $3 \times 3$ convolutions. We also present a closed form for the determinant of the matrix associated with this convolution and, thus, the means to invert this operator. Through the application of these invertibility conditions to the loss function, we have succeeded in training CNNs whose convolutional layers are invertible. This

work provides new insights into CNNs, opening the possibility of using them in new ways for image generation and classification. Most importantly, it provides a tool for better assessing how information is processed and abstracted by CNNs.

Regarding our research on the applications of AI in the medical field, we developed Neural Networks able to segment the whole volume of the lungs with an accuracy of over 99% and find the borders of suspicious masses, such as nodules, lymph nodes, and areas affected by pneumonia, with an accuracy equal to or greater than the best results in the state of the art. These algorithms are implemented in a pipeline able to reproduce the same exact workflow of a radiologist. We believe this technology will help improve and fasten the work of chest CT reading and will be at all radiologists' disposal in the near future. In addition to that, we are aware of the fact that we got the attention of one of the largest European innovation programs, plus two large hospitals in Spain (with a reference population of over two million people), and one of the most important European research centers.

## 9.1 Future Lines

The new emerging architectures are gradually fixing the pitfalls of deep learning and discovering new horizons of research, while the scientific community is always ready to take these innovations and convert them into products. Transformers and diffusion methods are partially solving the largest problems of detection and image generation, imposing themselves as the best architectures so far, but there is still a long way to go and new methods will surely steal their crowns, sooner or later. Meanwhile, on our side, we study to contribute to this cause and stay alert to spot the newest technologies and apply them in the medical field.

At this point, the limitations of our tools are still important. Our optimization algorithm for transformers still needs to be tested on very large datasets and could be improved by keeping working on the properties of the embedded matrices. Although, given the importance of the subject and the great latest achievements obtained by transformers in Large Language Models (LLM), will surely bring a wind of innovation, especially in this particular technology. The method for convolution inversion works very well on small and medium size images but it can stop working on high-resolution images because of the machine rounding error. The time needed for the inversion of the neural network is negligible, compared with training time but it grows exponentially when the size of the images increases. There is still more work to be done in software engineering before it can be used on large scale. But this current of innovation can provide new insights into CNNs, opening the possibility to use them in new ways for image generation. For example, Variational Auto-Encoders (VAEs), Generative Adversarial Networks (GANs), and Diffusion models are the models that could receive the largest profit from applying our method.

Regarding lung nodule segmentation, the natural following step of this research is to convert this Neural Network into its 3D formulation. During the reading sessions, the radiologists visualize the CT scans slice by slice (as we input them into the network). Then, even if the presented 2D method could evolve into a very useful instrument, a three-dimensional version of it surely will give an extra value in terms of the number of extractable features, opening the doors to a more accurate 3D reconstruction. Nevertheless, the network learns the information we provide to it. Thus, our efforts should also focus on the development of a more trustable and richer database and new tolerant loss functions, as the one proposed in this work. However, whenever we have manual annotations we have an error. Thus, the question is how to develop robust methods of working and learning from noisy ground-truth. In this sense, another future outlook would be to develop a reliable generative network for producing more CT images with computer-made nodules.

Similar considerations can be done on the COVID-19 pipeline. The robustness and the accuracy of this work open up a wide range of other possible applications of our method. For example, the proposed network could be adapted, using fine-tuning, for studying the worst cases of pneumonia, diffused metastasis, or other lung diseases. Also, our methodology could be applied to detect and segment a large variety of organs in other fields of medical imaging analysis.

Our work on deep learning in healthcare has been appreciated by many experts but still, we have to test it in clinical environments and or a large variety of hospital data, before it could have a real impact. In addition, there is still work to do on the side of feature extraction and integration with the hospital systems.

Every day our relationships with the hospitals become closer and, guided by their expertise, we contribute to the find solutions to the problems they find in their daily routine, always for better care of the patient.

# List of Tables


3.1 Classification accuracy results and reconstruction errors . . . . . 32

4.1 Image classification results on ImageNet . . . . . 38
4.2 Our results obtained on Food-101 dataset . . . . . 41

6.1 Scheme of the layers and number of filters . . . . . 58
6.2 Results for ours ten runs . . . . . 64
6.3 Test of different depths . . . . . 67
6.4 Test of different MCLs . . . . . 68
6.5 Test of different loss functions . . . . . 68
6.6 Average F1 score calculated for each radiologist . . . . . 69
6.7 Average Precision calculated for each radiologist . . . . . 69
6.8 Comparison of our method to the state-of-the-art . . . . . 71
6.9 Comparison with the results of Unet++ . . . . . 71
6.10 Kolmogorov-Smirnov test on IoU . . . . . 72

7.1 Results for lung segmentation . . . . . 84
7.2 Result for COVID-19 lesion segmentation . . . . . 84
7.3 Results for COVID-19 detection . . . . . 85
7.4 Results in classification and comparison with SoA . . . . . 86
7.5 Results for segmentation algorithm compared with the SoA . . . . . 90
7.6 Comparison with SoA using UNet as benchmark . . . . . 90

8.1 Confusion matrix results for biopsy outcome prediction . . . . . 98

B.1 Average IoU between the radiologists . . . . . 113
B.2 Average Sensitivity between the radiologists . . . . . 113

# List of Figures


1.1 Computer units shipped per year . . . 3
1.2 Usage in time of the wording: Artificial Intelligence . . . 4
1.3 Adversarial example on EfficientNet . . . 5
1.4 Marble statue using a computer . . . 6
1.5 Spiderman presenting his poster at data science conference . . . 7
1.6 Then a miracle occurs . . . 8

2.1 AlexNet . . . 12
2.2 ResNets training on ImageNet . . . 13
2.3 Examples of Inception layers . . . 14
2.4 Top-1 accuracy. Inception Vs. ResNets . . . 15
2.5 U-Net architecture . . . 16
2.6 Vision Transformer . . . 17
2.7 Pipelines of the transformers used in this work . . . 19

3.1 The two versions of our convolutional layers . . . 30
3.2 Example of the pooling layer . . . 30
3.3 Flowcharts of the test architectures . . . 30
3.4 Evolution of the determinant of the associated matrix . . . 32
3.5 Example of image reconstruction . . . 33
3.6 Example of the maximum image reconstruction error . . . 34
3.7 Histograms of the maximum image reconstruction error . . . 35

4.1 Accuracy evolution of Transformers during fine-tuning . . . 42
4.2 $C_1$ criterion on $W_v$ matrices for Deit . . . 43
4.3 Maximum value of the $C_2$ criterion on $W_q$ matrices for Deit . . . 43
4.4 Evolution of the singular values of $W_v$ matrices of DeiT . . . 44
4.5 Frobenius norm of the weight matrices . . . 44

