Linköping University | Department of Biomedical Engineering
Master thesis, 30 ECTS | Biomedical engineering
202019 | LIU-IMT-TFK-A–19/560–SE

Managing imbalanced training data by sequential segmentation in machine learning

Susana Bardolet Pettersson

Supervisor : **Anette Karlsson**, IMT, Linköping University
Fredrik Noring, Combitech AB

Examiner : **Magnus Borga**, IMT, Linköping University

LINKÖPING UNIVERSITY

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

Copyright

Abstract

Imbalanced training data is a common problem in machine learning applications. This problem refers to datasets in which the foreground pixels are significantly fewer than the background pixels. By training a machine learning model with imbalanced data, the result is typically a model that classifies all pixels as the background class. A result that indicates no presence of a specific condition when it is actually present is particularly undesired in medical imaging applications. This project proposes a sequential system of two fully convolutional neural networks to tackle the problem. Semantic segmentation of lung nodules in thoracic computed tomography images has been performed to evaluate the performance of the system. The imbalanced data problem is present in the training dataset used in this project, where the average percentage of pixels belonging to the foreground class is 0.0038 %. The sequential system achieved a sensitivity of 83.1 % representing an increase of 34 % compared to the single system. The system only missed 16.83% of the nodules but had a Dice score of 21.6 % due to the detection of multiple false positives. This method shows considerable potential to be a solution to the imbalanced data problem with continued development.

Acknowledgments

I would like to thank my supervisors Anette Karlsson and Fredrik Noring for guiding me through during the thesis work. A special thank to Johnny Larsson for giving me the opportunity to perform this work at Combitech and for providing all required material. I would also like to thank those that wrote their master thesis alongside with me. Lastly, I would like to express my gratitude to my examiner Magnus Borga for helping me find an interesting problem statement for the thesis work.

Linköping, 2019
Susana Bardolet

Contents

1 Introduction

This chapter introduces the major problems covered by this thesis work.

1.1 Background

Machine learning refers to an application of artificial intelligence that allows systems to automatically learn from data and predict outputs without being explicitly programmed [1]. Furthermore, these systems acquire the ability to progressively improve their performance from experience. The implementation of machine learning algorithms is popular in the computer vision field. The aim of this field is to enable a computer to understand and interpret digital images and videos.

In recent years, a new machine learning approach, deep neural networks, has revolutionized the research in the computer vision field. The breakthrough of this new approach is due to the increase of available training data and the development of more powerful hardware [2]. Artificial deep neural networks have made a step forward by adopting the way the human brain works. For a human brain, understanding visual scenes and recognizing objects is an easy task. For a computer, this task is more complicated. The resemblance between the brain and neural networks allows the computer to learn how to analyze the images similarly to the brain.

One application within the computer vision field in which the implementation of deep networks has achieved huge success is semantic segmentation. Semantic segmentation performs partition of objects in images by classifying each pixel of an image to a class. For example, it is used in the medical field to analyze images in order to identify and segment abnormalities, such as tumours.

Although deep learning networks have shown outstanding performance, they have a significant drawback: the requirement of a considerable amount of data. Moreover, this data should be balanced. Imbalanced data refers to the problem in which the classes that a pixel can be classified into are not represented equally. This is a problem in many medical imaging applications where non-healthy pixels are commonly considerably fewer than healthy pixels. The problem arises when training a network with imbalanced data. It results in a network

which learns to classify all pixels as the major class, healthy. The network will make correct predictions for all healthy pixels, which constitute the majority of the medical image. This means that the network will get high accuracy and assume that it performs adequately, when it in fact does not find any of the patholgical pixels. This is considered a weakness in the deep learning field, specifically in medical applications where it is particularly undesired to report a result that indicates no presence of a medical condition, when it truly is present.

The problem behind imbalanced training data arises due to the inverse proportionality characteristic between sensitivity and specificity. A sensitive network is biased to identify the positives cases, i.e., non-healthy pixels. A specific network identifies negative cases, i.e., healthy pixels. The perfect system would have both high sensitivity and specificity. However, it is difficult to acquire both high sensitivity and specificity when using a single network due to the inverse proportionality: when one increases, the other decreases.

This thesis will investigate the possibility to handle imbalanced data by implementing a sequential segmentation system of two networks. The first network will be sensitive. The second network will be specific, to remove the healthy pixels predicted as non-healthy from the segmentation performed by the first network. This approach will neglect the proportionality limitation and allow both high sensitivity and specificity.

1.2 Purpose

The main goal of this thesis is to investigate if a sequential segmentation system can overcome the imbalanced training data problem, considering lung nodule segmentation in thoracic computed tomography images. Is it possible to achieve better performance in terms of sensitivity, specificity and Dice, using this method instead of a one-step approach, i.e., a single network?

1.3 Problem Statements

Following aims were stated for the master thesis:

- Can a sequential system of two fully convolutional neural networks get better performance compared to a single fully convolutional neural network in terms of sensitivity, specificity and Dice when segmenting imbalanced data?
- How does the variability in the data affect the performance of both the sequential and the single system?
- Can the sequential system get results comparable to the radiologists that reviewed the computed tomography images that were used in this thesis?
- On the same dataset, how does the training and inference time differ between the two systems?

1.4 Limitations

The thesis work was conducted over 20 weeks, and therefore required limitations to confine the project to a feasible scope.

- This thesis only considered one deep neural architecture.
- Only one dataset was used which was a collection of thoracic computed tomography images. The number of images used to train and test the two systems was limited due to availability and computing resources.

2 Theory

This chapter contains relevant theory for the thesis. First of all the basics of machine learning and neural networks are explained including the fundamental components of a generic network and its optimization algorithm. Next, diving deeper in the machine learning field, deep learning and convolutional neural networks are described along with their most important features. This part will give the knowledge necessary to understand the implemented system. The evaluation method used to evaluate the performance of the systems is introduced. Lastly, the basics of computed tomography imaging are explained to get an understanding of the type of data this thesis uses to explore the problem statements.

2.1 Machine Learning

Machine learning is a method that automatically detects patterns in data, and then, by using the uncovered patterns, predicts future data or performs decision-making tasks. Predicting the future given some past data always involves some uncertainty and, therefore, machine learning can be seen as a form of applied statistics with a heightened emphasis on the use of computers algorithms to statistically estimate complex functions [2]. Financial services, health care, retail, social media and search engines are examples of approaches that usually have some functionality based on machine learning.

The machine learning type with the widest use is is supervised learning [1]. The goal is to learn a mapping from inputs x to outputs y based on a labelled set of input-output pairs. Each pair consists of an input object x and the desired output value called label, d. The label tells which class x belongs to. The form of the input and output can in principle be anything, an image, a graph, a sentence, etc. A common example is the development of a system that tells if an image contains specific objects. The first step is to collect the system a dataset with numerous images of the objects together with the desired output classes. The supervised learning algorithm analyzes this data and learns which features characterize each class and produces an inferred function that can be used for mapping new unseen instances.

A model, a dataset, an optimization algorithm and a cost function are the main components to build a machine learning algorithm [2]. This section provides the basics of these components and presents a fundamental model in machine learning called artificial neural networks.

2.1.1 Artificial Neural Networks

Artificial neural networks are brain-inspired systems. According to Haykin (2009) [3], neural networks resemble the brain in two aspects: firstly, neural networks acquire knowledge from its environment through a learning process and secondly, the acquired knowledge is stored in so-called synaptic weights which are interneuron connection strengths. Figure 2.1 shows a schematic representation of a neural network neuron, node, and its counterpart in a biological neuron. Each neuron receives signals from other neurons, x, and if the signal is high enough the neuron is triggered and the signal is transferred to the dendrites by synapses, w. Further, the cell body receives the input signal, wx, from the dendrites and produces an output y, which is calculated by using an activation function. After that, the output is sent forward to next neuron through the axon.

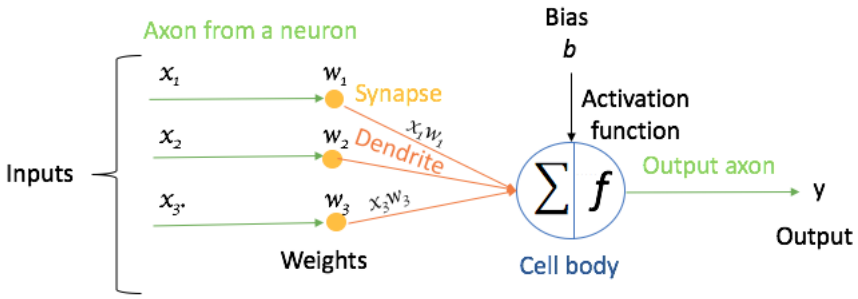


Figure 2.1: Model of a neural network node and its counterpart in a biological neuron where x is the signal received from other neurons, w corresponds to the synaptic weights, wx the input signal to the cell body that applies a function f to produce an output y

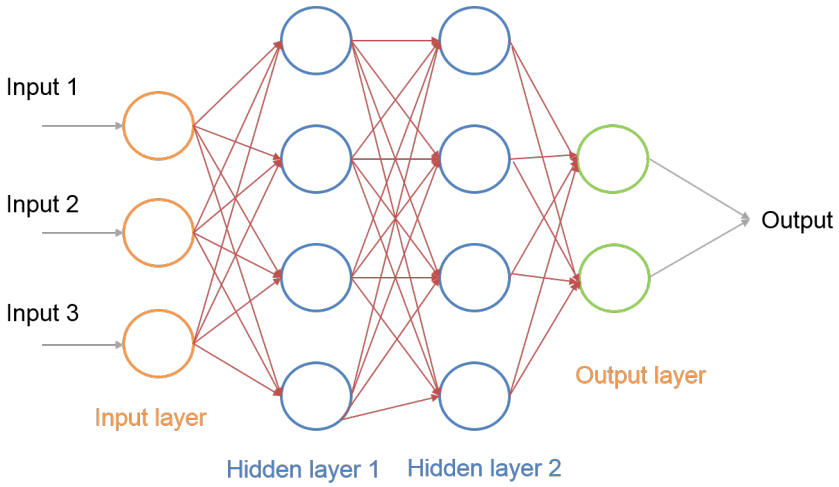


Figure 2.2: Schematic representation of the architecture of a neural network with four layers

A basic architecture of a neural network is represented in Figure 2.2. This neural network consists of four layers that are connected with weights, w. The layers between the input and output layer are the hidden layers. The input layer contains three nodes that are fully connected to the four nodes of the first hidden layer, the nodes of the first hidden layer are fully connected to the nodes of the second hidden layer which in turn are fully connected to the output layer. Fully connection implies every node in a layer is connected to each and

every node in the next layer. The output of a node is multiplied with the weight, w, of the next node. Figure 2.1 illustrates how a single node works. Each node has its own weights and bias associated. Due to the full connection, each node receives several inputs at the time and are added to each other according to

$$z = b + \sum_{i=1}^{n} w_i \times x_i \tag{2.1}$$

As seen in Figure 2.2 and Equation 2.1, each node in a neural network has an additional input called bias, b. The bias is usually represented as an input x = 1 and a weight w_0 (w_0 = b), and allows the activation function to be shifted to left or right in order to handle tasks whose optimal model does not pass through the origin. An activation function is then applied to the sum, z, to ensure stability. In other words, the activation function restricts the sum by keeping it between predetermined limits as, for example, between 0 and 1. The activation function used in this thesis is called Parametric Rectified Linear Unit (PReLU) and is defined by Equation 2.2. This activation function introduces a parameter, α, that allows a non-zero gradient when the node is not active and it is learned along with other network parameters.

$$y(z) = \begin{cases} z, & \text{if } z > 0; \\ \alpha \times z, & \text{otherwise.} \end{cases} \tag{2.2}$$

In order to get a basic understanding of what happens in the layers of a neural network a simple example will be explained next: A network is intended to recognize a specific object in an image. The first layer may analyse the pixel values of the image. The next layer could identify edges based on lines of similar pixel values. Next might recognize shapes and textures and so forth. As deeper layers are reached, the network will have created more complicated structures and patterns detectors in order to recognize more complex features. These architectures that consist of multiple layers with many nodes per layer and are able to represent increasingly complex features are known as deep networks and belong to the class deep learning which will be described in more detailed in section 2.2. [2]

Loss function

In order to improve the performance and get a robust network, it is necessary to penalize the network when it outputs wrong results. This is achieved by introducing the loss function (L) which is the error calculated as the difference between the predicted output and the actual output. There are different loss functions that give different errors for the same prediction which causes different considerable effects on the network performance. The aim during training is to minimize this function since a low loss function value implies good results.

2.1.2 Training an Artificial Neural Network

Before starting the training process some fixed parameters are established as the activation and loss function, the number and type of layers, and the initial weights. Generally, the initial weights of the nodes are initialized with random values calculated according to different initialisation techniques [4].

The training process is actually the learning process which takes the training inputs and desired outputs and updates the network parameters accordingly in order to calculate an output as close as possible to the desired output. This is achieved using a method called backpropagation [5] which consists of two phases: propagation and weight update. The first step is to propagate the training inputs through the network to generate the outputs. At this stage, the output of the randomly initialized network is obtained and at the same time the network has the corresponding desired output in order to calculate the loss function. Now

the machine learning problem becomes an optimization problem with the aim to minimize the loss function. One of the most fundamentals optimization techniques used is the gradient descendent which calculates the derivative of the loss function to optimize the weights [2]. Thereafter, the error is backpropagated from the end to the start to update the weights. Since the weights are updated with small steps, several iterations are necessary for the network to learn and finally converge. Figure 2.3 shows a schematic representation of the training process including all the steps mentioned above.

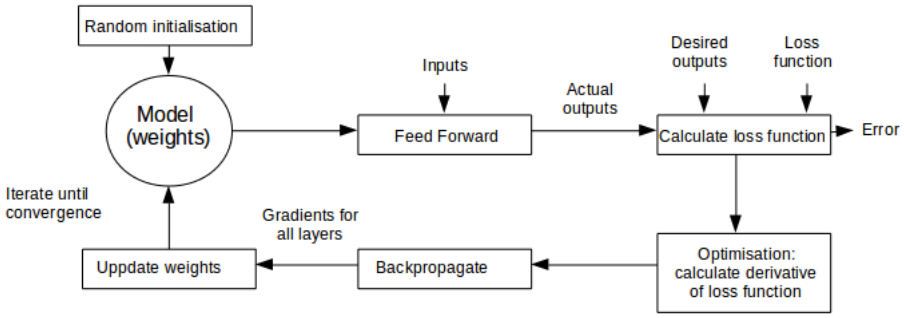


Figure 2.3: Schematic representation of the training process

Optimization algorithm: Stochastic gradient descendent

Gradient descendent is an optimization algorithm that calculates the gradient of the loss function with respect to all weights. The weights are updated accordingly to the gradient with the purpose that the network converges on a local minimum. The weight update step is defined by

$$w_{k+1} = w_k + \Delta w_k \quad \text{where} \quad \Delta w_k = -\eta \frac{\partial L}{\partial w_k} \tag{2.3}$$

where η is the learning rate or step size and $\frac{\partial L}{\partial w_k}$ is the gradient loss for the weight w_k. The learning rate is an hyper-parameter that adjusts how much the weights should be updated with respect to the loss gradient. The establishment of this parameter can be tricky because a small value means that the achievement of convergence will be very slow while a too large value can imply the convergence will never be reached and the system could therefore fail. By giving another look at Equation 2.3 and focus on the gradient loss instead, it can be seen that a negative gradient signifies that the local minimum of the loss function has not been achieved yet and, therefore, by increasing the weight the error will decrease. On the other hand, if the gradient is positive, it means that the local minimum has been passed and, therefore, an increase in the weight will entail an increase of the error. If the gradient is zero, the stable point is reached and no weight update is necessary, see Figure 2.4.

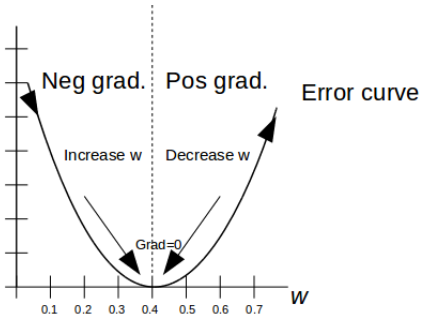


Figure 2.4: A negative gradient requires an increase of the weight to minimize the loss function. A positive gradient requires a decreasing of the weight

Instead of updating the weights after each training input or after the whole dataset, a common method is to calculate the gradient after a subset of inputs, a mini-batch. This method, known as stochastic gradient descent, is much faster and efficient but the result is an approximation of the gradient.

The Adaptive Moment Estimation (ADAM) optimizer [6], is a stochastic gradient descent optimization algorithm that stands out in performance considering memory requirements and computational efficiency. This method calculates individual adaptive learning rates for each parameter in its algorithm. ADAM is suited for applications that are large in terms of parameters and data. It is commonly used in deep learning applications and is used throughout this thesis.

Regularization techniques

The main challenge in machine learning is to develop a network that has the ability to generalize in order to perform well on unseen data and avoid overfitting. Overfitting is a modelling error that occurs when the system performs well on training data but really poor on unseen data. Deep neural networks are more prone to overfitting due to the complexity of the networks [7]. The non-linear hidden layers that constitute a deep neural network can learn highly difficult relationships between inputs and outputs. However, some of these relationships could be the result of sampling noise that exists in the training dataset but not in the test dataset which leads to a bad generalization of the network. Regularisation techniques are then used to reduce this problem by making slight modifications to the learning algorithm. Some of these techniques include weight penalties, dropout [8], soft weight sharing [9] and early stop of the training as soon as a decrease in performance in validation data is detected. In this thesis, only weight penalties are applied to generalize the systems.

Weight decay

The generalization ability of a neural network depends on an equilibrium between the complexity of the network and the information in the training data [10]. In order to decrease the networks complexity, a weight decay [11] can be introduced to limit the growth of the weights and force superfluous weights to zero. This can be achieved by adding a term, λ, to the loss function *L(w)* that penalizes large weights and limits the freedom of the model. Equation 2.4 shows the new loss function with weight decay and Equation 2.5 shows the weight update step using the new loss function.

$$\widetilde{L}(w) = L(w) + \frac{\lambda}{2}w^2 \tag{2.4}$$

$$w_{k+1} = w_k + \Delta w_k \quad \text{where} \quad \Delta w_k = -\eta \frac{\partial L}{\partial w_k} - \eta \lambda w_k \tag{2.5}$$

Batch normalisation

Batch normalisation [12] is a technique that takes a batch and normalizes the data in order to improve the stability and performance of the neural network. It can also be seen as a regularizer. The training of a deep neural network is complicated since each layer's input depend on the parameters in all previous layers. Therefore, small changes in the network parameters are amplified as deeper layers are reached. The network needs to be constantly adapting to new distributions which slow down the training. This phenomenon is known as covariance shift. Covariance shift requires careful tuning regarding learning rate and weight initialisation. Batch normalisation reduces the covariance shift by normalizing by zero mean and unit variance the inputs to each layer. This makes the network less sensitive to the learning rate parameter and weight initialisation. Higher learning rates can therefore be used which reduces training times. For this reason, batch normalisation is primarily seen as an optimization technique and is also implemented in this thesis.