5.1 Healthcare transformation . . . 47
5.2 Cancer Mission . . . 49
5.3 Number of cancer diagnoses at stage IV . . . 50

6.1 Proton therapy dose deposition . . . . . 54
6.2 Structure of our NN . . . . . 58
6.3 Structure of the MCL . . . . . 59
6.4 Example of high expert inter-variability . . . . . 65
6.6 Examples of segmentation in cases of wrong labeling . . . . . 66
6.5 Histogram of the F1 score results . . . . . 66
6.7 Examples of segmentation in other interesting cases . . . . . 67
6.8 MCL used in this study . . . . . 68
6.9 Cumulative Fraction Plot for IoU . . . . . 70

7.1 COVID-19 GGO examples . . . . . 74
7.2 Architecture of our CoLe-CNN+ . . . . . 78
7.3 Architecture of our MCL . . . . . 79
7.4 Architecture of the decoder for COVID-19 detection . . . . . 81
7.5 Examples of the results obtained with CoLe-CNN+ . . . . . 88
7.6 Examples of the results obtained with CoLe-CNN+, other . . . . . 89

8.1 Current workflow of radiologists in lung cancer on CT images . . . . . 93
8.2 Workflow of the CTs inside hospital systems . . . . . 95
8.3 DeepLung front-end example . . . . . 96
8.4 Sensitivity variation with False Positives . . . . . 97
8.5 Cumulative plots of Intersection over Union . . . . . 97
8.6 The three components of the team . . . . . 102

B.1 Examples of segmentation in presence of filaments and other masses . . . . . 115
B.2 Examples of segmentation in very noisy images . . . . . 115
B.3 Examples of segmentation in low contrast images . . . . . 116
B.4 Examples of segmentation in juxta-pleural cases . . . . . 116

APPENDIX A

# APPENDIX A

## A.1 Lemmas on Convolution

**Lemma A.1.1.** ***(Vector convolutional operation)***
*Let us have one row vector $\tau$ and one column vector $\varrho$ of size 3 (as in Equation 3.5). We define calculate their convolution as their outer product. In formula:*

$$\tau * \varrho = \begin{pmatrix} a & b & c \end{pmatrix} * \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{bmatrix} \alpha a & \alpha b & \alpha c \\ \beta a & \beta b & \beta c \\ \gamma a & \gamma b & \gamma c \end{bmatrix} = \tau \otimes \varrho^{\mathsf{T}}$$

*The outer product respects all the properties of convolution between vectors. In addition, the matrix associated with the kernel resulting from the vector convolution is equal to the matrix of Equation 3.8, in accordance with our formalism.*

**Lemma A.1.2.** ***(Determinant of a tri-diagonal block Toeplitz matrix with diagonal blocks)***
*Let us consider $T$ a squared tri-diagonal block Toeplitz matrix, in which each of the blocks are $n \times n$ diagonal matrices:*

$$P = \begin{bmatrix} B & C & 0 & 0 & \dots \\ A & B & C & 0 & \dots \\ 0 & A & B & C & \dots \\ 0 & 0 & A & B & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \qquad \begin{cases} A = \alpha \cdot \mathbb{I} \\ B = \beta \cdot \mathbb{I} \\ C = \gamma \cdot \mathbb{I} \end{cases}$$

*Skipping the trivial* $1\times 1$ *case, and knowing that the determinant of diagonal matrix is the product of the elements on the diagonal, the determinant of the first* $2\times 2$ *block of the* $P$ *matrix is:*

$$\Delta\begin{pmatrix} B & C \\ A & B \end{pmatrix} \equiv \Delta(B)\Delta(B-AB^{-1}C) = \beta^n\left(\beta - \frac{\alpha\gamma}{\beta}\right)^n$$
$$= (\beta^2-\alpha\gamma)^n = \Delta^n\begin{pmatrix} \beta & \gamma \\ \alpha & \beta \end{pmatrix}$$

*Similarly, for the first* $3\times 3$ *block it is:*

$$\Delta\begin{pmatrix} B & C & O \\ A & B & C \\ O & A & B \end{pmatrix} = \Delta(B)\Delta\left(\begin{pmatrix} B & C \\ A & B \end{pmatrix} - \begin{pmatrix} A \\ O \end{pmatrix} B^{-1}\begin{pmatrix} C & O \end{pmatrix}\right)$$
$$= \Delta(B)\Delta\begin{pmatrix} B-AB^{-1}C & C \\ A & B \end{pmatrix}$$
$$= \left[\beta(\beta^2-\alpha\gamma)-\alpha\beta\gamma\right]^n = \Delta^n\begin{pmatrix} \beta & \gamma & 0 \\ \alpha & \beta & \gamma \\ 0 & \alpha & \beta \end{pmatrix}$$

*Assuming that is true for a similar* $(n-1)\times(n-1)$ *matrix, we demonstrate it for the* $n\times n$ *case:*

$$\Delta(P) = \Delta(B)\Delta\begin{pmatrix} B-AB^{-1}C & C & O & \dots \\ A & B & C & \dots \\ O & A & B & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}_{n-1} = \beta^n\Delta^n\begin{pmatrix} \beta-\alpha\beta^{-1}\gamma & \gamma & 0 & \dots \\ \alpha & \beta & \gamma & \dots \\ 0 & \alpha & \beta & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}_{n-1}$$

$$= \beta^n\left[(\beta-\alpha\beta^{-1}\gamma)^n\,\Delta^n\begin{pmatrix} \beta & \gamma & 0 & \dots \\ \alpha & \beta & \gamma & \dots \\ 0 & \alpha & \beta & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}_{n-2} - \alpha^n\gamma^n\,\Delta^n\begin{pmatrix} \beta & \gamma & 0 & \dots \\ \alpha & \beta & \gamma & \dots \\ 0 & \alpha & \beta & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}_{n-3}\right] \tag{A.1}$$
$$= \Delta^n\begin{pmatrix} \beta & \gamma & 0 & \dots \\ \alpha & \beta & \gamma & \dots \\ 0 & \alpha & \beta & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}_n$$

where Equation A.1 comes out calculating the determinant for the $n\times n$ case, at the second step of the Laplace expansion. In the previous equation, we inserted the size of the matrix as index, and indicate with $\Delta^n$ the determinant at the $n$-th power.

APPENDIX B

# APPENDIX B

## B.1 More about Database Consistency

Tables B.1 and B.2 show more data related to the comparison between the segmentations made manually and ours made automatically. In these tables, we compare the results for IoU and Sensitivity. As expected by looking at the tables referring to the F1 score, the average of the Sensitivity is sensibly lower than the average calculated with the data of the radiologists. The reason for this outcome is still related to the slight tendency to under-segmentation of our method.

| | R1 | R2 | R3 | R4 | Average |
|---|---|---|---|---|---|
| R1 | - | 72.6 | 74.2 | 73.8 | |
| R2 | 72.6 | - | 74.2 | 73.4 | |
| R3 | 74.2 | 74.2 | - | 73.9 | |
| R4 | 73.8 | 73.4 | 73.9 | - | **73.7** |
| ours | 68.1 | 69.4 | 69.2 | 69.0 | **68.9** |

TABLE B.1. Average IoU in percentage between the radiologists. All the uncertainty values for the radiologists range from ±14.8 to ±16.1. From ±16.8 to ±17.3, for ours.

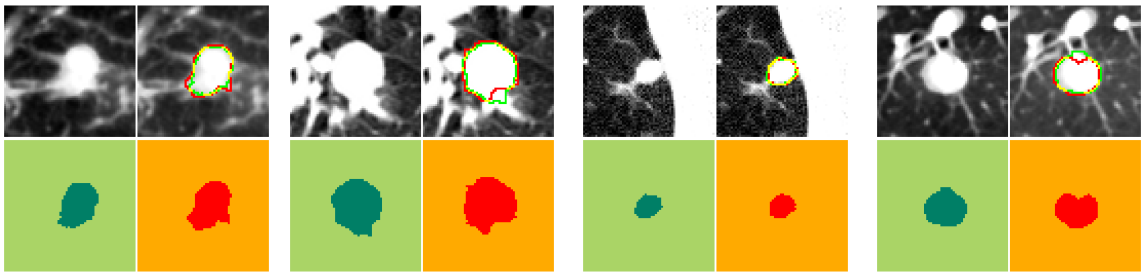
| | R1 | R2 | R3 | R4 | Average |
|---|---|---|---|---|---|
| R1 | - | 85.9 | 86.1 | 83.9 | |
| R2 | 85.9 | - | 84.8 | 85.2 | |
| R3 | 86.1 | 84.8 | - | 86.4 | |
| R4 | 83.9 | 85.2 | 86.4 | - | **85.4** |
| ours | 76.2 | 78.2 | 77.6 | 78.7 | **77.7** |

TABLE B.2. Average Sensitivity in percentage between the radiologists. All the uncertainty values for the radiologists range from ±15.8 to ±16.9. From ±18.0 to ±18.4, for ours.

## B.2 More Interesting Cases

The biggest challenge for a segmentation algorithm is to be reliable under the simplest conditions as well as the most complex ones. Our method demonstrated to have good sensitivity also in the most difficult cases. We hereby want to show more cases of successful segmentation, in presence

of filaments and other masses (Fig.B.1), in very noisy CT images (Fig.B.2), in situations of low contrast (Fig.B.3) and in the case of juxta-pleural nodules (Fig.B.4). All the images are provided with all the respective criteria that we used in this work and explained in Chapter 6.2.5.



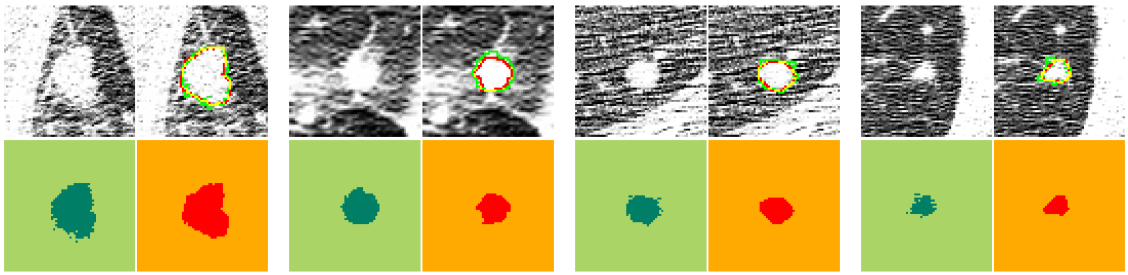
(A)
$\gamma$-index = 99.34%
IoU = 85.29%
Sensitivity = 86.46%
Precision = 98.43%
F1 = 92.06%

(B)
$\gamma$-index = 98.46%
IoU = 81.58%
Sensitivity = 89.08%
Precision = 90.64%
F1 = 89.86%

(C)
$\gamma$-index = 99.78%
IoU = 89.43%
Sensitivity = 94.83%
Precision = 94.02%
F1 = 94.42%

(D)
$\gamma$-index = 98.88%
IoU = 79.69%
Sensitivity = 91.84%
Precision = 85.76%
F1 = 88.7%

FIGURE B.1. Examples of detected and segmented nodules in presence of filaments and other masses. The output of our method is in green, the ground truth in red, and their superposition in yellow.



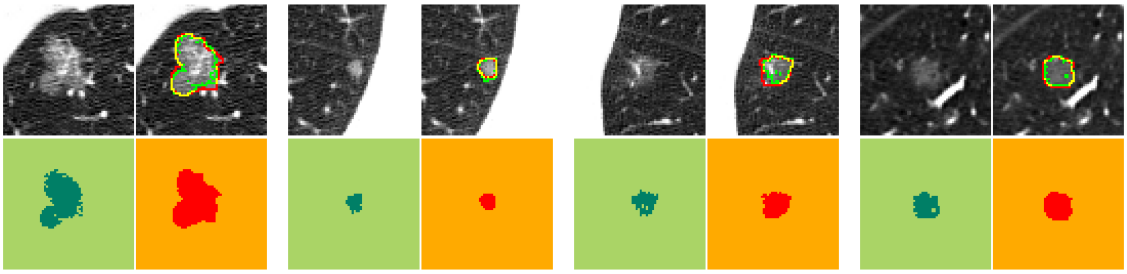
(A)
$\gamma$-index = 99.17%
IoU = 88.2%
Sensitivity = 93.62%
Precision = 93.84%
F1 = 93.73%

(B)
$\gamma$-index = 98.75%
IoU = 71.72%
Sensitivity = 98.87%
Precision = 72.31%
F1 = 83.53%

(C)
$\gamma$-index = 99.34%
IoU = 82.54%
Sensitivity = 97.5%
Precision = 84.32%
F1 = 90.4%

(D)
$\gamma$-index = 99.58%
IoU = 77.23%
Sensitivity = 90.7%
Precision = 83.87%
F1 = 87.22%

FIGURE B.2. Examples of detected and segmented nodules in very noisy conditions. The output of our method is in green, the ground truth in red, and their superposition in yellow.