2.2 Deep Learning

Conventional machine learning methods were not able to process data in their raw form and, therefore, they required human engineering and expertise to convert the raw data into valid features from which the machine could detect patterns [13]. These methods performed well on tasks that could be solved by choosing the right set of features to extract for that specific task and providing a simple learning algorithm. However, for many tasks, it is difficult to know which features are necessary to solve the task.

Deep learning methods, instead, allow the machine to automatically learn these features directly from raw data eliminating the engineering by hand. This approach is known as representation learning [2]. The machine not only learns to map the features to outputs but also what features should be extracted. For this reason, deep learning models have achieved huge success in tasks like visual object recognition, object detection, speech recognition and many others [13].

Deep learning architectures are composed of multiple layers that learn multiple levels of features. The machine learns complex features out of simpler features which implies the level of feature abstractness increases with deeper layers. Connecting more layers and making the architecture deeper leads to a machine that can represent more complicated and abstracted features. Nevertheless, it will require more time to learn [2].

2.2.1 Convolutional Neural Networks

A Convolutional Neural Network (CNN) [14] is a simple neural network that uses convolution as mathematical operation instead of general matrix multiplication [2]. Convolutional networks are specialized to process data that has a grid-like topology as images. These networks have achieved tremendous success in image analysis applications.

A schematic representation of the architecture of a convolutional neural network for a binary classification problem is shown in Figure 2.5. It consists of three different layers: convolutional, pooling and fully connected layer. There has to be at least one convolutional layer in the network in order for it to be called a convolutional network [2].

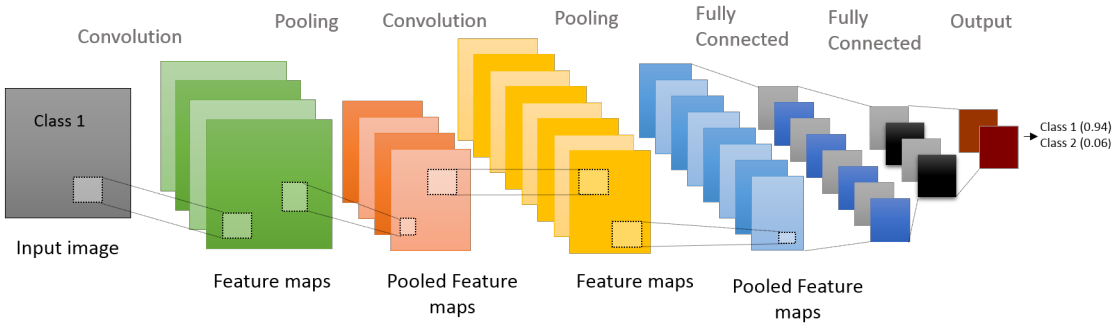


Figure 2.5: Standard architecture of a convolutional neural network

Convolutional layer

In a convolutional layer the weights are arranged as scalars in a kernel. The kernel is then convolved with the input image or a set of feature maps to produce a feature map. Each convolution results in a feature map and each feature map contains features the network has considered important during the learning process.

The success of convolution relies on three peculiarities: sparse interactions, weight sharing and spatial invariance. Sparse interactions mean that the network is not fully connected as in a conventional neural network, i.e., all nodes in a layer are not connected to all nodes in the next layer. Limiting the number of connections of each node implies that fewer parameters need to be stored. This is accomplished by having a kernel smaller than the input. A kernel smaller than the input implies that each weight of the kernel is used at every position of the input as the kernels move through the whole input. Therefore, the weights are shared on different spatial positions meaning that the number of weights needed is reduced.

These two peculiarities reduce memory requirements, improves statistical efficiency and make the convolutional layer harder to overtrain since fewer parameters need to be trained [2]. Furthermore, they introduce a translation invariance property to the layer. This is useful in the case that if a specific object has to be detected, it should not matter if the object is placed in a corner or in the centre.

In a convolutional layer, several convolutional kernels are used. For example, in Figure 2.5, the network takes an image with one channel and outputs four features maps. For this, it is necessary to have four different kernels, one from each input channel to each output channel. Usually, the kernels keep the size of the input image but they can also decrease it. The size is kept by adding zeros around the input before the convolution and after in order to only keep the central part of the output but with the same size as the input. The convolutional layer ends with an activation function.

Pooling layer

Pooling layers are used to downsample the outputs of the convolutions. This is accomplished by applying a pooling function that substitutes the output with a summary statistic of the nearby outputs. Nowadays the standard pooling function is max pooling [15] which replaces the convolutions output with the highest value within the output. For example, applying a max pooling layer with kernel size 2×2 and a stride of 2 to a convolution output of size 4×4 results in an output of size 2×2, i.e., a stride of 2 divides the width and height dimensions of the output by 2.

Fully Connected layer

Fully connected layers are generally placed at the end of a convolutional neural network. These layers are exactly the same layers as in a conventional neural network, i.e., each node in the previous layer is connected to each node in the fully connected layer.

The very last layer of the network, known as classification layer, computes the class scores. A softmax function is commonly used as the activation function of the nodes in this layer. The softmax function maps the non-normalized output of the last fully connected layer to a probability distribution over predicted output labels. Suppose that z is a vector that contains the sum of each node of the classification layer, as shown in Equation 2.1. Softmax performs the computation seen in Equation 2.6 to predict the output probabilities $\hat{y}_i$ where $|\mathbb{V}|$ is the number of classes. [2]

$$\hat{y}_i(\mathbf{z}) = \frac{e^{z_i}}{\sum_{i'=1}^{|\mathbb{V}|} e^{z_{i'}}} \tag{2.6}$$

After applying softmax each $\hat{y}_i$ is in the interval [0,1], and $\sum_i \hat{y}_i = 1$.

Receptive field The term receptive field [16] is the amount of nodes from the previous layer that affect a specific node in the current layer. As mentioned in previous sections, every single node is the result of a convolution performed in the previous layer. As this process is repeated over many layers, the amount of nodes that affect the next node is increased, see Figure 2.6. The entire networks receptive field corresponds to the receptive field of the classification layer node since it makes the prediction.

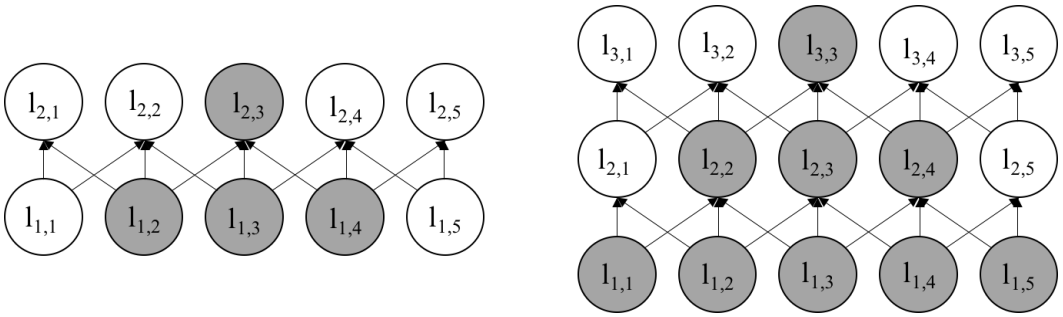


Figure 2.6: Schematic representation of the receptive field of a specific node. Left: The highlighted nodes are the units that affect the node $l_{2,3}$. Right: The deeper the layer the bigger is its respective receptive field. The highlighted nodes represent the nodes that affect node $l_{3,3}$

2.2.2 3D Semantic Segmentation

Convolutional neural networks are typically used on classification tasks, where only one class is predicted for the whole input image. However, in many visual applications, the output should also include localization, i.e., a class label is predicted to each pixel of the image. Classification needs to understand the context, what is in the input image. Segmentation not only needs to understand what is in the input image but also where.

In many medical imaging applications, data consists of 3D volumes commonly represented as stacks of 2D images. Segmentation can be performed directly on the 2D slices and merge the results afterwards. However, this approach ignores the spatial inter-slice correlation [17]. Running segmentation on 3D volumes solves that problem and predicts a class to each voxel, i.e., a pixel in 3D.

One constraint when using 3D images in combination with deep networks is that the entire volume cannot be used as input to the network due to Graphics Processing Unit (GPU) memory limitations. For this reason, the entire volume is split into sub-volumes called image-segments. Bigger segments increase performance since more accurate representation of the entire data is kept.

2.3 Evaluation

When the training has converged and the model is finished, the evaluation of the system is performed. There are many different quantitative measures to evaluate performance of voxel-label semantic segmentation algorithms [18]. Most of these quantitative measures can be derived from the four basic cardinalities, namely true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN), of the confusion matrix, see Figure 2.7.

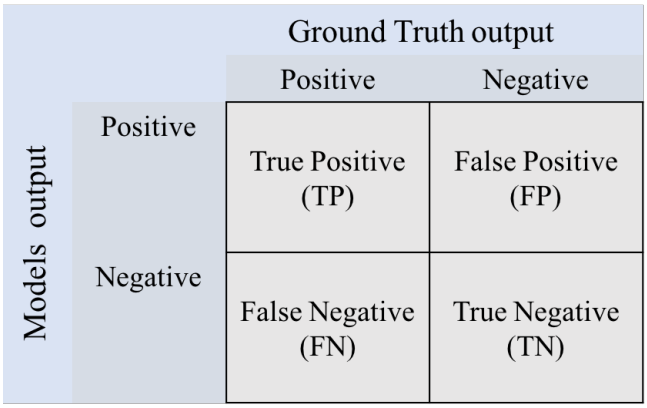


Figure 2.7: Confusion matrix and the four cardinalities

In order to answer the problem statements of this thesis, four quantitative measures will be calculated: sensitivity, specificity, Dice and F_2-score.

Sensitivity, also called recall, measures how well the model identifies positive cases. It is the number of true positives (TP), i.e., the number of nodule voxels correctly classified as nodule, upon the total number of nodule voxels observed, i.e., the addition of true positives (TP) and false negatives (FN), i.e,

$$Sensitivity = \frac{TP}{TP + FN} \tag{2.7}$$

Specificity measures how well the model identifies negative cases. It is the number of true negatives (TN), i.e., the number of non-nodule voxels correctly classified as non-nodule, upon the total number of non-nodule voxels observed, i.e., the addition of false positives(FP) and true negatives (TN), i.e,

$$Specificity = \frac{TN}{FP + TN} \tag{2.8}$$

Generally, accuracy is used to measure how well the model performs. It is a measurement of correctness calculated according to Equation 2.9. However, this measure can be misleading since it does not take into account the mislabelled voxels. This a significant problem in imbalanced data since the model will get high accuracy even though it classifies wrong for the minority class.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.9}$$

There are other accuracy-measures that are affected by mislabeling voxels, such as Dice score. The Dice score is calculated according to Equation 2.10 which considers both sensitivity and precision. Precision measures how well the model identifies positive cases among all retrieved cases. It is the number of true positives (TP) upon the addition of true positives (TP) and false positives (FP), see Equation 2.11.

$$Dice = 2 \times \frac{precision \times sensitivity}{precision + sensitivity} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{2.10}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.11}$$

F_2-score is a similar accuracy-measure as Dice with the difference that the false negatives are more weighted than false positives and, therefore, is more relevant in medical applications. It is calculated according to

$$F_2 = \frac{5 \times TP}{5 \times TP + 4 \times FN + FP} \tag{2.12}$$

2.4 Computed Tomography

Computed Tomography (CT) is an image generation technique based on radiation, particularly x-rays, used to create detailed images of internal parts of the body. Computed tomography consists of a motorized x-ray tube that shoots beams of x-rays as it rotates around the patient. An arc-shaped detector is located directly opposite the source and rotates at the same time. The x-rays which pass through the patient, are detected by the detector and transmitted to a computer for image reconstruction.

2.4.1 Basic Principle

The principle of computed tomography is to measure the spatial distribution of a physical quantity called attenuation.

Attenuation is defined as the natural logarithm of the ratio of initial intensity, I_0, to attenuated intensity, I, see Equation 2.13. In the simplest case, i.e., a homogeneous object with monochromatic radiation, the attenuated intensity is given by Equation 2.14 where μ is the linear attenuation coefficient and d the absorber thickness. It can be remarked that the intensity decreases exponentially with absorber thickness. By combining Equation 2.13 and 2.14, the total attenuation is given as the product between the linear attenuation coefficient and the absorber thickness.

$$Attenuation = ln\frac{I_0}{I} \tag{2.13}$$

$$I = I_0 \times e^{-\mu \times d} \tag{2.14}$$

However, the human body is not homogeneous and the total attenuation depends on the local value of the linear attenuation coefficient for each ray path interval, i.e., each structure of the body. This can be expressed as the integral over the local linear attenuation coefficients along the ray path, see Equation 2.15, and the total attenuation can be calculated as Equation 2.16

$$I = I_0 \times e^{-\int_0^d \mu \times ds} \tag{2.15}$$

$$Attenuation = ln\frac{I_0}{I} = \sum \mu_i \times d_i \tag{2.16}$$

Lastly, computed tomography scanner uses polychromatic x-rays, rays with different energies, and this factor has to be taken into account since linear attenuation coefficient

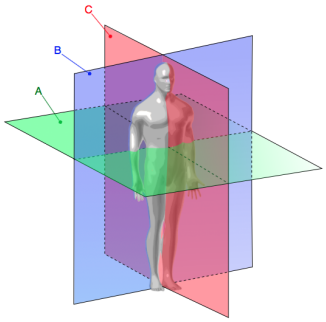
may depend strongly on energy, E. This effect is added in Equation 2.17 which shows the mathematical expression used in CT measurements. [19]

$$I = \int_0^{E_{max}} I_0(E) \times e^{-\int_0^d \mu(E) \times ds} \times dE \tag{2.17}$$

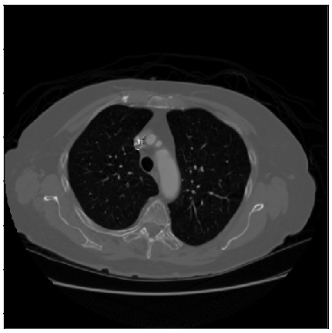
2.4.2 Computed Tomography Images

As mentioned before, the x-ray tube and the detector rotates around the patient. In one rotation, for each angular position of the source, an attenuation profile, also known as a projection, is obtained. This profile is a set of projection values. Each time the source completes one full rotation a 2D image slice of the patient is constructed by using a sophisticated algebraic reconstruction technique which analyzes all projections and assigns a numerical value to each pixel of the slice. This value is the average of all the attenuation values contained within the corresponding pixel [19]. The patient is then moved, usually 1-10 mm, and the process is repeated to produce another image slice. The image slices can either be displayed individually or stacked together as a 3D image of the patient. Figure 2.8b shows an individual image slice. The advantage of acquiring 3D images is the ability to reconstruct images in three different plans: coronal, axial and sagittal, see Figure 2.8a. It is helpful to view the anatomy in all three planes when evaluating the extent of a disease in a patient.

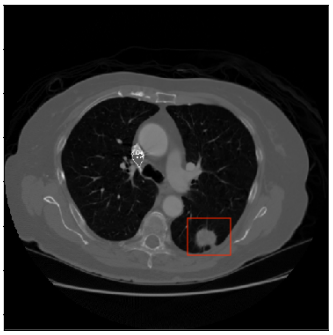
However, the attenuation coefficient is not very descriptive and due to energy dependency it is difficult to compare images obtained with scanners of different voltages. For this reason, CT values are specified in Hounsfield units.



(a) Anatomy planes: (A) Axial, (B) coronal, (C) sagittal. Figure from [20].



(b) CT axial slice.



(c) CT with a marked nodule.

Figure 2.8: Example of anatomy planes, an axial CT image and an axial CT image with the presence of a lung nodule

Hounsfield units

The Hounsfield Unit (HU) is a quantitative value for describing radiodensity in CT images. It is a linear transformation of the linear attenuation coefficient into a scale of arbitrary units in which water has value 0 HU and air -1000 HU. It is calculated by

$$HU = \frac{\mu - \mu_{water}}{\mu_{water} - \mu_{air}} \times 1000 \tag{2.18}$$

The Hounsfield scale has usually a range from -1024 HU to 3071 HU for medical scanners [19]. Most of the body areas present positive HU units, exceptions are lung tissue and fat which present negative values due to their low attenuation and density ($\mu_{water} > \mu_{lung}$).

The range of Hounsfield scale contains 4096 gray levels but humans can only distinguish

a maximum of 80 gray levels [19]. In order to allow the observer to interpret the images, a limited number of HU are displayed. This is achieved by defining a window, an interval of interest, to represent the complete gray scale. The centre of the window corresponds approximately to the mean of the HU unit of the structure of interest and the window width defines the image contrast. For example, a narrow window is chosen when differences in attenuation of the structures to be differentiated are really small as in the brain while a wide window is used for large differences as the lungs and skeleton. This results in a change of the appearance of the image to highlight particular structures.

2.4.3 Lung Nodules

Lung nodules, also known as coin lesions, are lung tissue abnormalities. Their form is overall round or oval-shaped with a diameter that can vary from 3 to 30 mm, see Figure 2.8c. Although lung cancer always manifests lung nodules, not all lung nodules are cancerous. Actually, most lung nodules are benign and are the results of scars or inflammations from any type of lung infection. Despite the fact that most nodules are benign, there is a big challenge in developing systems that find and segment nodules since it is a relevant way to diagnose lung cancer. [21]

3 Related Work

Several articles and projects within the field have been used as inspiration to the development of this thesis work. This chapter briefly presents the previous work done in the field of semantic segmentation along to the most known networks. It also describes the original architecture of the network that this thesis is based on and its novel contributions and history. Furthermore, it gives an overview of the different works done to attempt to solve the imbalanced training data problem.

3.1 Semantic Segmentation

Deep learning has rapidly become a methodology of choice for semantic segmentation. The breakthrough came when Long et al. first introduced fully convolutional neural networks in [22]. A Fully Convolutional Neural Network (FCN) is a conventional convolutional neural network where the last fully connected layers are replaced by convolutional layers in order to output spatial maps instead of class probabilities, see Figure 3.1. In [22], powerful existing convolutional network models (AlexNet [23], VGG [24], GoogLeNet [25], ResNet [26]) were transformed into fully convolutional networks in order to make dense predictions, i.e., predict a label for each voxel. The key insight is to keep the ability of convolutional neural networks to learn hierarchies of features and refine the spatial information. In other words, fully convolutional networks combine what and where.

The removal of fully connected layers allows the network to handle inputs of arbitrary size. Moreover, the number of weight parameters is reduced and, consequently, the training time and computational cost. It can be mentioned that several studies as [27] have shown that the number of parameters in a network can be decreased and still maintain the same performance.

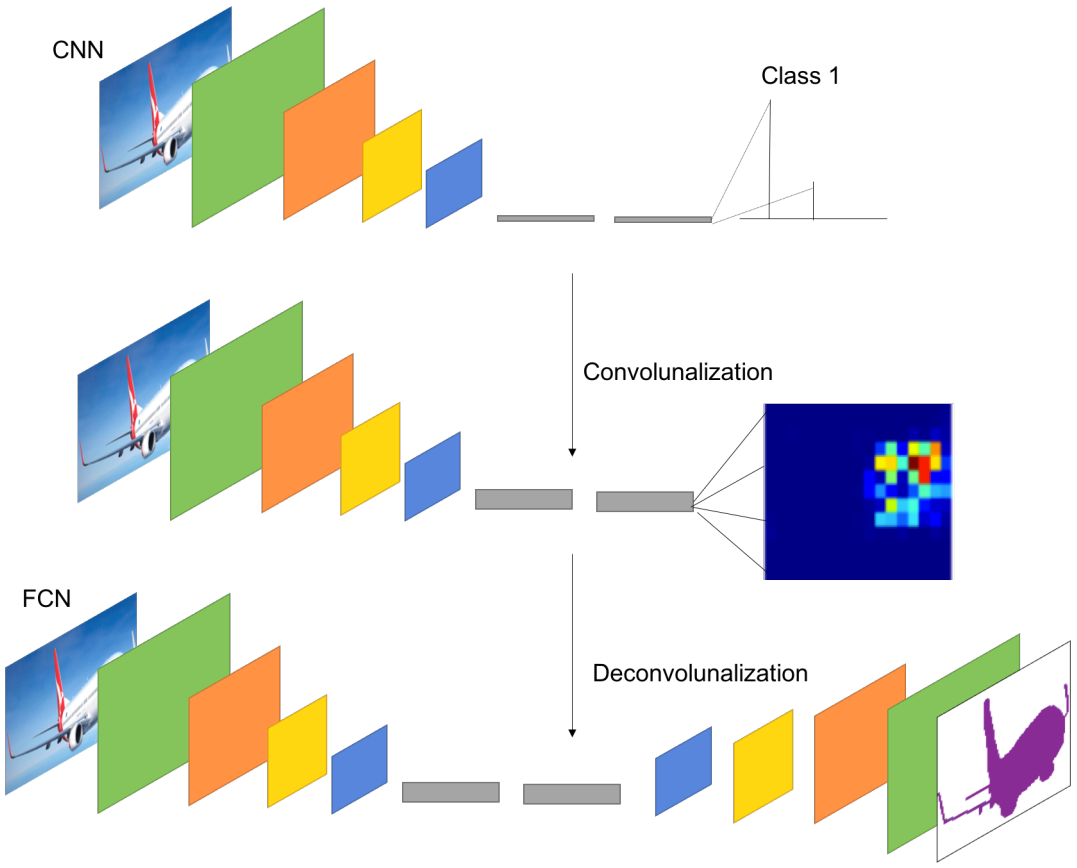


Figure 3.1: Fully Convolutional Network. The top image shows a classifier, CNN, that is next transformed to an FCN by replacing fully connected layers with convolution layers as seen in the middle image. The middle image shows a network that produces spatial heatmaps and by including a deconvolution layer for upsampling, dense predictions can be performed. Figure inspired by Long et al. [22]

Nowadays, the most successful deep learning techniques for semantic segmentation stem from Long et al. research. Other variants to the FCN presented by Long et al. but with similar architecture are [28], [29], [30], [31]. They all present an encoder which produces feature maps or low-resolution image representations and a decoder which maps those low-resolution images to pixel-wise predictions. In general terms, the encoder stage is a suitable CNN whose fully connected layers have been removed. The encoder in [28], [22], [31] has the same architecture as the convolution part of the VGG net [24]. The VGG net is a very deep network of 16-19 weight layers with very small (3×3) convolution filters. Usually, the differences between the architectures lie on how the upsampling and pixel-wise classification is performed, i.e., on the decoder. For example, SegNet [28] uses unpooling to upsample the feature maps in the decoder. This network presents a symmetrical architecture and each decoder has its corresponding encoder. Of these, during max-pooling in the encoder, the indices of the pixels locations are stored and passed to the decoder. The decoder, by using the stored max-pooling indices, upsamples the feature maps. This means that SegNet does not learn the upsampling whereas FCN based architectures use learnable deconvolutions initialized with bilinear interpolation filters to upsample the input feature maps.

3.1.1 Semantic Segmentation in Medical Applications

Segmentation tasks in medical imaging applications are extremely relevant. Therefore, after the success of methods based on FCN and CNN for segmentation tasks of natural images,

likewise methods were developed for medical imaging analysis [32], [33], [34], [35], [36], [37].

In both [32] and [36] an architecture made of two convolutional pathways is used to perform brain lesion segmentation. The motivation is to get both local and larger contextual information when segmenting a voxel. The way they achieved this purpose differs between the two pieces of research. In [32], one pathway has smaller (7×7) receptive field compared to the other (13×13). In [36], the inputs of the two pathways are centred at the same image but one of them is extracted from a downsampled, i.e., lower resolution, version of the image.

The two-step approach that this thesis implements is inspired by several articles as [35], and particularly from [37] and [38]. The work in [37] presents a two-step approach to segment lesions in the liver from CT images. A first network is trained to find the region of interest of the liver which is further sent to the second network to segment lesions within the liver. This is motivated by the fact that smaller input regions entail to more accurate segmentation. In addition, a preprocessing and a postprocessing step are also implemented. As a preprocessing step in order to exclude irrelevant organs and objects, the Hounsfield unit values of the CT that belonged to the liver were windowed and, thereafter, contrasted by histogram equalization. This facilitates the first network to segment the liver. The postprocessing step is performed by implementing a 3D dense conditional random fields CRFs [39] to achieve higher segmentation accuracy. Similarly, [38] presents a sequence of two networks where the first network outputs a predicted segmentation mask. The mask is then used to shrink the input of the second network and get rid of the unnecessary background. The main differences compared to this thesis are the preprocessing step which only normalizes the data, there is no postprocessing step and the networks present the same architecture. The reason is to be able to answer the problem statement since the implementation of other processing steps or use of different architectures will not allow knowing the reason the sequential system worked better or worse.

3.1.2 U-Net

This thesis is based on an architecture called *3D U-Net* [34]. It is a fully convolutional neural network developed to perform dense volumetric segmentations. The network is based on the previous U-Net architecture [33] from Ronneberger et al.

The 3D U-Net replaces all 2D operations from the previous U-Net with their 3D counterparts, reduces the number of downsampling blocks from four to three, reducing therefore the number of convolution layers from twenty-three to eighteen and applies batch normalisation before each activation. Its architecture is shown in Figure 3.2.

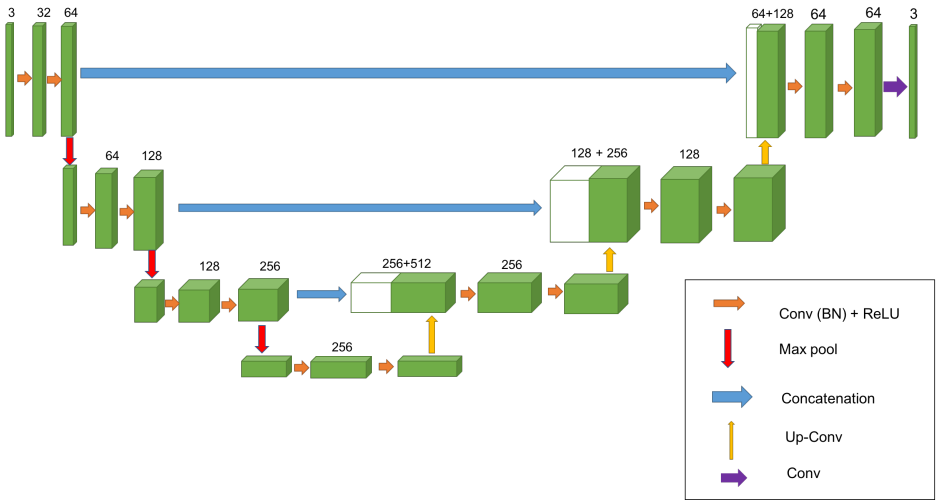


Figure 3.2: 3D U-Net architecture. The feature maps are represented as blue boxes. Above the boxes is denoted the number of feature maps. Figure inspired by Çiçek et al. [34]

The left side represents the contracting path or encoder and the right side the expansive path or decoder. The encoder consists of the application of two $3 \times 3 \times 3$ convolutions, each followed by a rectified linear unit and a $2 \times 2 \times 2$ max pooling operation with a stride of two in each dimension for downsampling. The number of feature maps is doubled after each downsampling step. The decoder consists of an upsampling of the feature map followed by an upconvolution of $2 \times 2 \times 2$ by strides of two in each dimension which halves the number of feature maps, a concatenation with the equivalently feature map from the encoder (known as skip connections), and two $3 \times 3 \times 3$ convolutions followed by ReLU. In the end, a $1 \times 1 \times 1$ convolution layer is applied to map the 64-component feature vector to the number of desired classes. Due to the downsampling blocks of the network, there is a constraint regarding the input size defined by

$$\text{Input size} = 92 + M \times 8, \text{where } M \geqslant 0 \tag{3.1}$$

Deep networks with convolutions of 3D kernels overwhelm the computational cost due to the big amount of learnable parameters in the network. Therefore, when dealing with 3D data all convolutional kernels should be small in order to preserve computational speed and memory usage. By using the smallest kernel size ($3 \times 3 \times 3$), the U-Net architecture has 1,906,995 parameters in total.

Furthermore, for these types of networks it is extremely important to perform a good initialisation of the weights. The reason is that, otherwise, parts of the network will never contribute, while others may give excessive activations. For the U-Net architecture, the initial weights are initialized from a Gaussian distribution.

The choice of using 3D U-Net is motivated by its outstanding performance on very different biomedical segmentation applications. The availability of widely developed documentation, the novel contributions to the field of deep learning and the implementation in several works proves the significance of this network.

3.2 Imbalanced Training Data

One of the main challenges in using fully convolutional networks is when the training data is imbalanced, which is frequent in many medical imaging applications. A clear example is the segmentation of lung nodules where the number of nodule voxels is much lower than healthy

voxels. When training a network with imbalanced data results in a network extremely biased towards the non-nodule class. This is particularly undesired in medical applications since false negatives are more important than false positive.

It is difficult to teach a machine to recognize something when it hardly ever sees it. For this reason, several methods have been proposed to address this problem and they can be divided into two main categories: data level and algorithmic level. Methods that combine the two levels are also available. Data level methods operate on the training data and change its class distribution by performing oversampling [40] or undersampling [41]. The other category keeps the training set unchanged but adjusts the training algorithm by using class experts [42], two-step training [32] or different types of loss functions as weighted [33], similarity [43], [44], and asymmetric similarity [45], [46].

Oversampling and undersampling are methods that result in having an equal number of samples of each class. Oversampling replicates randomly samples that belong to the minority class. This method, however, may lead to overfitting [40]. Undersampling, as opposed to oversampling, removes random samples from the majority class. This method presents several drawbacks as the removal of data that may contain important information and the reduction of data available.

Another method called Class Expert Generative Adversarial Network (CE-GAN) has been proposed in [42] as the solution for the imbalanced data problem. Class Experts (CE) uses layers that have been pretrained to recognize the features of a single class. The Generative Adversarial Network (GAN) is the algorithm used to pretrain the layers. Each layer is trained with only a single class and the GAN algorithm is beneficial since it is able to determine whether an input data is from the assigned class or not due to the use of a discriminative model in the process.

Imbalanced data can also be handled by implementing a new form of training. A two-phase training is presented in [32]. The first phase consists of training the network with patches that contain all classes equally. During the second phase, the output layer is re-trained with the imbalanced data in order to get a more representative distribution of the classes.

During recent years many studies have derived more robust and appropriate loss functions in order to tackle imbalanced data. The loss functions that have presented a big potential to address this problem are named below. All the losses are explained for binary classification, i.e., foreground and background. Let N be the number of image elements, i.e., voxels, r_n the referenced foreground voxels, p_n the predicted foreground voxels and for the background class $1 - r_n$ and $1 - p_n$ respectively.

Weighted cross-entropy (WCE) The weighted cross-entropy was introduced in [33] in order to reduce weights for the frequently seen background class and increase weights for the foreground class. It can be expressed as

$$WCE = -\frac{1}{N}\sum_{n=1}^{N} wr_n log(p_n) + (1 - r_n) log(1 - p_n), \tag{3.2}$$

where $w = \frac{N - \sum_n p_n}{\sum_n p_n}$ is the weight assigned to the foreground class.

Dice Loss (DL) Milletari et al. proposed in [43] a loss function based on Dice score coefficient which is a measure of overlap to assess segmentation performance. It can be

expressed

$$DL = 1 - \frac{\sum_{n=1}^{N} p_n r_n + \epsilon}{\sum_{n=1}^{N} p_n + r_n + \epsilon} - \frac{\sum_{n=1}^{N} (1 - p_n)(1 - r_n) + \epsilon}{\sum_{n=1}^{N} 2 - p_n - r_n + \epsilon} = 1 - \frac{2 \times TP}{2 \times TP + FP + FN} \tag{3.3}$$

The Dice score is the harmonic mean of precision and recall since it weighs false positives and false negatives equally. For this reason, this loss forms a symmetric similarity loss function.

Generalized Dice Loss (GDL) The Generalized Dice Loss is based on the Generalized Dice Score (GDS) and it was proposed in [44] as a loss function. It is a weighted loss where the contribution of each class, label, is corrected by the inverse of its square volume. It can be expressed as

$$GDL = 1 - 2\frac{\sum_{l=1}^{2} w_l \sum_n r_{ln} p_{ln}}{\sum_{l=1}^{2} w_l \sum_n r_{ln} + p_{ln}}, \text{where l denotes the class and } w_l = \frac{1}{(\sum_{n=1}^{N} r_{ln})^2} \tag{3.4}$$

Tversky loss function (TL) The Tversky loss function is based on the Tversky index. It is an asymmetric similarity loss function since it weighs false negatives and false positives unequally by multiplying them with different constants. Different approaches based on Tversky index have been developed in which the difference relies on how the weights are distributed.

The Tversky loss function proposed in [46] is expressed in equation 3.5. The best results were given when $\alpha = 0.3$ and $\beta = 0.7$. It is worthy to mention that in the case of $\alpha = \beta = 0.5$ this loss simplifies to be the Dice loss.

$$\begin{aligned} TL &= 1 - \frac{\sum_{n=1}^{N} p_n r_n}{\sum_{n=1}^{N} p_n r_n + \alpha \sum_{n=1}^{N} p_n (1 - r_n) + \beta \sum_{n=1}^{N} (1 - p_n)(1 - r_n)} \\ &= 1 - \frac{TP}{TP + \alpha FP + \beta FN} \end{aligned} \tag{3.5}$$

4 Method

This chapter describes the implementation of the two systems explored in this thesis. A description of the data is given, as well as its origin and the preprocessing steps performed to make it usable. This is followed by the motivations of the architecture of the two implemented systems.

4.1 Data

The data used as input to the implemented networks were thoracic computed tomography images. These images were in DICOM format and belonged to the Lung Image Database Consortium and Image Database Resource Initiative (LIDC-IDRI) database, which is accessible for public download from The Cancer Imaging Archive [47]. This database contains 1018 helical thoracic CT images with lung nodules of different sizes and shapes.

These CT images have been reviewed independently by four thoracic radiologists in a two-phase reading process. During the first phase, the radiologists were asked to detect nodules and mark them as (1) nodule equal to or greater than 3 mm, (2) nodule smaller than 3 mm or (3) non-nodule greater than 3 mm. For the first group of nodules, the radiologists had to draw a complete outline around the nodule. The outline is an outer border, meaning that pixels belonging to the nodule do not overlap with the outline.

The four radiologists read the same cases and did the annotations independently. The results of the first phase were compiled and sent back to the readers for the second part of the process. In the second phase, the radiologists read the cases independently again with the benefit that they could see the markings from the other three radiologists and their own markings. They then made a final decision about the marking of each case. The four radiologists did not agree on the classification and shape of all lung nodules. The annotations of the four radiologists were saved in XML-files, and they constituted the ground truth images for this project.

The nodules with a size greater than 3 mm have a higher probability of being cancerous and are therefore of higher relevance clinically. Furthermore, the non-nodules are other pulmonary lesions that do not possess malignancy [47]. Hence, this thesis only focused on nodules equal to or greater than 3 mm, and considered the rest as non-malignant nodules.

There were 2669 lesions marked by at least one radiologist as a nodule equal to or greater than 3 mm while only 928 (34.8 %) of these nodules were marked by all four radiologists [48].

4.1.1 Preprocessing of Images

Deep learning algorithms require large amounts of data in order to develop a generalized model. However, the quality of the data also affects the performance of the network. The LIDC-IDRI database has been created by the collaboration of seven academic centers and eight medical imaging companies [47]. This means that the images differ in terms of image size, voxel dimension, data type, modality and manufacture, see table 4.1.

CT images	
Width and Height	512 pixels
Number of slices	[80-625]
Pixel spacing	[0.48828125-0.9765625]\[0.48828125-0.9765625] mm
Slice thickness	1,1.25,2,2.5,5 mm
Data type	int16, uint16, uint32
Manufacturer	GE Medical Systems, Toshiba, Siemens, Philips
Modality	CT, DX, CR

Table 4.1: Characteristics of the computed tomography images of the LIDC-IDRI database.

Here follows an explanation of the preprocessing steps performed in this project in order to normalize the data. The algorithms used for collecting, preprocessing, visualising the data and the creation of the ground truth were implemented in python v2.7.

- Step 1: Axial CT modality. The lung nodule outline is only seen in the axial CT modality.
- Step 2: Int16 as data type. All images were converted to the data type int16 since it was the most common data type within the dataset. Additionally, the Hounsfield scale comprises negative units, and, therefore, a data type that allows negative values was required.
- Step 3: Normalization of voxel values. All voxel values were converted to Hounsfield units, according to Equation 4.1. As mentioned in section 2.4.2, each voxel value represents the attenuation coefficient (IV) of the corresponding tissue. The rescale intercept (I) and the rescale slope (S) were extracted from the metadata of the images.

$$HU = IV \times S + I, \tag{4.1}$$

The scan field of a CT scanner is a cylinder and, therefore, the most suitable geometry to scan is a cylinder. However, the output is squared and the pixels outside of the cylinder boundaries are handled differently depending on the manufacturer. These pixel values had to be changed before the conversion to the Hounsfield units in order to correspond to air, according to the Hounsfield scale.