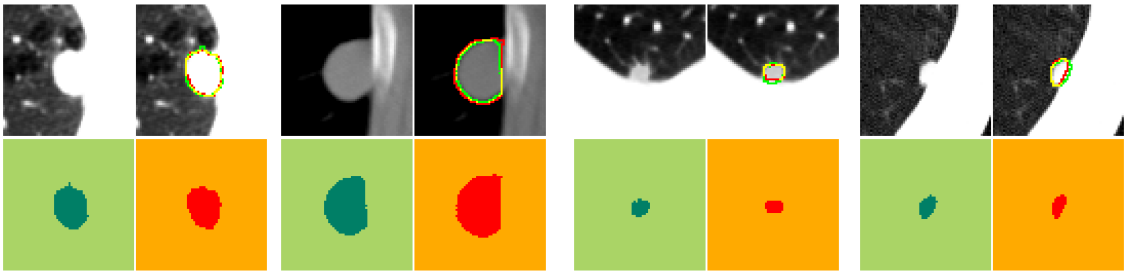
(A)
$\gamma$-index = 98.1%
IoU = 78.53%
Sensitivity = 80.04%
Precision = 97.64%
F1 = 87.97%

(B)
$\gamma$-index = 99.85%
IoU = 80%
Sensitivity = 92.31%
Precision = 85.71%
F1 = 88.89%

(C)
$\gamma$-index = 99.12%
IoU = 65.31%
Sensitivity = 67.13%
Precision = 96%
F1 = 79.01%

(D)
$\gamma$-index = 99.51%
IoU = 78.05%
Sensitivity = 79.5%
Precision = 97.71%
F1 = 87.67%

FIGURE B.3. Examples of detected and segmented nodules in low contrast conditions. The output of our method is in green, the ground truth in red, and their superposition in yellow.



(A)
$\gamma$-index = 99.56%
IoU = 91.55%
Sensitivity = 97.74%
Precision = 93.53%
F1 = 95.59%

(B)
$\gamma$-index = 98.97%
IoU = 87.64%
Sensitivity = 87.64%
Precision = 100%
F1 = 93.41%

(C)
$\gamma$-index = 99.76%
IoU = 74.6%
Sensitivity = 95.92%
Precision = 77.05%
F1 = 85.45%

(D)
$\gamma$-index = 99.9%
IoU = 71.62%
Sensitivity = 92.98%
Precision = 75.71%
F1 = 83.46%

FIGURE B.4. Examples of detected and segmented nodules in juxta-pleural cases. The output of our method is in green, the ground truth in red, and their superposition in yellow.

# BIBLIOGRAPHY


[1] Ai T, Yang Z, Hou H, Zhan C, Chen C, Lv W, Tao Q, Sun Z, Xia L - *Correlation of Chest CT and RT-PCR Testing in Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases* - https://doi.org/10.1148/radiol.2020200642 [2020]

[2] Aresta G, Jacobs C, Araújo T, Cunha A, Ramos I, van Ginneken B, Campilho A - *iW-Net: an automatic and minimalistic interactive lung nodule segmentation deep network* - Scientific Reports 9, 1–9 [2019]

[3] Armato SG III, McLennan G, Bidaut L, McNitt-Gray MF, Meyer CR, Reeves AP, Zhao B, Aberle DR, Henschke CI, Hoffman EA, Kazerooni EA, MacMahon H, van Beek EJR, Yankelevitz D, et al. - *The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): A completed reference database of lung nodules on CT scans* - Physics, 38: 915–931 [2011]

[4] Bahdanau, D., Cho, K., Bengio, Y. - *Neural Machine Translation by Jointly Learning to Align and Translate* - CoRR, abs/1409.0473 [2014]

[5] Bao, H., Dong, L., Wei, F. - *BEiT: BERT Pre-Training of Image Transformers* - ArXiv, abs/2106.08254 [2021]

[6] Behrmann, J., Grathwohl, W., Chen, R.T.Q., Duvenaud, D., Jacobsen, J.-H. - *Invertible Residual Networks* - 36th International Conference on Machine Learning, PMLR, pp. 573–582 [2019]

[7] Bibal, A., Lognoul, M., de Streel, A., Frénay, B. - *Legal requirements on explainability in machine learning* - Artif Intell Law 29, 149–169 [2021]

[8] Böttcher, A., Grudsky, S.M. - *Spectral properties of banded Toeplitz matrices* - Society for Industrial and Applied Mathematics (SIAM) [2005]

[9] Bonavita I., Rafael-Palou X., Ceresa M., Ribas V., Piella Gemma., González Ballester M. A. - *Integration of convolutional neural networks for pulmonary nodule malignancy assessment in a lung cancer classification pipeline* - Computer Methods and Programs in Biomedicine, Volume 185, 2020, 105172, ISSN 0169-2607 [2020]

[10] Bossard, L., Guillaumin, M., Van Gool, L. - *Food-101 – Mining Discriminative Components with Random Forests* - Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8694. Springer, Cham. [2014]

[11] Bray F, Ferlay J, Soerjomataram I, Siegel RL, Torre LA, Jemal A - *Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries* - A Cancer Journal for Clinicians 68, 394–424 [2018]

[12] Canziani, A., Paszke, A., Culurciello, E. - *An Analysis of Deep Neural Network Models for Practical Applications* - arXiv.1605.07678 [2017]

[13] Cao H, Liu H, Song E, Hung CC, Ma G, Xu X, Jin R, Lu J - *Dual-branch residual network for lung nodule segmentation* - Applied Soft Computing, Volume 86, ISSN 1568-4946 [2020]

[14] Capobianco, E. - *High-dimensional role of AI and machine learning in cancer research* - Br J Cancer 126, 523–532 [2022]

[15] Chavan, A., Shen, Z., Liu, Zhuang, Liu, Zechun, Cheng, K.-T., Xing, E.P. - *Vision Transformer Slimming: Multi-Dimension Searching in Continuous Optimization Space* - IEEE/CVF Conference on CVPR, pp. 4931–4941 [2022]

[16] Chen, X., Yao, L., Zhang, Y. - *Residual Attention UNet for Automated Multi-Class Segmentation of COVID-19 Chest CT Images* - arXiv:2004.05645 [cs, eess, q-bio] [2020]