- Step 4: Removal of artefacts. Artefacts degrade the quality of CT images. Due to time limitations, the artefacts were removed by setting all the pixels values above 1900 HU to soft tissue. Bone is the body structure with the highest HU value, 1800-1900 HU [49]. For this reason, in this project, all HU values above 1900 were considered artefacts.

– Step 5: Normalization of voxel dimension. All images were resampled to the dimension of the image with highest resolution, i.e., $0.48828 \times 0.48828 \times 1$ mm. The highest resolution was chosen in order to keep all the information. This signified a change in the width and height dimensions, and was different for all images.

– Step 6: Conversion to NIfTI format. Due to network requirements, the input data was required to be in NIfTI format and not in DICOM.

– Step 7: Creation of the ground truth data. A ground truth volume for each CT image was created by reading the corresponding XML-file. For this thesis project, two ground truth datasets were required to implement the sequential system which is further explained in section 4.2.3. In one dataset, all nodules that were annotated by at least one radiologist were labelled. This corresponded to an agreement level of 25 %. In the second dataset, only the nodules that were annotated by at least three radiologists were labelled. This dataset constituted the essential ground truth dataset of the project, i.e., the dataset used to evaluate the performance of the two systems. This was motivated by the fact that if at least three radiologists agreed regarding a nodule, it was highly probable that it truly was a nodule. Figure 4.1 shows an example of the two different datasets.

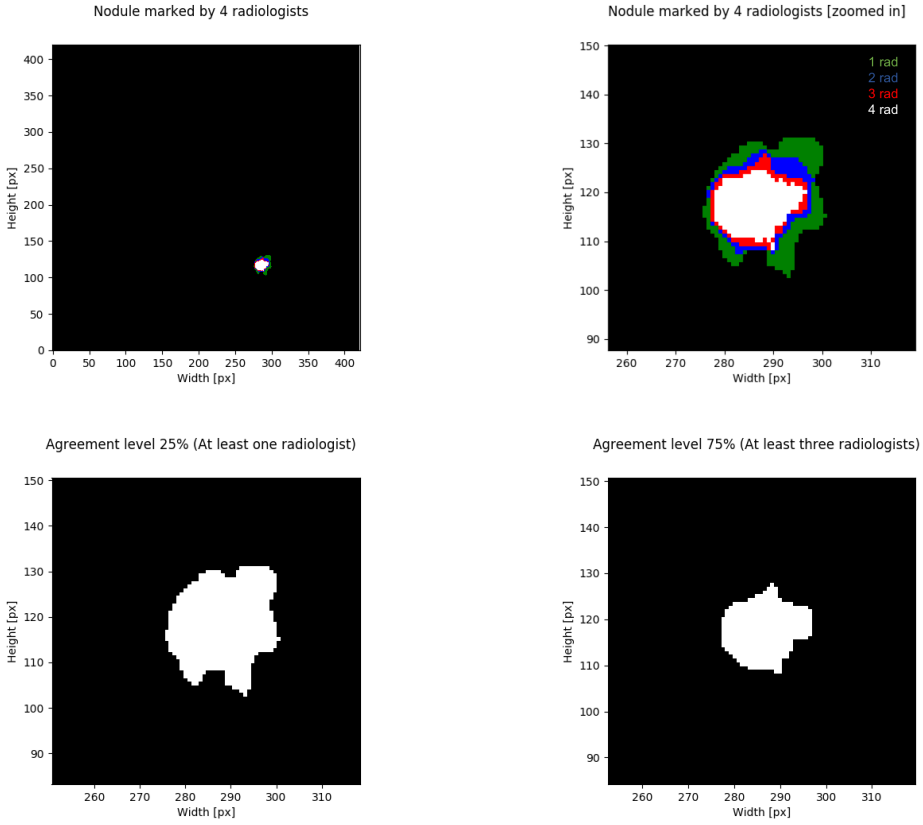


Figure 4.1: Top left: A ground truth slice with a nodule marked by the four radiologists. Green pixels are marked by one radiologist, blue by two, red by three and white by all four. Top right: a zoomed in ground truth slice . Bottom left: A ground truth slice corresponding to the dataset with an agreement level of 25 %, where all the marked pixels were included. Bottom right: Only the pixels marked by at least three radiologists, i.e., white and red pixels, were included. This constituted the dataset with an agreement level of 75 %

4.1.2 Datasets

Apart from the two ground truth datasets created, three additional datasets were distinguished. The difference between the three datasets were the number of data and the difficulty in nodule identification. Each of these three dataset was split into training, validation and test sets. The training set was used to train the network. After each training phase, the validation set was used to evaluate the performance of the network and determine when the network had converged. In the end, the test set was used to evaluate the network accuracy. Table 4.2 shows the distribution of the three datasets.

Datasets										
Datasets	Number of images	Training images (60 %)			Validation images (20 %)			Test images (20 %)		
		P	NP	Tot	P	NP	Tot	P	NP	Tot
Dataset 1	125	75	0	75	20	5	25	20	5	25
Dataset 2	275	165	0	165	45	10	55	45	10	55
Dataset 3	574	344	0	344	100	15	115	100	15	115

Table 4.2: Representation of the different datasets. All training images had the presence (P) of nodules, while validation and test images had both images with and without nodules (no presence, NP).

Originally the database from the LIDC-IDRI contained 1018 images. However, some of these contained errors in the XML-file and did not have complete metadata. There were missing tags, such as *Slice Location*, *Slice Thickness* and *Pixel Spacing* which were necessary to normalize the images. Furthermore, 264 images did not contain any nodules. Only 574 images were included in dataset 3, while the rest were dismissed. The comparison between the performance of the two systems was done with the results obtained using this dataset.

Dataset 1 contained images with low noise, and most importantly, nodules that were clear and easy to distinguish. The images that presented the largest nodules were included in this dataset. Several examples are shown in Figure 4.2, 4.3, 4.4.

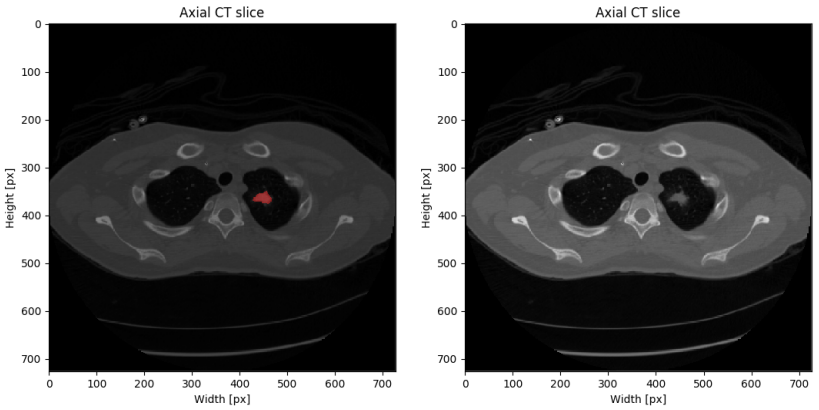


Figure 4.2: A slice of a CT image included in dataset 1. Left: The axial CT slice with the ground truth mask of agreement level 75 %. Right: The same axial CT slice without the mask. As illustrated, the nodule can easily be distinguished and there is no presence of other structures that can be misinterpreted as a nodule

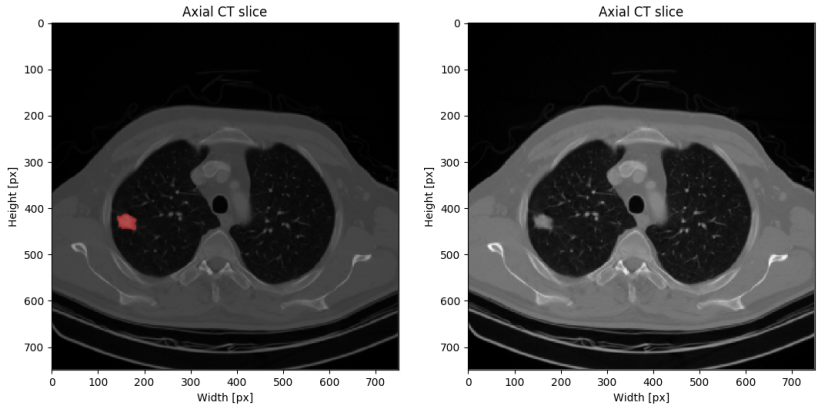


Figure 4.3: A slice of a CT image included in dataset 1. Left: The axial CT slice with the ground truth mask of agreement level 75 %. Right: The same axial CT slice without the mask. The nodule can easily be recognized as it presents a big diameter and clear shape. In this slice, the blood vessels are more prominent

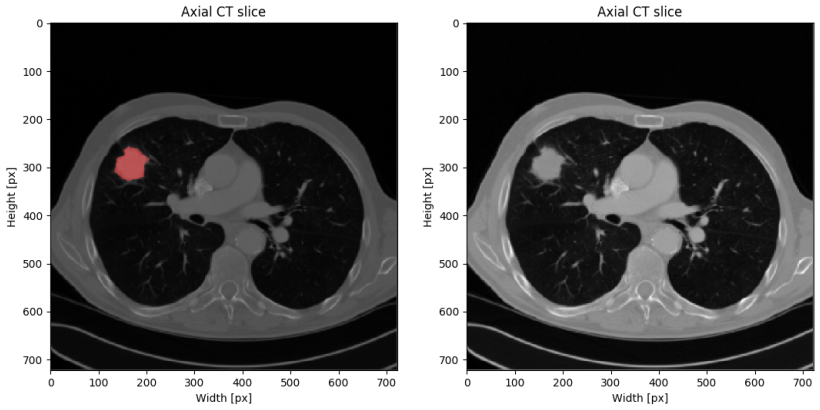


Figure 4.4: A slice of a CT image included in dataset 1. Left: The axial CT slice with the ground truth mask of agreement level 75 %. Right: The same axial CT slice without the mask. This slice shows one of the largest and most clear nodule in the entire dataset

Dataset 2 contained all the images from dataset 1, as well as 150 additional images. The new images contained nodules with smaller size, but they could still be recognized because of their shape and appearance. The presence of other structures, such as blood vessels, could make the detection more difficult. Several examples can be seen in Figures 4.5, 4.6 and 4.7.

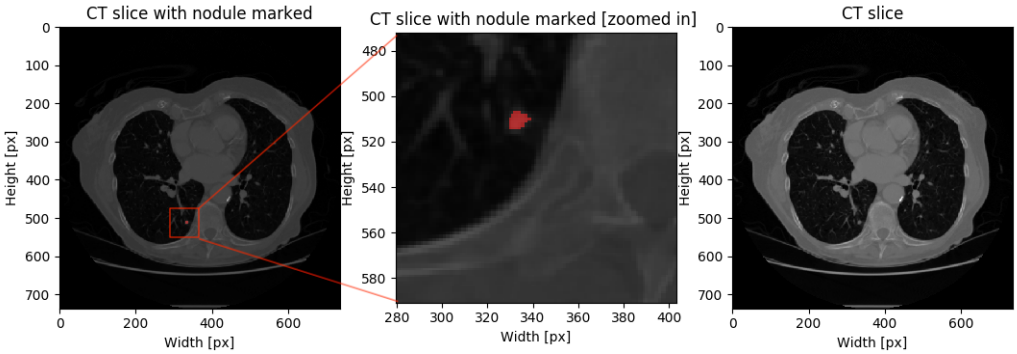


Figure 4.5: A slice of a CT image included in dataset 2. Left: The axial CT slice with the ground truth mask of agreement level 75 %. Right: The same axial CT slice without the mask. The nodule is of smaller size, and due to its appearance it can be misinterpreted as a blood vessel (brighter spots)

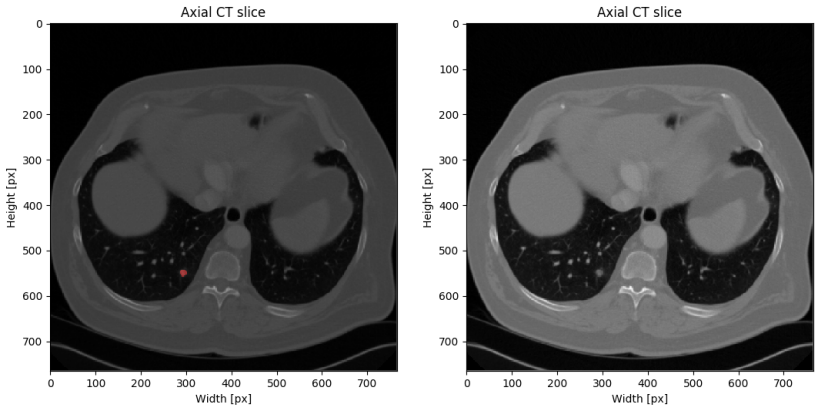


Figure 4.6: A slice of a CT image included in dataset 2. Left: The axial CT slice with the ground truth mask of agreement level 75 %. Right: The same axial CT slice without the mask. The size of the nodule is smaller compared to the nodules in dataset 1, but can still be distinguished

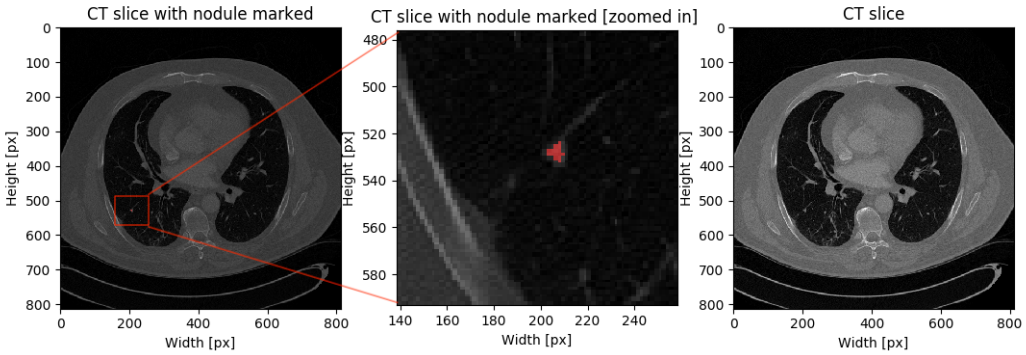


Figure 4.7: A slice of a CT image included in dataset 2. Left: The axial CT slice with the ground truth mask of agreement level 75 %. Right: The same axial CT slice without the mask. The presence of other structures with similar shape and pixel values makes it more difficult to detect the nodules

Dataset 3 contained all CT images available from the LIDC-IDRI database. Besides the entire dataset 2, 290 additional images were added. This dataset contained all types of nodules. The nodules most difficult to detect were those with a small size and those close to many blood vessels, as the blood vessels could be falsely interpreted as nodules. Several examples can be seen in Figures 4.8, 4.9 and 4.10.

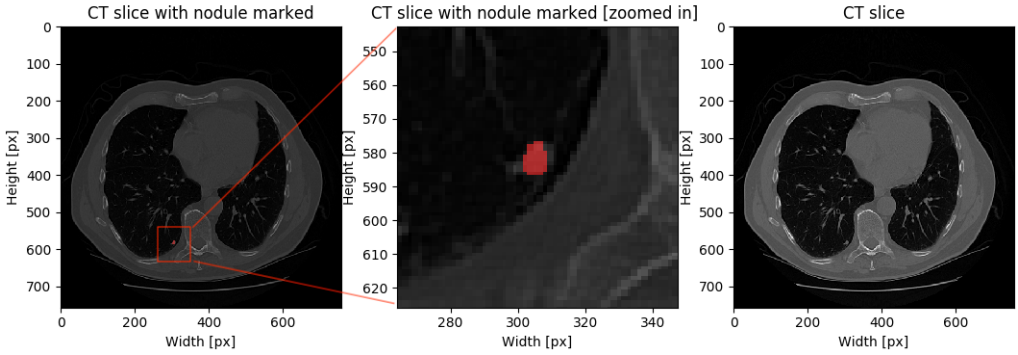


Figure 4.8: A slice of a CT image included in dataset 1. Left: The axial CT slice with the ground truth mask of agreement level 75 %. Right: The same axial CT slice without the mask. The nodule presented in this image can be considered difficult to distinguish due to the presence of blood vessels with similar appearance

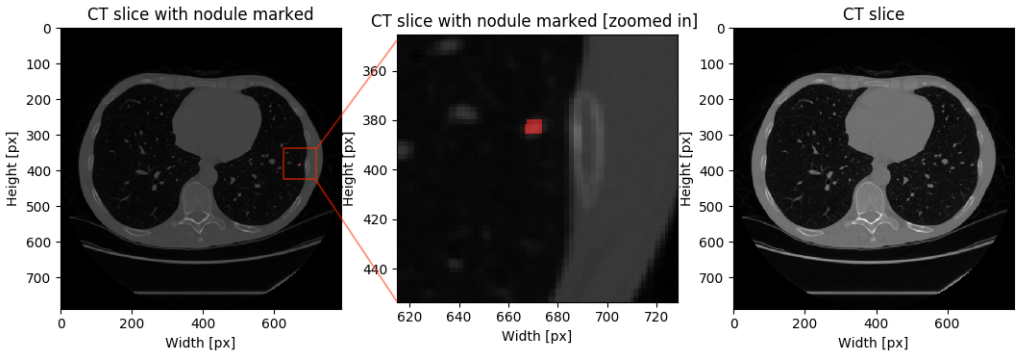


Figure 4.9: A slice of a CT image included in dataset 2. Left: The axial CT slice with the ground truth mask of agreement level 75 %. Right: The same axial CT slice without the mask. This nodule is very difficult to segment as it is hidden due to the presence of multiple blood vessels

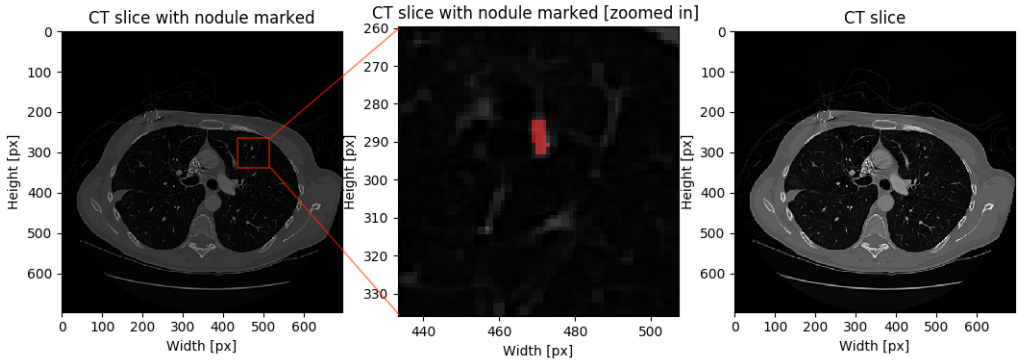


Figure 4.10: A slice of a CT image included in dataset 2. Left: The axial CT slice with the ground truth mask of agreement level 75 %. Right: The same axial CT slice without the mask. Another example of a difficult nodule to segment

4.2 Implementation

NiftyNet [50] is an open source platform for research in medical image analysis. NiftyNet contains the implementation of a 3D U-Net network with a TensorFlow backend. The hardware available had the following specifications:

- CPU: Intel Core i7-6700K, 4 cores @ 4.00GHz
- GPU: GeForce GTX 1070, 8GB
- RAM: 32 GB

In this section, the two systems used in the project are explained. Both systems were implemented with the same network architecture in order to be able to compare performance. Each system was implemented on the three, i.e., three different models of each system were developed.