[17] Cohen, J.P., Morrison, P., Dao, L. - *COVID-19 Image Data Collection* - arXiv:2003.11597 [cs, eess, q-bio] [2020]

[18] Cuocolo R., Perillo T., De Rosa E., Ugga L., Petretta M. - *Current applications of big data and machine learning in cardiology* - J Geriatr Cardiol. Aug;16(8):601-607. [2019]

[19] Cuiping Bao, Xuehuan Liu, Han Zhang, Yiming Li, Jun Liu - *Coronavirus Disease 2019 (COVID-19) CT Findings: A Systematic Review and Meta-analysis* - Journal of the American College of Radiology, Volume 17, Issue 6, Pages 701-709, [2020]

[20] Dalla Pozza, N., Buffoni, L., Martina, S., Caruso, F. - *Quantum reinforcement learning: the maze problem* - Quantum Machine Intelligence. 4. 10.1007/s42484-022-00068-y [2022]

[21] Dara S., Dhamercherla S., Jadav S.S., Babu C.M., Ahsan M.J. - *Machine Learning in Drug Discovery: A Review* - Artif Intell Rev. 2022;55(3):1947-1999. doi: 10.1007/s10462-021-10058-4 [2021]

[22] Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L. - *ImageNet: A large-scale hierarchical image database* - 2009 IEEE Conference on Computer Vision and Pattern Recognition, 248-255. [2009]

[23] Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., Guo, B. - *CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows* - 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 12114-12124 [2021]

[24] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N. - *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale* - [2021]

[25] Feng, J., Feng, X., Chen, J., Cao, X., Zhang, X., Jiao, L., Yu, T. - *Generative Adversarial Networks Based on Collaborative Learning and Attention Mechanism for Hyperspectral Image Classification* - Remote Sensing 12 [2020]

[26] Fogel, A.L., Kvedar, J.C. - *Artificial intelligence powers digital medicine* - npj Digital Medicine 1, 5 [2018]

[27] Ganea, O., Becigneul, G., Hofmann, T. - *Hyperbolic Neural Networks* - Advances in Neural Information Processing Systems. Curran Associates, Inc. [2018]

[28] Da Fonseca, C.M., Petronilho, J., - *Explicit inverses of some tridiagonal matrices* - Linear Algebra and its Applications 325 7–21 [2001]

[29] Dai, Z., Liu, H., Le, Q.V., Tan, M. - *CoAtNet: Marrying Convolution and Attention for All Data Sizes* - ArXiv, abs/2106.04803 [2021]

[30] Elharrouss, O., Subramanian, N., Al-Maadeed, S. - *An encoder-decoder-based method for COVID-19 lung infection segmentation* - arXiv:2007.00861 [cs, eess] [2020]

[31] Fan D.P., *et al.,* - *Inf-Net: Automatic COVID-19 Lung Infection Segmentation from CT Images* - arXiv:2004.14133 [2020]

[32] Fang, Y., Zhang, H., Xie, J., Lin, M., Ying, L., Pang, P., Ji, W., - *Sensitivity of Chest CT for COVID-19: Comparison to RT-PCR* - Radiology 296, E115–E117 [2020]

[33] Finzi, M., Izmailov, P., Maddox, W., Kirichenko, P., Wilson, A.G. - *Invertible Convolutional Networks* - Workshop on Invertible Neural Nets and Normalizing Flows, International Conference on Machine Learning [2019]

[34] Fu, Z., Tang, N., Chen, Y. *et al.,* - *CT features of COVID-19 patients with two consecutive negative RT-PCR tests after treatment.* - Sci Rep 10, 11548 [2020]

[35] Fukushima, K. - *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position* - Biol. Cybernetics 36, 193–202 [1980]

[36] Gilbert, A.C., Zhang, Yi, Lee, K., Zhang, Yuting, Lee, H. - *Towards Understanding the Invertibility of Convolutional Neural Networks* - arXiv:1705.08664 [2017]

[37] Gomez, A.N., Ren, M., Urtasun, R., Grosse, R.B. - *The Reversible Residual Network: Backpropagation Without Storing Activations* - arXiv:1707.04585 [2017]

[38] Goodfellow I.J., Bengio Y., Courville A. - *Deep Learning* - MIT Press [2016]

[39] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y. - *Generative Adversarial Nets* - NIPS [2014]

[40] Goodfellow, I.J., Shlens, J., Szegedy, C. - *Explaining and Harnessing Adversarial Examples* - arXiv:1412.6572 [2015]

[41] Graham, R.N.J., Baldwin, D.R., Callister, M.E.J., Gleeson, F.V. - *Return of the pulmonary nodule: the radiologist's key role in implementing the 2015 BTS guidelines on the investigation and management of pulmonary nodules* - Br J Radiol 89, 20150776 [2016]

[42] Gray, R.M. - *Toeplitz and Circulant Matrices: A Review* - Foundations and Trends in Communications and Information Theory: Vol. 2: No. 3, pp 155-239 [2006]

[43] Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D. - *A Survey of Methods for Explaining Black Box Models* - ACM Comput. Surv. 51, 93:1-93:42 [2018]

[44] Han, G., Liu, X., Han, F., Santika, I.N.T., Zhao, Y., Zhao, X., Zhou, C. - *The LISS—A Public Database of Common Imaging Signs of Lung Diseases for Computer-Aided Detection and Diagnosis Research and Medical Education* - IEEE Transactions on Biomedical Engineering 62, 648–656 [2015]

[45] Hanin, B. - *Which Neural Net Architectures Give Rise to Exploding and Vanishing Gradients?* - Advances in Neural Information Processing Systems. Curran Associates, Inc. [2018]

[46] Hancock MC, Magnan JF - *Lung nodule segmentation via level set machine learning* - arXiv:1910.03191 [2019]

[47] Harrison X.B., *et al.,* - *Performance of radiologists in differentiating COVID-19 from viral pneumonia on chest CT* - arXiv:2003.13865v15 [2020]

[48] He K, Zhang X, Ren S, Sun J - *Deep Residual Learning for Image Recognition* - 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778 [2015]

[49] Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M.M.A., Yang, Y., Zhou, Y. - *Deep Learning Scaling is Predictable, Empirically* - DeepAI [2017]

[50] Hosny, Ahmed and Parmar, Chintan and Quackenbush, John and Schwartz, Lawrence H and Aerts, Hugo JWL - *Artificial intelligence in radiology* - Nature Reviews Cancer 18,8 500-510 [2018]

[51] Hsu, H.H., Ko, K.H., Chou, Y.C., Wu, Y.C., Chiu, S.H., Chang, C.K., Chang, W.C. - *Performance and reading time of lung nodule identification on multidetector CT with or without an artificial intelligence-powered computer-aided detection system* - Clinical Radiology 76, 626.e23-626.e32 [2021]