4.2.1 Network Architecture

The network architecture implemented in both systems is illustrated in its entirety in Figure 4.11. Architecturally, the only change performed was the reduction of feature maps

in the deeper layers. The reason for this decision was the possibility to use a batch size larger than one.

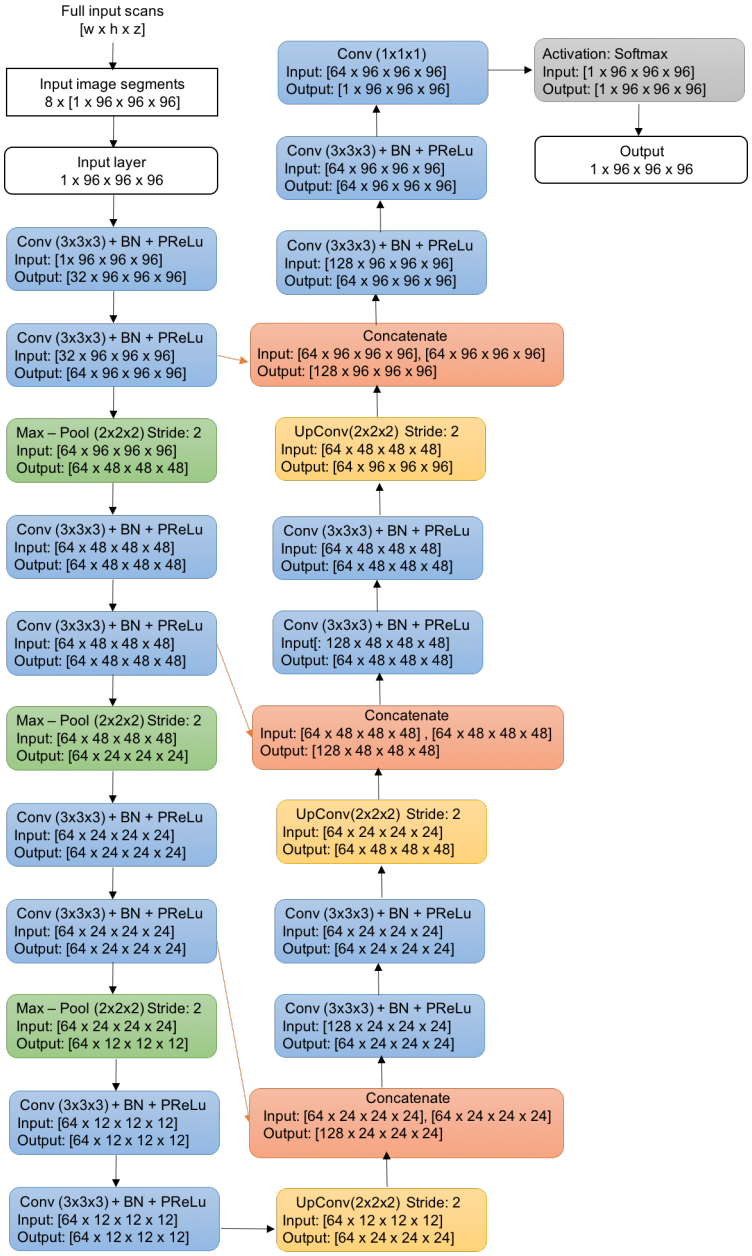


Figure 4.11: Schematic representation of the 3D U-Net network architecture implemented. Each blue box represents three steps: convolution, batch normalization and PReLu activation. The output has the same size as the input and contains a prediction for each voxel

4.2.2 Training

As previously mentioned, the whole 3D image volume cannot be used as input to the network due to memory constraints. For this reason, the training was performed on sampled volume segments of size $96 \times 96 \times 96$ pixels. Eight segments were sampled from each input image volume. The sampling of these segments occurred randomly where each class had the same probability of being sampled. Two of these segments were put in batches, i.e., two segments

were utilized in each training iteration. The Generalized Dice loss mentioned in Equation 3.4 was the loss function implemented in the network to calculate the error.

4.2.3 Single System

This system was developed to be compared to the sequential system. It consisted of a one-step approach in which the semantic segmentation of lung nodules was performed by using the 3D U-Net network. A schematic representation of this method is illustrated in Figure 4.12.

This method was trained, validated, and tested with the three different datasets mentioned in section 4.1.3. The ground truth was the set which had an agreement level of 75 %.

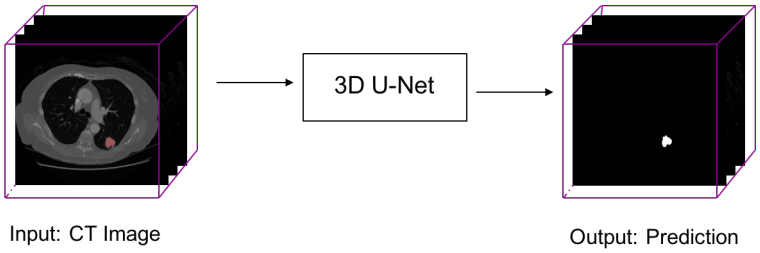


Figure 4.12: Schematic representation of the single system. The 3D U-Net network is fed with input images and outputs a prediction

4.2.4 Sequential System

The sequential system consisted of a two-step approach, in which two 3D U-Net networks were implemented with the same architecture. The aim was to tackle the imbalanced data problem by specializing the networks: the first network having high sensitivity and the second high specificity.

The first network was trained to have very high sensitivity in order to find all nodule voxels, i.e., not acquire any false negatives. High sensitivity results in a network biased to predict the foreground class. Consequently, the prediction of many false positives was bound to occur. This problem was supposed to be solved by the second network which was focused to have high specificity. A high specificity results in the rejection of false positives. The results would be only true positives and true negatives.

To achieve the characteristics of high sensitivity and high specificity, the training of the two networks in the sequential system differed in two aspects: the input volumes and the ground truth datasets. The first network was trained with the 25 % ground truth dataset. This was motivated by the fact that if at least one radiologist thought that a specific voxel belonged to the nodule class, it was because a certain grade of similarity or correlation existed between that voxel and a nodule voxel. The idea was to make the first network more sensitive to nodule voxels. The input volumes were the entire CT images. The input volumes of the second network had a resolution of $96 \times 96 \times 96$ pixels. These volumes were created from both images with and without nodules. The 75 % ground truth dataset was used.

When performing inference, only the predictions from the first network were passed on to the second network. This was implemented by applying a bounding box of size $96 \times 96 \times 96$ pixels around the predictions. These small volumes became the input to the second network. Figure 4.13 illustrates a schematic representation of this system. The positions of the bounding boxes were saved in order to reconstruct the prediction. This step removed a

lot of uninteresting background information and made the data more balanced for the second network.

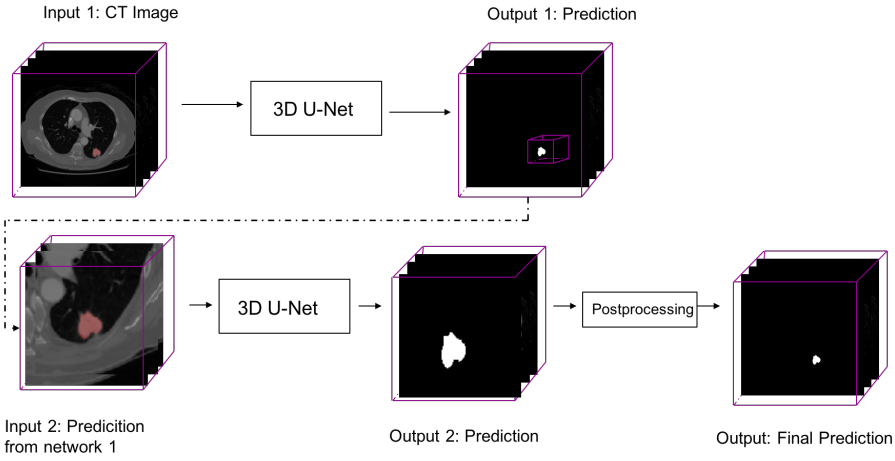


Figure 4.13: Schematic representation of sequential system. The first network which has high sensitivity makes a first prediction of the input image. A bounding box is placed around the predictions and are passed on to the second network. The second network makes a prediction of the inputs. A postprocessing step to put together all the bounding boxes and background is performed

4.2.5 Evaluation

Evaluation was performed quantitatively and qualitatively. The quantitative measures were sensitivity, specificity, Dice and F_2-score (described in section 2.3). Dice and F_2-score measures accuracy. These measures were calculated for the images that contained nodules. The images without presence of nodules were evaluated according to the number of false positives. In addition to the results from each image, the median of all images is presented to give a simple evaluation of the general performance of the system. The median was selected, as the average is more susceptible to outliers. The qualitative measurement was performed by an observer that examined the predictions and calculated the number of nodules found, with the 75 % ground truth dataset as reference.

4.2.6 Pipeline

In order to get a general overview of the whole implementation, Figure 4.14 illustrates the general workflow from data gathering to the final prediction.

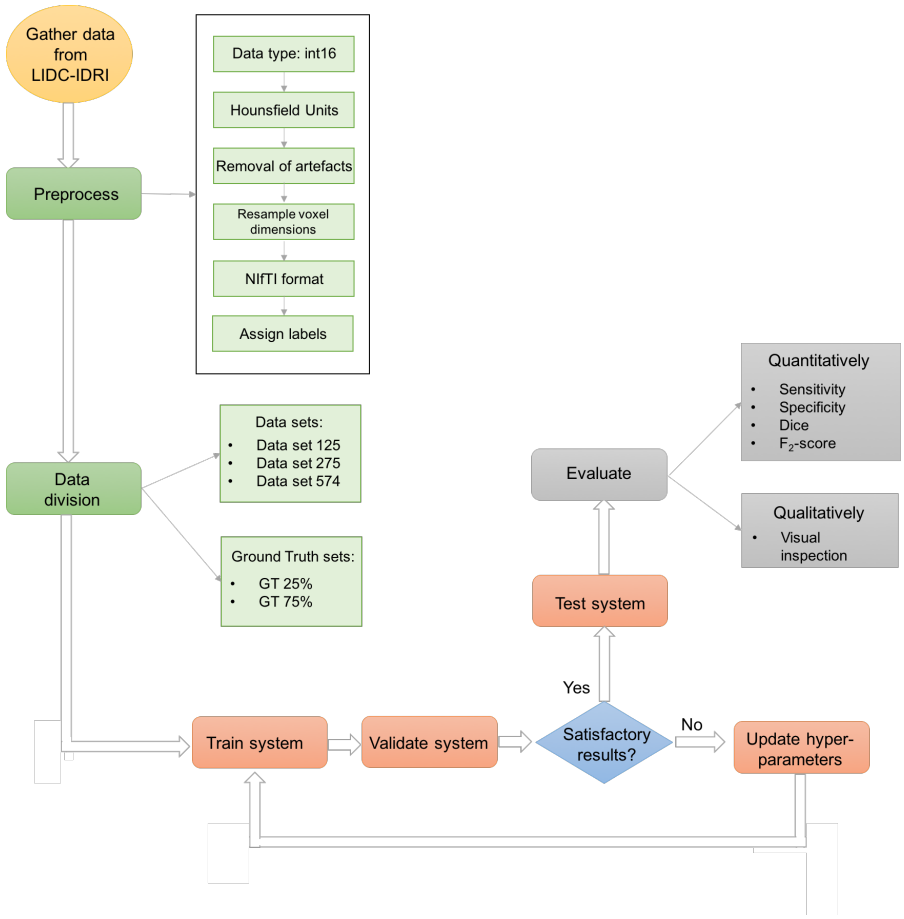


Figure 4.14: Schematic representation of the overall implementation. Data was gathered from the LIDC-IDRI database and preprocessed. The database was divided into three datasets. The system was then fed with the data. When a training process was completed, a validation was performed using the validation data. If the results were not satisfactory, some hyper-parameters were tuned and the training was performed again. This procedure was iterated until satisfactory results were achieved. Testing was then performed by feeding the network with data previously unseen. The last step was to evaluate the final segmentation predictions created during testing

5 Results

This chapter presents the results obtained. The quantitative metrics sensitivity, specificity, Dice and F_2-score of the two systems for the three datasets are presented. The qualitative results are also included. The differences in training and inference time between the two different systems are illustrated. Finally, a comparison in performance between the three different datasets is presented.

All the networks have been trained with a learning rate of 0.0001, weight decay of 0.0001 and a batch size of two.

5.1 Dataset 1

This section provides quantitative and qualitative results of the single system (S1) and the sequential system (S2) for dataset 1. Dataset 1 contained 125 patients in total. The test study was performed with 25 patients. The results for each patient are illustrated in Appendix A.

Model 1 - Sensitivity				
	Quantitative			Qualitative
System	Median	Average	Standard Deviation	Nodules found
S1	82.90%	75.29%	21.19%	42/46 (91.30 %)
S2	80.39%	75.28%	19.09%	41/46 (89.13 %)

Table 5.1: The median, average, standard deviation and nodules found of the sensitivity for the singular system (S1) and the sequential system(S2) are presented.

Model 1 - Specificity			
	Quantitative		
System	Median	Average	Standard Deviation
S1	99.99724033 %	99.99645237 %	2.84×10^{-3} %
S2	99.99152775 %	99.997580539 %	12.6×10^{-3} %

Table 5.2: The median, average, standard deviation of the specificity for the singular system (S1) and the sequential system(S2) are presented.

Model 1 - Accuracy				
	Dice score			F_2-score
System	Median	Average	Standard Deviation	Median
S1	48.23%	48.04%	21.74%	58.46%
S2	24.28%	27.42%	18.86%	41.31%

Table 5.3: The median, average, standard deviation of the Dice score for the singular system (S1) and the sequential system(S2) are presented. The median of the F_2-score is also presented.

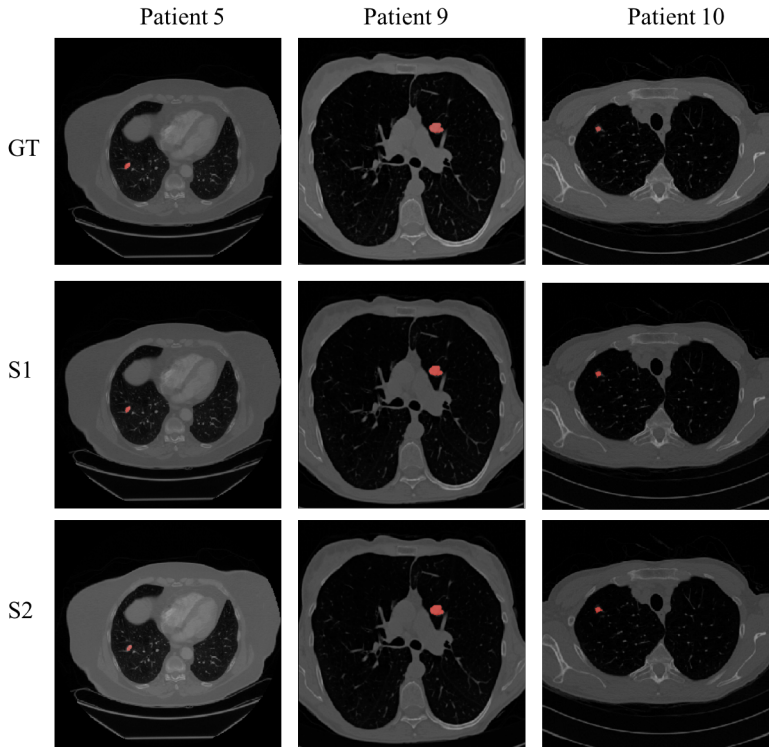


Figure 5.1: Examples of well segmented nodules. The first row shows the ground truth slices of three patients. The second and third row illustrate the predictions performed by the single system (S1) and the sequential system (S2) respectively. Patient number 10 achieved a sensitivity of 100 % in both systems

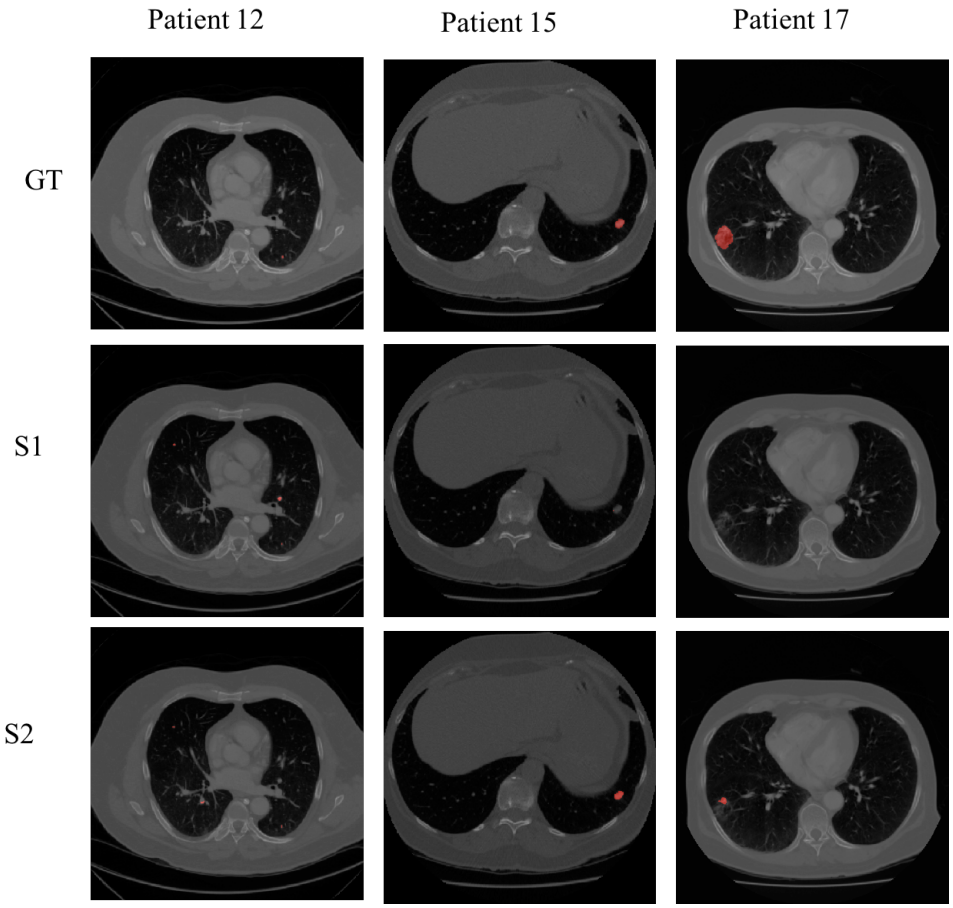


Figure 5.2: Examples of poorly segmented nodules. In some predictions a nodule is missed, the entire shape is not segmented or healthy tissue is incorrectly segmented. The first row shows the ground truth slices of three patients. The second and third row illustrate the predictions performed by the single system (S1) and the sequential system (S2)