[52] Hu, S., Gao, Y., Niu, Z., Jiang, Y., Li, L., Xiao, X., Wang, M., Fang, E.F., Menpes-Smith, W., Xia, J., Ye, H., Yang, G. - *Weakly Supervised Deep Learning for COVID-19 Infection Detection and Classification From CT Images* - IEEE Access 8, 118869–118883 [2020]

[53] Huang, P., Park, S., Yan, R., Lee, J., Chu, L. C., Lin, C. T., Hussien, A., Rathmell, J., Thomas, B., Chen, C., Hales, R., Ettinger, D. S., Brock, M., Hu, P., Fishman, E. K., Gabrielson, E., Lam, S. - *Added Value of Computer-aided CT Image Features for Early Lung Cancer Diagnosis with Small Pulmonary Nodules: A Matched Case-Control Study* - Radiology, 286(1), 286-295 [2018]

[54] Huang, X., Sun, W., Tseng, T.-L. (Bill), Li, C., Qian, W. - *Fast and fully-automated detection and segmentation of pulmonary nodules in thoracic CT scans using deep convolutional neural networks* - Computerized Medical Imaging and Graphics 74, 25–36. [2019]

[55] Hubel, D.H., Wiesel, T.N. - *Receptive fields of single neurones in the cat's striate cortex* - J Physiol 148, 574–591 [1959]

[56] Jacobsen, J.-H., Smeulders, A., Oyallon, E. - *i-RevNet: Deep Invertible Networks* - arXiv:1802.07088 [2018]

[57] Jiang J, Hu Y, Liu C, Halpenny D, Hellmann MD, Deasy JO, Mageras G, Veeraraghavan H - *Multiple Resolution Residually Connected Feature Streams For Automatic Lung Tumor Segmentation From CT Images* - IEEE Transactions on Medical Imaging 38, no.1 :134–44 [2019]

[58] Jumper, J., Evans, R., Pritzel, A. et al. - *Highly accurate protein structure prediction with AlphaFold* - Nature 596, 583–589 [2021]

[59] Karami, M., Schuurmans, D., Sohl-Dickstein, J., Dinh, L., Duckworth, D. - *Invertible Convolutional Flow* - Advances in Neural Information Processing Systems [2019]

[60] Kathleen Walch - *How AI Is Impacting Operations At LinkedIn* - Forbes [2020]

[61] Kazeminia, S., Baur, C., Kuijper, A., van Ginneken, B., Navab, N., Albarqouni, S., Mukhopadhyay, A. - *GANs for Medical Image Analysis* - arXiv:1809.06222 [cs, stat] [2019]

[62] Keetha, N.V., P, S.A.B., Annavarapu, C.S.R. - *U-Det: A Modified U-Net architecture with bidirectional feature network for lung nodule segmentation* - arXiv:2003.09293 [cs, eess, stat] [2020]

[63] Kong, W., Agarwal, P.P. - *Chest Imaging Appearance of COVID-19 Infection* - Radiology: Cardiothoracic Imaging 2, e200028 [2020]

[64] Krizhevsky, A., Sutskever, I., Hinton, G.E. - *ImageNet Classification with Deep Convolutional Neural Networks* - Advances in Neural Information Processing Systems. Curran Associates, Inc. pp. 1097–1105 [2012]

[65] Lassau, N., Ammari, S., Chouzenoux, E. *et al.,* - *Integrating deep learning CT-scan model, biological and clinical variables to predict severity of COVID-19 patients.* - Nat Commun 12, 634 [2021]

[66] LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., Jackel, L. - *Handwritten Digit Recognition with a Back-Propagation Network* - Advances in Neural Information Processing Systems. Morgan-Kaufmann. [1989]

[67] Lee, J., Kim, H.S., Kim, N., Ryu, E.M., Kang, J.W. - *Learning to Detect Cracks on Damaged Concrete Surfaces Using Two-Branched Convolutional Neural Network* - Sensors 2019, 19, 4796. [2019]

[68] Li F., Sone S., Abe H., MacMahon H., Armato S.G. III, Doi K. - *Lung cancers missed at low-dose helical CT screening in a general population: comparison of clinical, histopathologic, and imaging findings* - Radiology. 2002;225(3):673-683. doi:10.1148/radiol.2253011375 [2002]

[69] Liang, T., Liu, Z., Wu, C.C. *et al.,* - *Evolution of CT findings in patients with mild COVID-19 pneumonia.* - Eur Radiol 30, 4865–4873 [2020]

[70] Lin, J., Gan, C., Han, S. - *Temporal Shift Module for Efficient Video Understanding* - ArXiv, abs/1811.08383 [2018]

[71] Lin, M., Chen, Q., Yan, S. - *Network In Network* - arXiv.1312.4400 [2014]

[72] Liu L., Chen X., Petinrin O.O., Zhang W., Rahaman S., Tang Z.R., Wong K.C. - *Machine Learning Protocols in Early Cancer Detection Based on Liquid Biopsy: A Survey* - Life (Basel). Jun 30;11(7):638 [2021]

[73] Liu J, Gong J, Wang L, Sun X, Nie S - *Segmentation Refinement of Small-Size Juxta-Pleural Lung Nodules in CT Scans* - Iranian Journal of Radiology 16 [2019]

[74] Liu, Y., Zhang, Y., Wang, Y., Hou, F., Yuan, J., Tian, J., Zhang, Y., Shi, Z., Fan, J., He, Z. - *A Survey of Visual Transformers* - ArXiv, abs/2111.06091 [2021]

[75] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B. - *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows* - 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 9992-10002 [2021]

[76] Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., Wei, F., Guo, B. - *Swin Transformer V2: Scaling Up Capacity and Resolution* - arXiv:2111.09883 [2022]

[77] Louis Columbus - *Roundup Of Machine Learning Forecasts And Market Estimates* - Forbes [2020]

[78] Low DA, Harms WB, Mutic S, Purdy JA - *A technique for the quantitative evaluation of dose distributions* - Medical Physics 25, 656–661 [1998]

[79] Ma, J., Wang, Y., An, X., Ge, C., Yu, Z., Chen, J., Zhu, Q., Dong, G., He, J., He, Z., Nie, Z., Yang, X. - *Towards Efficient COVID-19 CT Annotation: A Benchmark for Lung and Infection Segmentation* - arXiv:2004.12537 [cs, eess] [2020]

[80] MacMahon, H., Naidich, D.P., Goo, J.M., Lee, K.S., Leung, A.N.C., Mayo, J.R., Mehta, A.C., Ohno, Y., Powell, C.A., Prokop, M., Rubin, G.D., Schaefer-Prokop, C.M., Travis, W.D., Van Schil, P.E., Bankier, A.A. - *Guidelines for Management of Incidental Pulmonary Nodules Detected on CT Images: From the Fleischner Society 2017* - Radiology 284, 228–243 [2017]

[81] Martens, J., Ballard, A., Desjardins, G., Swirszcz, G., Dalibard, V., Sohl-Dickstein, J., Schoenholz, S,. - *Rapid training of deep neural networks without skip connections or normalization layers using Deep Kernel Shaping* - arxiv:2110.01765 [2021]

[82] Martini K, Barth BK, Nguyen-Kim TD, et al. - *Evaluation of Pulmonary Nodules and Infection on Chest CT with Radiation dose Equivalent to Chest Radiography: Prospective Intra-individual Comparison Study to Standard Dose CT* - Eur J Radiol 2016;85:360-5 [2016]

[83] Messay T, Hardie RC, Tuinstra TR - *Segmentation of pulmonary nodules in computed tomography using a regression neural network approach and its application to the Lung Image Database Consortium and Image Database Resource Initiative dataset* - Medical Image Analysis 22, 48–62 [2015]

[84] Milletari F, Navab N, Ahmadi SA - *V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation* - 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, 2016, pp. 565-571 [2018]

[85] Moosavi-Dezfooli, S.-M., Fawzi, A., Frossard, P. - *DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks* - IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, pp. 2574–2582 [2016]

[86] Morozov, S., Andreychenko, A., Blokhin, I., Vladzymyrskyy, A., Gelezhe, P., Gombolevskiy, V., Gonchar, A., Ledikhova, N., Pavlov, N., Chernina, V. - *MosMedData: Chest CT Scans with COVID-19 Related Findings* - v. 1.0 [2020]

[87] Morris, M.X., Rajesh, A., Asaad, M., Hassan, A., Saadoun, R., Butler, C.E. - *Deep Learning Applications in Surgery: Current Uses and Future Directions* - Am Surg 31348221101490 [2022]

[88] Paliwal, A., Gimeno, F., Nair, V., Li, Y., Lubin, M., Kohli, P., Vinyals, O., - *Reinforced Genetic Algorithm Learning for Optimizing Computation Graphs* - International Conference on Learning Representations [2020]

[89] Pezzano G., Ribas Ripoll V., Radeva P. - *CoLe-CNN: Context-learning convolutional neural network with adaptive loss function for lung nodule segmentation* - Comput Methods Programs Biomed. 2021 Jan;198:105792. doi: 10.1016/j.cmpb.2020.105792. Epub 2020 Oct 15. PMID: 33130496. [2021]

[90] Pezzano G., Díaz O., Ribas Ripoll V., Radeva P. - *CoLe-CNN+: Context learning - Convolutional neural network for COVID-19-Ground-Glass-Opacities detection and segmentation* - Comput Biol Med. 2021 Sep;136:104689. doi: 10.1016/j.compbiomed.2021.104689. Epub 2021 Jul 31. PMID: 34364263; PMCID: PMC8324386. [2021]

[91] Pham, H., Dai, Z., Xie, Q., Luong, M.-T., Le, Q.V. - *Meta Pseudo Labels* - arXiv:2003.10580 [2021]

[92] Philipp, G., Song, D., Carbonell, J.G. - *The exploding gradient problem demystified - definition, prevalence, impact, origin, tradeoffs, and solutions* - arXiv:1712.05577 [2018]

[93] Qian, W., Zhao, X., Sun, W., Qi, S., Sun, J., Zhang, B., Yang, Z. - *Fine-grained lung nodule segmentation with pyramid deconvolutional neural network* - Medical Imaging 2019: Computer-Aided Diagnosis. Presented at the Computer-Aided Diagnosis, SPIE, San Diego, United States, p. 136 [2019]

[94] Rafael-Palou X., Aubanell A., Ceresa M., Ribas V., Piella Gemma., González Ballester M. A. - *Detection, growth quantification and malignancy prediction of pulmonary nodules using deep convolutional networks in follow-up CT scans.* - arXiv:2103.14537 [2021]

[95] Rafael-Palou X., Aubanell A., Ceresa M., Ribas V., Piella Gemma., González Ballester M. A. - *Detection, growth quantification and malignancy prediction of pulmonary nodules using deep convolutional networks in follow-up CT scans* - Supplementary material [2021]

[96] Rafael-Palou X., Aubanell A., Ceresa M., Ribas V., Piella Gemma., González Ballester M. A. - *An Uncertainty-aware Hierarchical Probabilistic Network for Early Prediction, Quantification and Segmentation of Pulmonary Tumour Growth* - ArXiv abs/2104.08789 [2021]

[97] Rafael-Palou X., Bonavita I., Aubanell A., Ceresa M., Ribas V., Piella Gemma., González Ballester M. A. - *Re-Identification and growth detection of pulmonary nodules without image registration using 3D siamese neural networks* - Medical Image Analysis, Volume 67, 2021, 101823, ISSN 1361-8415 [2020]

[98] Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Pinto, A.S., Keysers, D., Houlsby, N. - *Scaling Vision with Sparse Mixture of Experts* - arXiv:2106.05974 [2021]

[99] Ronneberger O, Fischer P, Brox T - *U-Net: Convolutional Networks for Biomedical Image Segmentation* - Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham [2015]

[100] Roy R, Chakraborti T, Chowdhury AS - *A deep learning-shape driven level set synergism for pulmonary nodule segmentation* - Pattern Recognition Letters 123, 31–38 [2019]

[101] Simonyan K, Zisserman A - *Very Deep Convolutional Networks for Large-Scale Image Recognition* - arXiv:1409.1556 [2015]

[102] Singla, S., Wallace, E., Feng, S., Feizi, S. - *Understanding Impacts of High-Order Loss Approximations and Features in Deep Learning Interpretation* - International Conference on Machine Learning, PMLR, pp. 5848–5856 [2019]

[103] Shi, F., Wang, J., Shi, J., Wu, Z., Wang, Q., Tang, Z., He, K., Shi, Y., Shen, D. - *Review of Artificial Intelligence Techniques in Imaging Data Acquisition, Segmentation and Diagnosis for COVID-19* - IEEE Reviews in Biomedical Engineering 1–1 [2020]