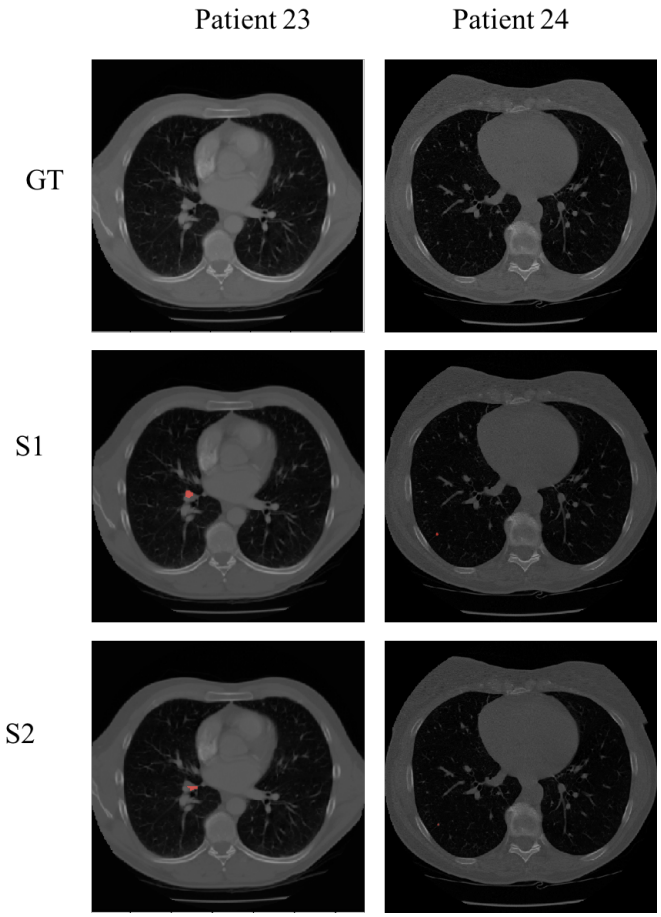


Figure 5.3: Examples of predicted nodules in patients with no presence of lung nodules

5.2 Dataset 2

This section provides quantitative and qualitative results of the single system (S1) and the sequential system (S2) for dataset 2. Dataset 2 contained 275 patients in total. The test study was performed with 55 patients. The results for each patient are illustrated in Appendix A.

Model 2 - Sensitivity				
	Quantitative			Qualitative
System	Median	Average	Standard Deviation	Nodules found
S1	71.09%	62.81%	31.33%	87/98 (88.78 %)
S2	83.79%	78.37%	19.55%	90/98(91.83 %)

Table 5.4: The median, average, standard deviation and nodules found of the sensitivity for the singular system (S1) and the sequential system(S2) are presented.

Model 2 - Specificity			
	Quantitative		
System	Median	Average	Standard Deviation
S1	99.99790183 %	99.99737419 %	2.24515×10^{-3} %
S2	99.99065834 %	99.98769872 %	10×10^{-3} %

Table 5.5: The median, average, standard deviation of the specificity for the singular system (S1) and the sequential system (S2) are presented.

Model 2 - Accuracy				
	Dice score			F_2-score
System	Median	Average	Standard Deviation	Median
S1	42.50%	42.17%	27.90%	53.86%
S2	20.03%	26.78%	22.90%	35.53%

Table 5.6: The median, average, standard deviation of the Dice score for the singular system (S1) and the sequential system(S2) are presented. The median of the F_2-score is presented.

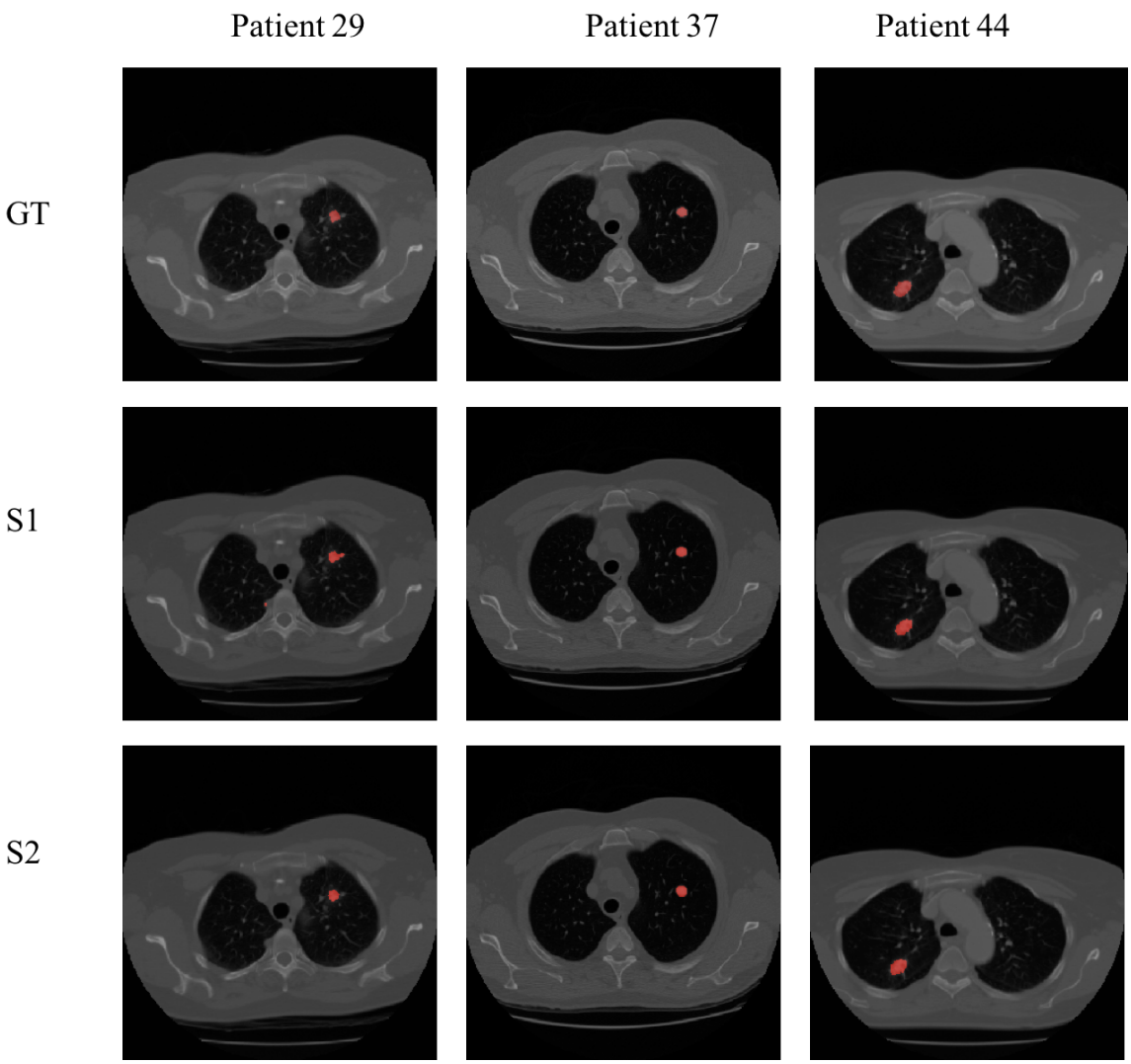


Figure 5.4: Examples of well segmented nodules. The first row shows the ground truth slices of three patients. The second and third row illustrate the predictions performed by the single system (S1) and the sequential system (S2)

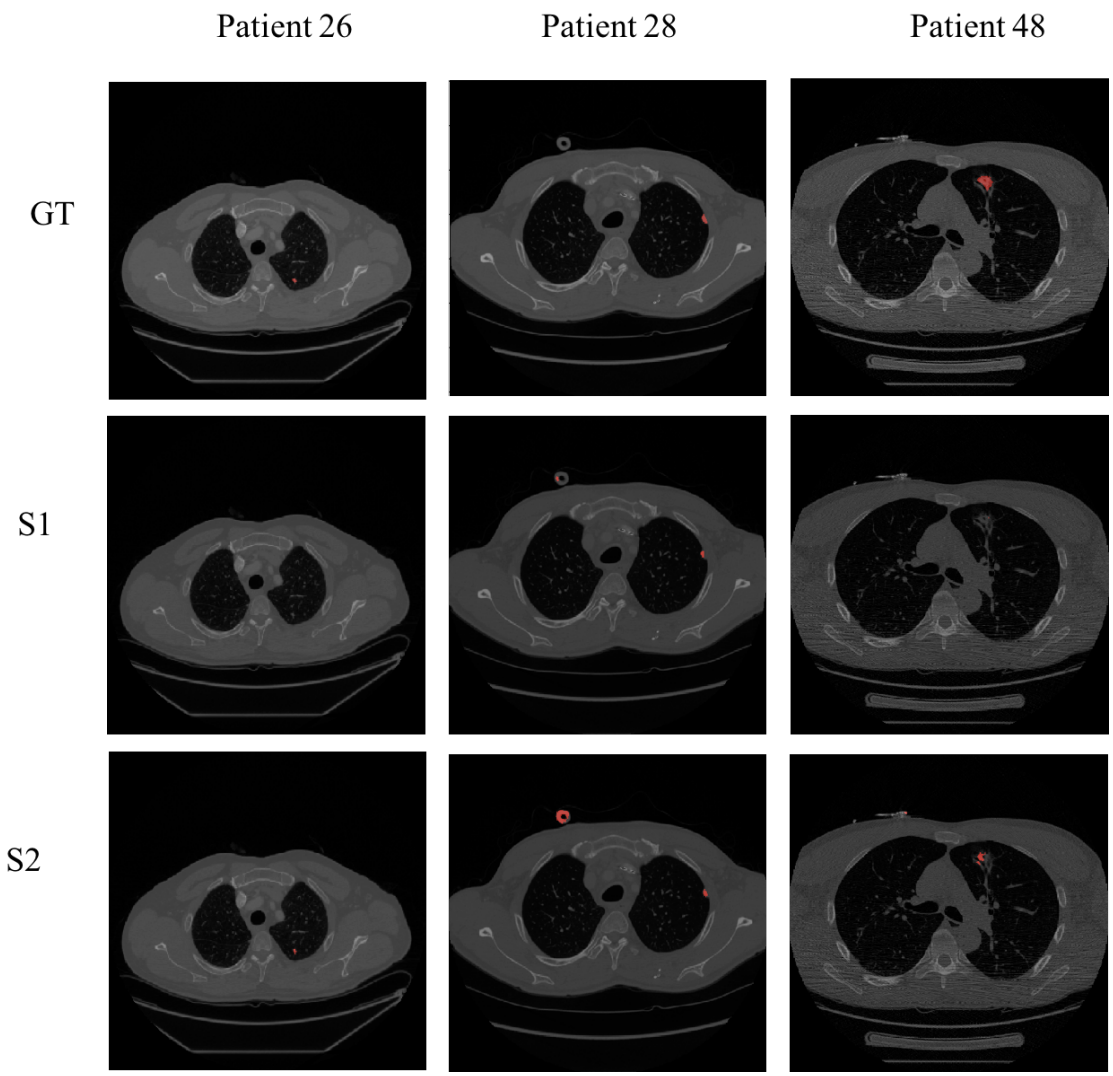


Figure 5.5: Examples of poorly segmented nodules. In some predictions a nodule is missed, the entire shape is not segmented or healthy tissue is incorrectly segmented. The first row shows the ground truth slices of three patients. The second and third row illustrate the predictions performed by the single system (S1) and the sequential system (S2)

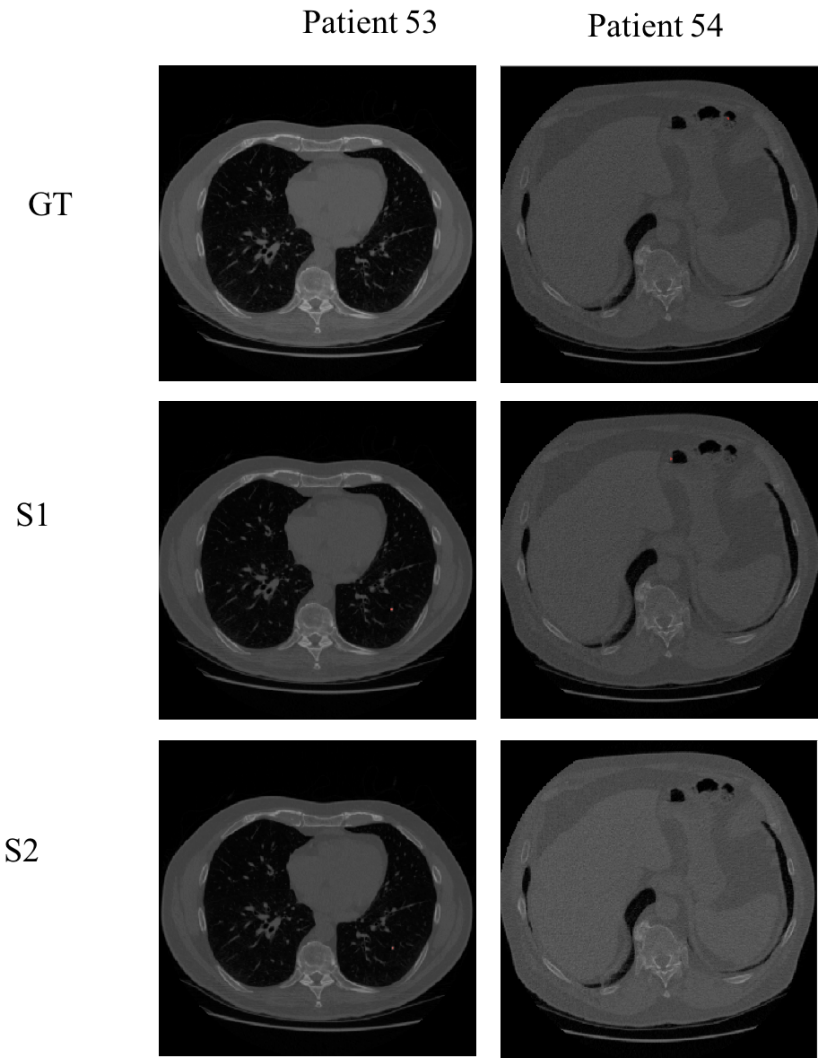


Figure 5.6: Examples of incorrect segmentations in patients with no presence of lung nodules

5.3 Dataset 3

This section provides quantitative and qualitative results of the single system (S1) and the sequential system (S2) for the dataset 3. Dataset 3 contained 574 patients in total, i.e., all of the available data from the LIDC-IDRI database. The test study was performed with 115 patients. The results for each patient are illustrated in Appendix A.

Model 3 - Sensitivity				
	Quantitative			Qualitative
System	Median	Average	Standard Deviation	Nodules found
S1	62.03%	53.82%	31.78%	172/208 (82.69 %)
S2	83.10%	70.80%	29.50%	173/208 (83.17 %)

Table 5.7: The median, average, standard deviation and number of nodules of the sensitivity for the singular system (S1) and the sequential system(S2) are presented.

Model 3 - Specificity			
	Quantitative		
System	Median	Average	Standard Deviation
S1	99.99873866 %	99.99809282 %	1.8×10^{-3} %
S2	99.9954062 %	99.99380404 %	5.51×10^{-3} %

Table 5.8: The median, average, standard deviation of the specificity for the singular system (S1) and the sequential system(S2) are presented.

Model 3 - Accuracy				
	Dice score			F_2-score
System	Median	Average	Standard Deviation	Median
S1	32.71%	37.16%	28.47%	41.76%
S2	21.60%	30.95%	27.34%	38.05%

Table 5.9: The median, average, standard deviation of the Dice score for the singular system (S1) and the sequential system(S2) are presented. The median of the F_2-score is presented.

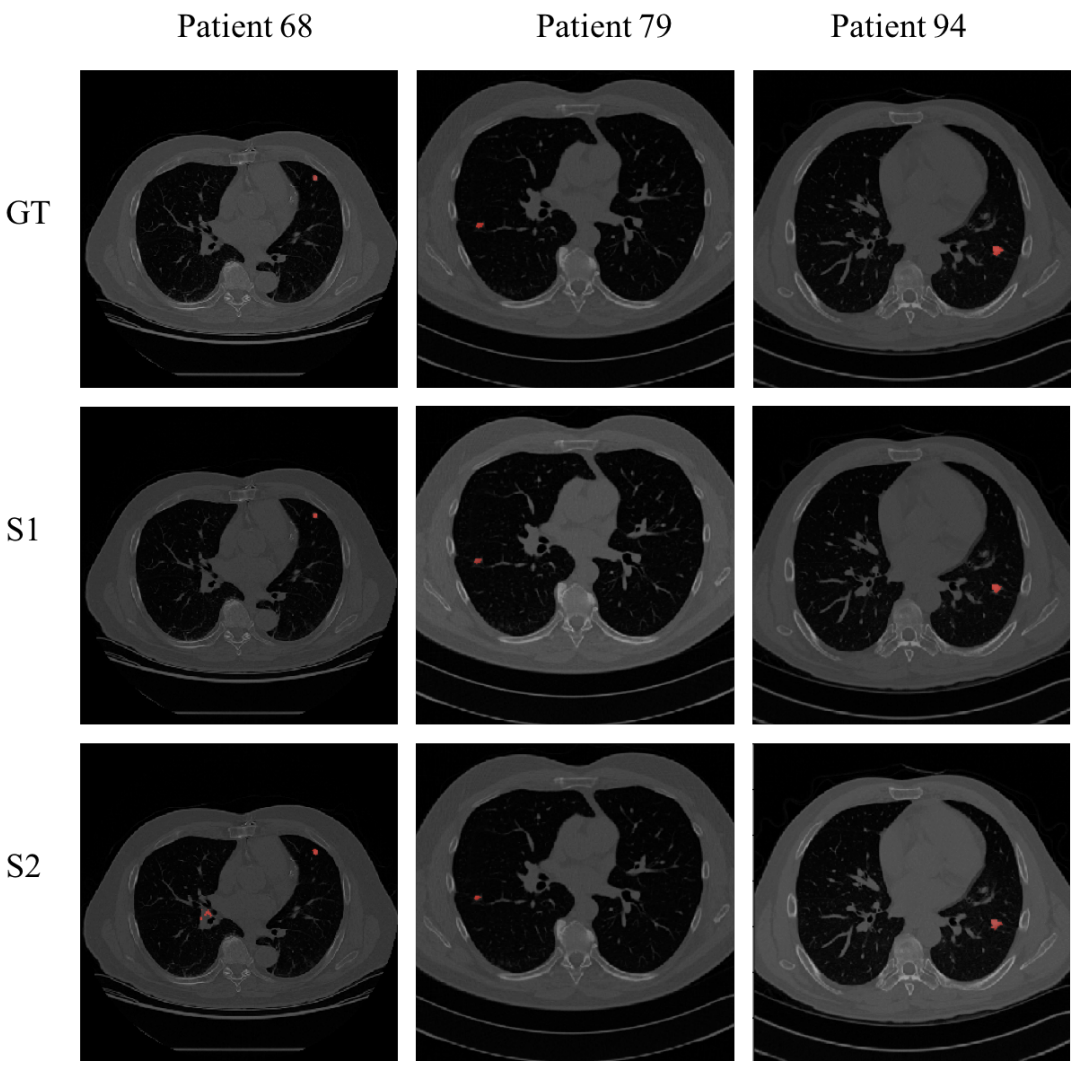


Figure 5.7: Examples of well segmented nodules. The first row shows the ground truth slices of three patients. The second and third row illustrate the predictions performed by the single system (S1) and the sequential system (S2)

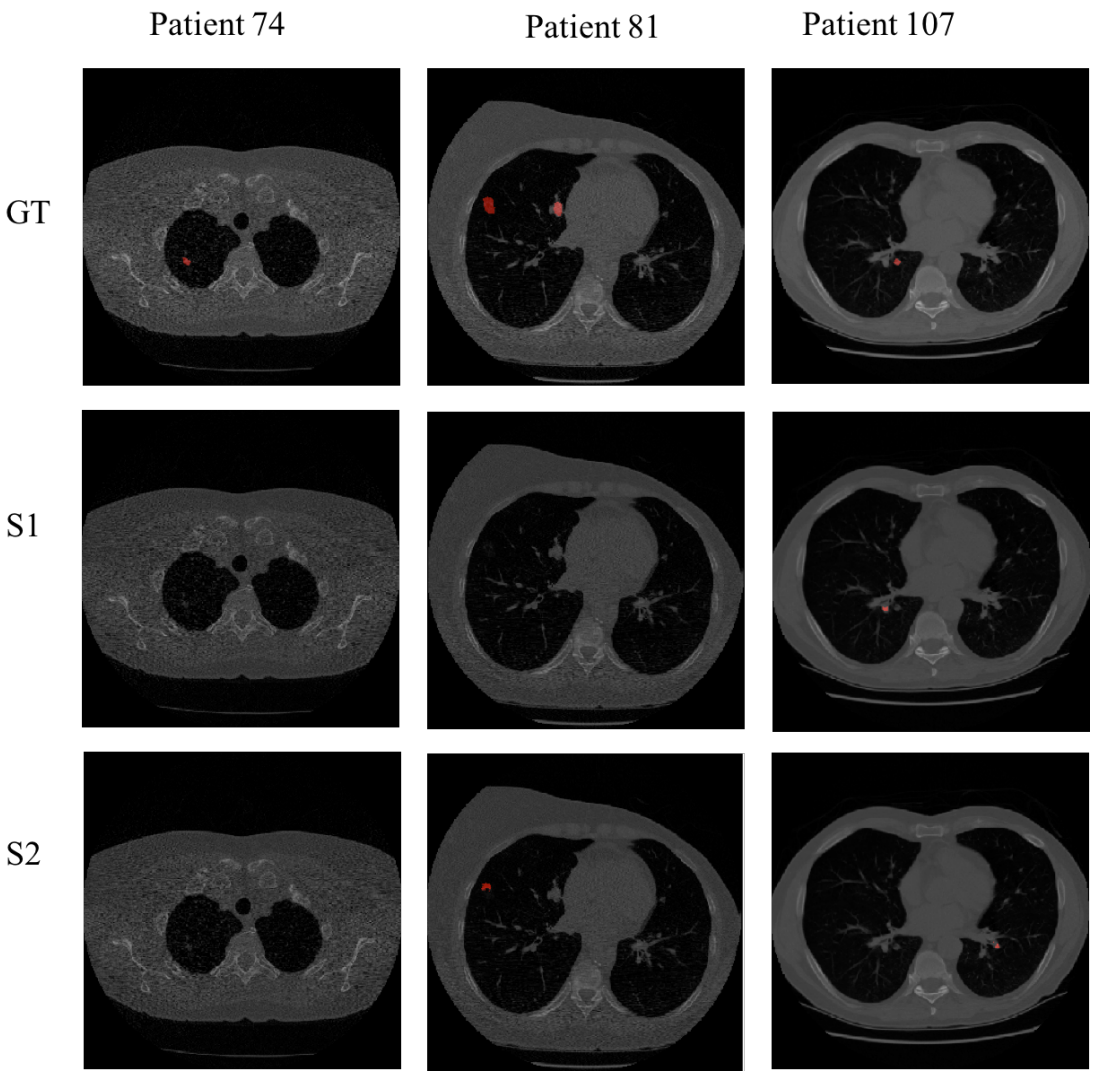


Figure 5.8: Examples of poorly segmented nodules. In some predictions a nodule is missed, the entire shape is not segmented or healthy tissue is incorrectly segmented. The first row shows the ground truth slices of three patients. The second and third row illustrate the predictions performed by the single system (S1) and the sequential system (S2) respectively

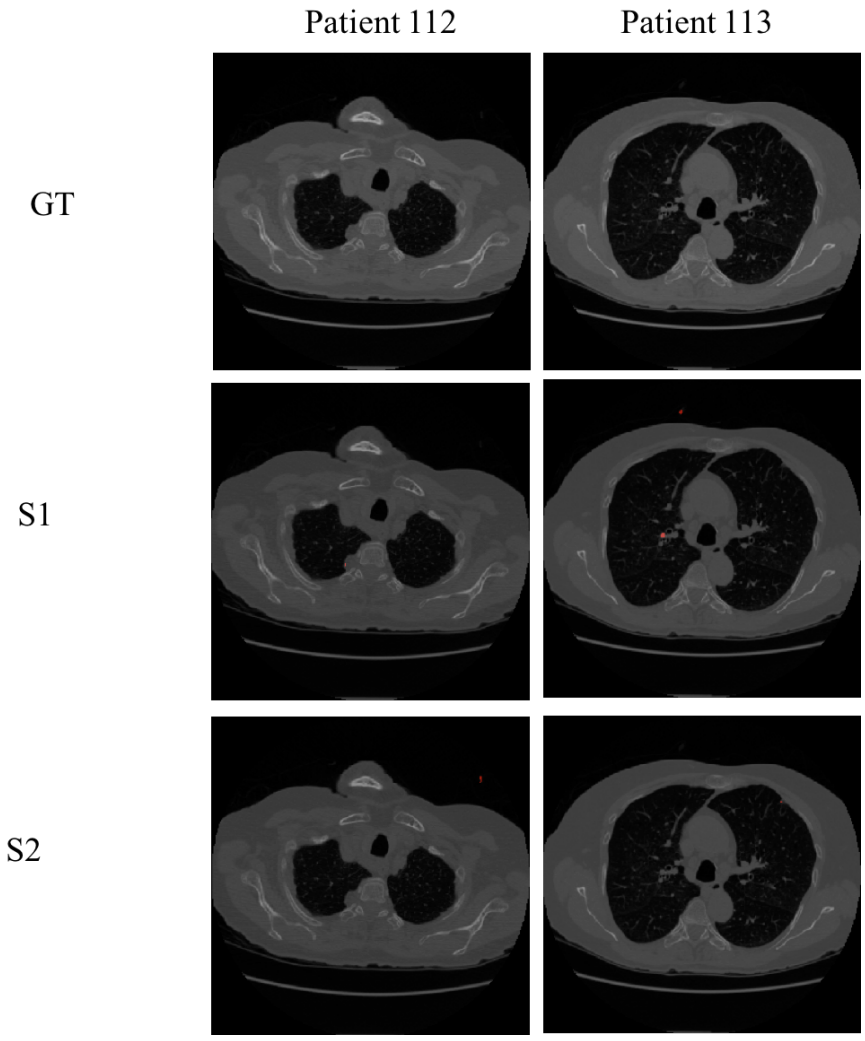


Figure 5.9: Examples of incorrect segmentations in patients with no presence of lung nodules

5.4 Training and Inference Time

This section compares the training and inference time of the two systems

Training and inference time		
Method	Training time	Inference time for one CT image
Single system	36h	47s
Sequential system	86h	1min 5s

Table 5.10: Training and inference time of the two systems.

5.5 Datasets

This section gives an overview of the influence of the data on the performance of the systems.

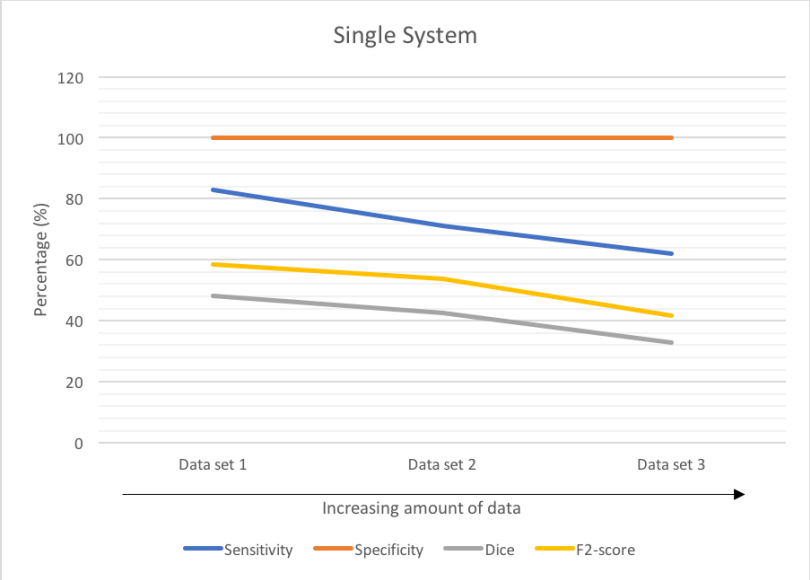


Figure 5.10: The median value of sensitivity, specificity, Dice and F_2-score of the three datasets using the single system

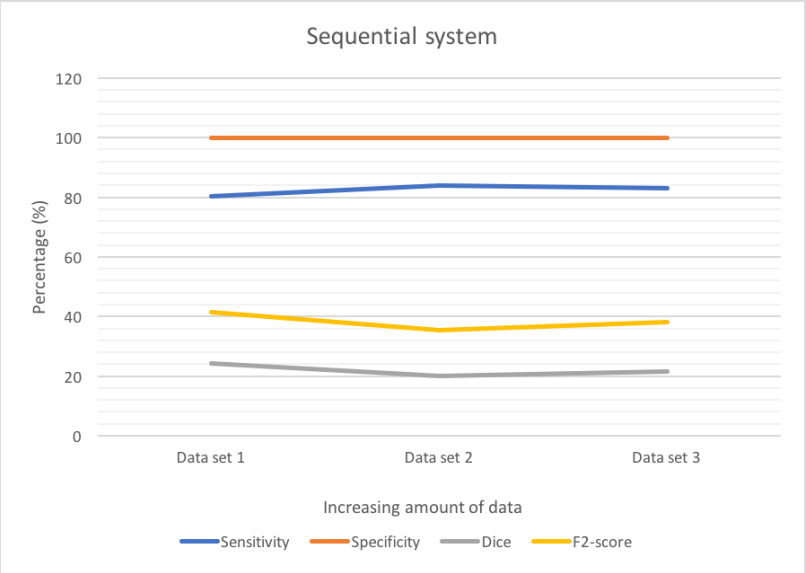


Figure 5.11: The median value of sensitivity, specificity, Dice and F_2-score of the three datasets using the sequential system

Figure 5.12: The specificity of the two systems is plotted in order to observe the trend of the curves in detail

5.6 Ground Truth Reliability

This section provides an overview of the reliability of the ground truth. This was calculated by evaluating the performance of the radiologists with the same metrics used to evaluate the two systems. The segmentation performed by each radiologist was compared to the ground truth dataset corresponding to the 75 % agreement level. This calculations were limited since (1) the radiologists were not independent because they could see the annotations from the other radiologists before they took the final decision and (2) the ground truth was created from the annotations of these radiologists. This means that the radiologists were judged entirely based on their own results, see section 4.1. This means that the radiologists' performance can be assumed to be significantly overestimated.

Sensitivity			
	Quantitative		
System	Median	Average	Standard Deviation
Radiologist 1	69.05%	63.50%	34.50%
Radiologist 2	99.74%	98.91%	1.50%
Radiologist 3	98.33%	83.81%	29.60%
Radiologist 4	95.10%	91.08%	10.90%
All 4	96.70%	84.33%	26.88%

Table 5.11: The median, average and standard deviation of the sensitivity for each radiologist, as well as the combination of all four, are presented. Twenty random samples from the test images were used.

Specificity			
	Quantitative		
System	Median	Average	Standard Deviation
Radiologist 1	99.99978 %	99.99939 %	8.58×10^{-4}%
Radiologist 2	99.99944 %	99.99842 %	2.16×10^{-3} %
Radiologist 3	99.99916 %	99.99824 %	2.17×10^{-3} %
Radiologist 4	99.99946 %	99.99840 %	1.97×10^{-3} %
All 4	99.99945 %	99.99861 %	1.92×10^{-3}%

Table 5.12: The median, average and standard deviation of the specificity for each radiologist, as well as the combination of all four, are presented. Twenty random samples from the test images were used.

Dice score			
	Quantitative		
Radiologist	Median	Average	Standard Deviation
Radiologist 1	61.17%	62.14%	26.90%
Radiologist 2	82.58%	80.71%	11.40%
Radiologist 3	72.74%	65.98%	25.41%
Radiologist 4	79.38%	76.54%	83.27%
All 4	76.06%	71.34%	22.27%

Table 5.13: The median, average and standard deviation of the Dice score for each radiologist, as well as the combination of all four, are presented. Twenty random samples from the test images were used.

Four new ground truth datasets were created from twenty random samples from the test images. The aim was to see how the ground truth datasets vary depending on which radiologist is responsible for the segmentation. Each ground truth dataset has considered the segmentations of three of the four radiologists available and an agreement level of 67 % has been established. The four different ground truth were compared to each other, giving six different combinations.

Resemblance between the datasets			
Quantitative			
Ground Truth datasets	Median Resemblance	Average Resemblance	Standard deviation
GT1 and GT2	96.70%	94.50%	8.49%
GT1 and GT3	95.63%	95.17%	3.78%
GT1 and GT4	91.21%	88.44%	13.34%
GT2 and GT3	95.62%	95.77%	2.63%
GT2 and GT4	91.36%	88.48%	12.69%
GT3 and GT4	91.12%	88.25%	13.12%

Table 5.14: Resemblance between the ground truth datasets based on nodule voxels.

6 Discussion

This chapter includes an analysis of the results obtained, followed by a reflection of the data and the implementation. Proposals for future work concludes the chapter.

6.1 Analysis of Results

This section analyses the results in order to answer the problem statements.

Performance

The sequential system was based on sensitivity and specificity, and these two were therefore the main quantitative metrics in this project. The sequential system achieved a sensitivity of 83.1 % while the single system achieved a sensitivity of 62.03 %. In this aspect, an improvement of 34 % is obtained with the sequential system. The system performs consistently with more than 50 % of the data above an average sensitivity of 70.80 %.

Specificity remains constant and near 100 % for both systems. This was suspected since specificity takes into account the correctly predicted background voxels, i.e, true negatives which constitutes more than 99 % of the input volumes. Despite a specificity near 100 %, a trend can be seen in Figure 5.12. The single system presents a higher specificity.

The Dice scores are low in both systems but particularly in the sequential system with a score of 21.6 %. The single system was more accurate reaching a Dice score of 32.71 %. By analyzing the qualitative results, the single and sequential system segmented a majority of the nodules correctly, only missing 17.3 % and 16.8 % respectively.

Data Influence

The variability in data affects the single system the most. As observed in the results, the sensitivity decreases from 82.9 % when using the first dataset to 62.03 % when using the third dataset. All quantitative measurements experienced a decrease of 25 % except specificity that shows a slight increase as the number of data expands. The influence of data in the sequential system is almost negligible. The specificity, Dice and F_2-score measurements experience a

decrease when the second dataset is used, but increases again with the use of the entire dataset. The sensitivity reaches its highest value, 83.79 %, in the second dataset and is almost kept constant for the third dataset for which it is only decreased by 0.008 %.

Training and Inference time

The sequential system consists of two networks, and therefore the training time is the training time of both trainings combined. For this reason, the training time of the sequential system is twice as high as the single system, which only consists of one network. The number of iterations of the different trainings vary depending on the dataset used.

Ground truth reliability

Three radiologists achieved a sensitivity above 95 % when comparing their own segmentations with the 75 % ground truth dataset. However, the first radiologist achieved a sensitivity of 69.05 %. The results of the specificity were similar. The first radiologist obtained a Dice score of 61.17 % while the other three achieved a Dice score above 70 %. When comparing the ground truths, it can be seen that all ground truth datasets presents a standard deviation around 5-10 %.