[104] Shu, H., Wang, J., Chen, H., Li, L., Yang, Y., Wang, Y. - *Adder Attention for Vision Transformer* - Neural Information Processing Systems. [2021]

[105] Simonyan, K., Zisserman, A. - *Very Deep Convolutional Networks for Large-Scale Image Recognition* - arXiv:1409.1556 [2015]

[106] Su, C., Xu, Z., Pathak, J., Wang, F. - *Deep learning in mental health outcome research: a scoping review* - Transl Psychiatry 10, 1–26 [2020]

[107] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A - *Going Deeper with Convolutions* - 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1-9 [2014]

[108] Szegedy C, Ioffe S, Vanhoucke V, Alemi AA - *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning* - Presented at the Thirty-First AAAI Conference on Artificial Intelligence [2017]

[109] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. - *Rethinking the Inception Architecture for Computer Vision* - CVPR, IEEE, Las Vegas, NV, USA, pp. 2818–2826. [2016]

[110] Tang H, Zhang C, Xie X - *NoduleNet: Decoupled False Positive Reductionfor Pulmonary Nodule Detection and Segmentation* - Medical Image Computing and Computer Assisted Intervention – MICCAI 2019. Lecture Notes in Computer Science, vol 11769. Springer, Cham [2019]

[111] Tian, Z., Si, X., Zheng, Y., Chen, Z., Li, X. - *Multi-step medical image segmentation based on reinforcement learning* - J Ambient Intell Human Comput. [2020]

[112] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., J'egou, H. - *Training data-efficient image transformers and distillation through attention* - International Conference on Machine Learning [2020]

[113] Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H. - *Going Deeper With Image Transformers* - IEEE/CVF ICCV, pp. 32–42 [2021]

[114] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I. - *Attention is All you Need* - Advances in Neural Information Processing Systems. Curran Associates, Inc. [2017]

[115] Vorwerk, H., Zink, K., Schiller, R., Budach, V., Böhmer, D., Kampfer, S., Popp, W., Sack, H., Engenhart-Cabillic, R. - *Protection of quality and innovation in radiation oncology: The prospective multicenter trial the German Society of Radiation Oncology (DEGRO-QUIRO study)* - Strahlenther Onkol 190, 433–443 [2014]

[116] Wang, S., Kang, B., Ma, J., Zeng, X., Xiao, M., Guo, J., Cai, M., Yang, J., Li, Y., Meng, X., Xu, B. - *A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19)* - Infectious Diseases [2020]

[117] Wang S, Zhou M, Liu Za, Li Zh, Gu D, Zang Y, Dong D, Gevaert O, Tian J - *Central focused Convolutional Neural Networks: Developing a data-driven model for lung nodule segmentation* - Medical Image Analysis 40, 172–183 [2017]

[118] Wang, G., Liu, X., Li, C., Xu, Z., Ruan, J., Zhu, H., Meng, T., Li, K., Huang, N., Zhang, S. - *A Noise-Robust Framework for Automatic Segmentation of COVID-19 Pneumonia Lesions From CT Images* - IEEE Transactions on Medical Imaging 39, 2653–2663 [2020]

[119] Wu, Y.-H., Gao, S.-H., Mei, J., Xu, J., Fan, D.-P., Zhao, C.-W., Cheng, M.-M. - *An Explainable COVID-19 Diagnosis System by Joint Classification and Segmentation* - arXiv:2004.07054 [2020]

[120] Wu B, Zhou Z, Wang J, Wang Y - *Joint Learning for Pulmonary Nodule Segmentation, Attributes and Malignancy Prediction* - arXiv:1802.03584 [2018]

[121] Xie, J., Zeng, R., Wang, Q., Zhou, Z., Li, P. - *SoT: Delving Deeper into Classification Head for Transformer.* - arXiv:2104.10935 [2021]

[122] Yacoby, Y., Pan, W., Doshi-Velez, F. - *Failure Modes of Variational Autoencoders and Their Effects on Downstream Tasks* - arXiv:2007.07124 [2022]

[123] Yan, Q., Wang, B., Gong, D., Luo, C., Zhao, W., Shen, J., Shi, Q., Jin, S., Zhang, L., You, Z. - *COVID-19 Chest CT Image Segmentation – A Deep Convolutional Neural Network Solution* - arXiv:2004.10987 [cs, eess] [2020]

[124] Zhai, X., Kolesnikov, A., Houlsby, N., Beyer, L. - *Scaling Vision Transformers* - arXiv:2106.04560 [cs.CV] [2022]

[125] Ye, Z., Zhang, Y., Wang, Y., Huang, Z., Song, B. - *Chest CT manifestations of new coronavirus disease 2019 (COVID-19): a pictorial review* - Eur Radiol 30, 4381–4389 [2020]

[126] Yu F., Zhu Y., Qin X., Xin Y., Yang D., Xu T. - *A multi-class COVID-19 segmentation network with pyramid attention and edge loss in CT images* - IET Image Processing. 2021;1–10 [2021]

[127] Yuan, L., Chen, D., Chen, Y.-L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., Liu, C., Liu, M., Liu, Z., Lu, Y., Shi, Y., Wang, L., Wang, J., Xiao, B., Xiao, Z., Yang, J., Zeng, M., Zhou, L., Zhang, P. - *Florence: A New Foundation Model for Computer Vision* - arXiv:2111.11432 [2021]

[128] Zhang, X., Jonassen, I., Goksøyr, A. - *Machine Learning Approaches for Biomarker Discovery Using Gene Expression Data* - Helder I., N. (Ed.), Bioinformatics. Exon Publications, Brisbane (AU). [2021]

[129] Zhang Y, Shi L, Simoff MJ, J Wagner O, Lavin J. - *Biopsy frequency and complications among lung cancer patients in the United States* - Lung Cancer Manag. 2020 Aug 17;9(4):LMT40 [2020]

[130] Zheng, C., Deng, X., Fu, Q., Zhou, Q., Feng, J., Ma, H., Liu, W., Wang, X. - *Deep Learning-based Detection for COVID-19 from Chest CT using Weak Label* - medRxiv 2020.03.12.20027185 [2020]

[131] Zheng, L., Lei, Y. - *A Review of Image Segmentation Methods for Lung Nodule Detection Based on Computed Tomography Images* - MATEC Web Conf. 232, 02001. [2018]

[132] Zhou, T., Canu, S., Ruan, S. - *An automatic COVID-19 CT segmentation network using spatial and channel attention mechanism* - arXiv:2004.06673 [cs, eess] [2020]

[133] Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N., Liang, J. - *UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation* - arXiv:1912.05074 [cs, eess] [2020]