6.2 Discussion of Results

The results were partially unexpected based on the architecture motivation of the sequential system. The sequential system achieved a sensitivity of 83.1 % for the whole dataset. This is an adequate result considering that sensitivity is calculated voxel-wise. This may be an unfair estimation of the performance of the system. Although the system has found a nodule and predicted the area, it can be punished by adding or missing some border voxels. To some degree the ground truth is unreliable. As seen in image 4.1, the four radiologists could not agree about the shape of the nodule and only 34.8 % of nodules marked by one radiologist were marked by all four. This means that the ground truth dataset with an agreement level of 75 % can present mislabelled voxels and uncertainty. Additionally, in some cases the radiologists considered different shapes for the same nodule, and the coinciding pixels do not correspond to an entire nodule with a diameter greater than 3 mm. Hence, the systems do not segment these nodules because the diameter is too small.

Section 5.6 shows the variability in the ground truth and the performance of the radiologists. It is not surprising, knowing the amount of information that must be reviewed in order to establish an accurate diagnosis. There are many slices per patient, containing many pixels that the radiologists must go through. An example of this disagreement can be observed in Figure 4.1. The statistical measurements described in section 5.6 were performed in order to prove that it is suboptimal to compare the results of the system to a level of performance that is impossible to achieve. The radiologists cannot reach 100 % accuracy, and their segmentation is used as ground truth. The implemented systems can only ever reach an accuracy as good as the ground truth it is trained with. The sequential system still achieved a sensitivity higher than the first radiologist, showing that this system can be at least as accurate as an experienced radiologist. Furthermore, there is a variance in the ground truth. Depending on which radiologists are included in the ground truth dataset the number of nodule voxels can vary. A standard deviation of almost 10 % is presented in the ground truth. The ground truth will never be 100% reliable, as it is created with a factor of human error and subjectivity.

The challenge of the sequential system relies on the second network which had the task of rejecting the false positives. The first network achieved a relatively high sensitivity, but as previously mentioned, many healthy voxels were also predicted as nodules. The high

number of false positives was responsible for the low Dice score achieved by the sequential system. There are several reasons that could have affected the poor performance of the second network. The training was performed in volumes that contained nodules, and the number of volumes with no presence of nodules was relatively low. This was predicted to work well since the validation performed on the second network achieved a Dice score of 90 %, and most volumes with no nodules obtained zero false positives. However, this training method may have made the second network biased to the foreground class. For this reason, the patients with no presence of nodules got many false positives as presented in the results. In addition, due to the difficulty of increasing sensitivity in the first network, the time to train and tune the second network was limited.

Furthermore, Dice score may not be an optimal accuracy measure for this type of applications where the ground truth shows a degree of subjectivity. A more statistical measurement that considers area, localization and number of nodules would be more suitable. For example, if a healthy voxel is segmented as a nodule, and it is close to a segmented nodule, it should not be considered a false positive even though it is not included in the ground truth.

The results from the sequential system suggests that there is potential to use this method as an approach to manage imbalanced training data. However, further research and tuning is necessary, specifically in the second network. One suggestion that may improve the performance of the second network is the further tuning of the Generalized Dice loss function. The contribution of each class could be corrected by the inverse of its volume instead of the inverse of its square volume, since the training data used for the second network was more balanced. Another approach that would be interesting to investigate further would be the use of the Tversky loss function. In this approach the first network would focus on the false negatives while the second network would focus on the false positives.

6.3 Limitations

One of the main challenges during this thesis project was the variability in the data and the attempt to normalize it.

The anisotropic resolution of the CT images complicated the training of discriminative filters. Images with different resolution, i.e., voxel size, needed a different number of voxels to represent the same structure. This created difficulties for the network to establish a pattern and extract features. The resampling to a standard resolution signifies a width and a height different from 512×512 pixels. The new width and height of each image depended on the original voxel dimension. This divergence has been an obstacle to the learning process. By analyzing the results, it can be observed that the systems have made nodule predictions outside of the lungs. The systems have been fed small volumes of size $96 \times 96 \times 96$ pixels, meaning that the entirety of the CT image has not been seen by the systems. Hence, it is difficult to get an overview of the anatomy presented in the image. For this reason, it would be desirable to have equal resolution and image size within the whole dataset. This could be achieved by performing image registration. Although, as previously mentioned in section 2.2.2, the whole image cannot be used as input due to GPU limitations, it could be possible to use volumes with the entire width and height, but not all image slices. Another possibility would be to downsample all images, e.g. from 512×512 to 218×218 slices. However, the risk of downsampling would be that it would decrease the image quality. Furthermore, the image processing could be improved by a more advanced algorithm for the removal of artefacts. However, the basic approach for artefact removal used in this project have seemingly not yielded any negative effects.

The imbalanced data problem was the main challenge of the project. In the beginning of

the project, the systems were biased to segment the whole input as the background class. The database that was used consisted of extremely imbalanced data. The images with nodules presented an average percentage of 0.0038 % nodule voxels. For this reason, all the images with no presence of nodules ($\approx$ 200) were dismissed to make the data slightly less imbalanced. These images were only included in the validation and test set. Due to difficulties in the learning process, a first dataset containing clear, evident, and big nodules was used to investigate if the problem lied within the system or the imbalanced data. Additionally, two other datasets were created in order to investigate how the data affected the performance of the systems.

The hardware restrictions have also limited the development of the systems. A batch size of two has been used to train the networks. By using larger batch sizes, better approximations of the entire data set would have been achieved during training. The hardware limitations made it necessary to reduce the number of feature maps in deeper layers and to use smaller kernel sizes and image segment volumes. Small kernel sizes and image segment volumes yields small receptive fields and possibly insufficient spatial knowledge of the voxel being examined.

The constraint in the 3D U-Net regarding input size mentioned in Equation 3.1 has also constituted an obstacle. The main idea was to have a bounding box of smaller size, e.g. $64 \times 64 \times 48$ pixels. A nodule spreads out as much as 30 slices and 60 pixels in width and height. This would remove even more background, making the data more balanced in comparison to the first network of the sequential system. Furthermore, the second network does not depend on the spatial information as much as the first network, and therefore a smaller input segment would be suitable. In the implementation performed in this thesis, both networks have inputs of size $96 \times 96 \times 96$ pixels due to the given constraint. It would be interesting to try the sequential system with a fully convolutional network without this limitation.

Three networks have been used in the implementation of this thesis and each network has been trained, validated and tested for three datasets. Consequently, nine models were implemented. Each model needed at least 36 hours of training. This time consumption has limited the tuning of hyper-parameters and the exploration of different loss functions and optimizers.

6.4 Future Work

Managing imbalanced data is an important aspect to consider in machine learning. Therefore, it is necessary to perform further investigation in this field. An approach worth further investigation would be the comparison between the sequential system and the singular system when using the same resources. The sequential system uses twice the amount of resources. It would be interesting to analyze the results when increasing the resources of the singular system by adding nodes and layers to the same amount as the sequential system.

A new performance evaluation criteria could also be explored. This is necessary in segmentation applications whose ground truth are not fully reliable. A statistic metric that takes into account marking proximity could be a solution. Voxels near the nodules marked by the radiologists would not be taken into account when calculating performance, neither as false positives or true positives.

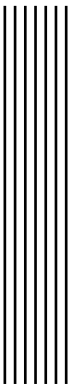
The algorithms developed can be used for any type of segmentation by changing the data and retraining the networks. It would be interesting to see how the sequential system performs with other types data, with more reliable ground truth and less imbalance within the data.

7 Conclusion

The main objective of this thesis was to explore a new approach to manage imbalanced training data. A deep learning framework was developed to segment lung nodules from thoracic CT images where the average percentage of nodule voxels was 0.0038 %.

The sequential system achieved a higher sensitivity than the single network. The sequential and single system obtained a sensitivity of 83.1 % and 62.03 % respectively. When the dataset contained images of clear and easily distinguishable nodules, both systems performed similarly. However, when the dataset was more generalized, the performance of the sequential system was superior, achieving an improvement of 34 %.

The sequential system only missed 16.83 % of the nodules, and shows promise to reach better results with continued development. However, it obtained a low Dice score of 21.6 % since multiple false positives were segmented. This performance is too low, and the system cannot be used in medical applications yet.



References

[1] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012 (cit. on pp. 1, 3).

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016 (cit. on pp. 1, 3, 5, 6, 8–10).

[3] Simon S. Haykin. *Neural networks and learning machines*. Third. Pearson Education, 2009 (cit. on p. 4).

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification." In: *CoRR* abs/1502.01852 (2015) (cit. on p. 5).

[5] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323 (1986) (cit. on p. 5).

[6] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2014). URL: http://arxiv.org/abs/1412.6980 (cit. on p. 7).

[7] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *J. Mach. Learn. Res.* 15 (2014), pp. 1929–1958 (cit. on p. 7).

[8] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Improving neural networks by preventing co-adaptation of feature detectors". In: *CoRR* abs/1207.0580 (2012) (cit. on p. 7).

[9] Steven J. Nowlan and Geoffrey E. Hinton. "Simplifying Neural Networks by Soft Weight-Sharing". In: *Neural Computation* 4 (1992), pp. 473–493 (cit. on p. 7).

[10] Tishby, Levin, and Solla. "Consistent inference of probabilities in layered networks: predictions and generalizations". In: *International 1989 Joint Conference on Neural Networks* 2 (1989), pp. 403–409 (cit. on p. 7).

[11] Anders Krogh and John A. Hertz. "A Simple Weight Decay Can Improve Generalization". In: *Proceedings of the 4th International Conference on Neural Information Processing Systems* (1991), pp. 950–957 (cit. on p. 7).

[12] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *CoRR* abs/1502.03167 (2015). URL: http://arxiv.org/abs/1502.03167 (cit. on p. 8).

[13] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *Nature* 521 (2015), pp. 436–444 (cit. on p. 8).

[14] Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Computation* 1 (1989), pp. 541–551 (cit. on p. 8).

[15] Yi-Tong Zhou, Rama Chellappa, Aseem Vaid, and B. Keith Jenkins. “Image restoration using a neural network”. In: *IEEE Trans. Acoustics, Speech, and Signal Processing* 36 (1988). DOI: 10.1109/29.1641. URL: https://doi.org/10.1109/29.1641 (cit. on p. 9).

[16] David H Hubel and Torsten N Wiesel. “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex”. In: *The Journal of physiology* 160.1 (1962), pp. 106–154 (cit. on p. 10).

[17] Alexey A. Novikov, David Major, Maria Wimmer, Dimitrios Lenis, and Katja Bühler. “Deep Sequential Segmentation of Organs in Volumetric Medical Scans”. In: *CoRR* abs/1807.02437 (2018). URL: http://arxiv.org/abs/1807.02437 (cit. on p. 10).

[18] Abdel Aziz Taha and Allan Hanbury. “Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool”. In: *BMC Medical Imaging* (2015). ISSN: 1471-2342. DOI: 10.1186/s12880-015-0068-x. URL: https://doi.org/10.1186/s12880-015-0068-x (cit. on p. 11).

[19] W.A. Kalender. *Computed Tomography: Fundamentals, System Technology, Image Quality, Applications*. Wiley, 2011. ISBN: 9783895786440. URL: https://books.google.es/books?id=gfLWmRjoyPMC (cit. on pp. 13, 14).

[20] *Wikipedia: Human anatomy planes signatures*. https://commons.wikimedia.org/wiki/File:Human_anatomy_planes_signatures.svg. Accessed: 2018-12-26 (cit. on p. 13).

[21] *Lung nodules: A guide for the patient*. https://lungcanceralliance.org/wp-content/uploads/2017/09/Understanding_Lung_Nodules_Brochure_dig.pdf. Accessed: 2018-12-26 (cit. on p. 14).

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully Convolutional Networks for Semantic Segmentation”. In: *CoRR* abs/1411.4038 (2014). URL: http://arxiv.org/abs/1411.4038 (cit. on pp. 15, 16).

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet classification with deep convolutional neural networks”. In: *Commun. ACM* 60 (2017). URL: http://doi.acm.org/10.1145/3065386 (cit. on p. 15).

[24] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *CoRR* abs/1409.1556 (2014). URL: http://arxiv.org/abs/1409.1556 (cit. on pp. 15, 16).

[25] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. “Going deeper with convolutions”. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. 2015, pp. 1–9. URL: https://doi.org/10.1109/CVPR.2015.7298594 (cit. on p. 15).

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *CoRR* abs/1512.03385 (2015). URL: http://arxiv.org/abs/1512.03385 (cit. on p. 15).

[27] Song Han, Huizi Mao, and William J. Dally. “Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding”. In: *CoRR* abs/1510.00149 (2015). URL: http://arxiv.org/abs/1510.00149 (cit. on p. 15).

[28] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation”. In: *CoRR* abs/1511.00561 (2015). URL: http://arxiv.org/abs/1511.00561 (cit. on p. 16).

[29] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. “Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs”. In: *CoRR* abs/1412.7062 (2014). URL: http://arxiv.org/abs/1412.7062 (cit. on p. 16).

[30] Golnaz Ghiasi and Charless C. Fowlkes. “Laplacian Pyramid Reconstruction and Refinement for Semantic Segmentation”. In: *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*. 2016, pp. 519–534. URL: https://doi.org/10.1007/978-3-319-46487-9%5C_32 (cit. on p. 16).

[31] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. “Learning Deconvolution Network for Semantic Segmentation”. In: *CoRR* (2015). URL: http://arxiv.org/abs/1505.04366 (cit. on p. 16).

[32] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. “Brain tumor segmentation with deep neural networks”. In: *Medical image analysis* 35 (2017), pp. 18–31 (cit. on pp. 17, 19).

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*. 2015, pp. 234–241. URL: https://doi.org/10.1007/978-3-319-24574-4%5C_28 (cit. on pp. 17, 19).

[34] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. “3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation”. In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2016 - 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II*. 2016, pp. 424–432. DOI: 10.1007/978-3-319-46723-8_49. URL: https://doi.org/10.1007/978-3-319-46723-8%5C_49 (cit. on pp. 17, 18).

[35] Holger R. Roth, Le Lu, Nathan Lay, Adam P. Harrison, Amal Farag, Andrew Sohn, and Ronald M. Summers. “Spatial Aggregation of Holistically-Nested Convolutional Neural Networks for Automated Pancreas Localization and Segmentation”. In: *CoRR* abs/1702.00045 (2017). URL: http://arxiv.org/abs/1702.00045 (cit. on p. 17).

[36] Konstantinos Kamnitsas, Christian Ledig, Virginia F. J. Newcombe, Joanna P. Simpson, Andrew D. Kane, David K. Menon, Daniel Rueckert, and Ben Glocker. “Efficient Multi-Scale 3D CNN with Fully Connected CRF for Accurate Brain Lesion Segmentation”. In: *CoRR* abs/1603.05959 (2016). URL: http://arxiv.org/abs/1603.05959 (cit. on p. 17).

[37] Patrick Ferdinand Christ et al. “Automatic Liver and Lesion Segmentation in CT Using Cascaded Fully Convolutional Neural Networks and 3D Conditional Random Fields”. In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2016 - 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II*. 2016, pp. 415–423. URL: https://doi.org/10.1007/978-3-319-46723-8%5C_48 (cit. on p. 17).

[38] Yuyin Zhou, Lingxi Xie, Wei Shen, Elliot K. Fishman, and Alan L. Yuille. “Pancreas Segmentation in Abdominal CT Scan: A Coarse-to-Fine Approach”. In: *CoRR* abs/1612.08230 (2016). URL: http://arxiv.org/abs/1612.08230 (cit. on p. 17).

[39] Philipp Krähenbühl and Vladlen Koltun. “Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials”. In: *CoRR* abs/1210.5644 (2012). URL: http://arxiv.org/abs/1210.5644 (cit. on p. 17).

[40] Kevin W. Bowyer, Nitesh V. Chawla, Lawrence O. Hall, and W. Philip Kegelmeyer. “SMOTE: Synthetic Minority Over-sampling Technique”. In: *CoRR* abs/1106.1813 (2011). URL: http://arxiv.org/abs/1106.1813 (cit. on p. 19).

[41] Lara J. Kanbar, Charles C. Onu, Wissam Shalish, Karen A. Brown, Guilherme M. Sant’Anna, Robert E. Kearney, and Doina Precup. “Undersampling and Bagging of Decision Trees in the Analysis of Cardiorespiratory Behavior for the Prediction of Extubation Readiness in Extremely Preterm Infants”. In: *CoRR* abs/1808.07992 (2018). URL: http://arxiv.org/abs/1808.07992 (cit. on p. 19).

[42] Fanny and Tjeng Wawan Cenggoro. “Deep Learning for Imbalance Data Classification using Class Expert Generative Adversarial Network”. In: *CoRR* abs/1807.04585 (2018). URL: http://arxiv.org/abs/1807.04585 (cit. on p. 19).

[43] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. “V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation”. In: *CoRR* abs/1606.04797 (2016). URL: http://arxiv.org/abs/1606.04797 (cit. on p. 19).

[44] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sébastien Ourselin, and M. Jorge Cardoso. “Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations”. In: *CoRR* abs/1707.03237 (2017). URL: http://arxiv.org/abs/1707.03237 (cit. on pp. 19, 20).

[45] Seyed Raein Hashemi, Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, Sanjay P Prabhu, Simon K Warfield, and Ali Gholipour. “Asymmetric similarity loss function to balance precision and recall in highly unbalanced deep medical image segmentation”. In: *arXiv preprint arXiv:1803.11078* (2018) (cit. on p. 19).

[46] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. “Tversky loss function for image segmentation using 3D fully convolutional deep networks”. In: *CoRR* abs/1706.05721 (2017). URL: http://arxiv.org/abs/1706.05721 (cit. on pp. 19, 20).

[47] The Cancer Imaging Archive. *LIDC-IDRI*. URL: https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI. (Accessed: 06.09.2018) (cit. on pp. 21, 22).

[48] Samuel G Armato III et al. “The lung image database consortium (LIDC) and image database resource initiative (IDRI): a completed reference database of lung nodules on CT scans”. In: *Medical Physics* 38.2 (2011) (cit. on p. 22).

[49] Olivia Nackaerts, Frederik Maes, Hua Yan, Paulo Couto Souza, Ruben Pauwels, and Reinhilde Jacobs. “Analysis of intensity variability in multislice and cone beam computed tomography”. In: *Clinical oral implants research* 22.8 (2011), pp. 873–879 (cit. on p. 22).

[50] Eli Gibson and Wenqi Li et al. “NiftyNet: a deep-learning platform for medical imaging”. In: *Computer Methods and Programs in Biomedicine* (2018). ISSN: 0169-2607. URL: https://www.sciencedirect.com/science/article/pii/S0169260717311823 (cit. on p. 28).

A Results

Model 1 - Single system (S1)					
	Quantitative				Qualitative
Patient	Sensitivity	Specificity	Dice	F2-score	nodules
1	86.10%	99.90%	86.85%	86.40%	3/3
2	74.85%	99.90%	56.40%	66.19%	3/3
3	85.86%	99.90%	78.73%	82.86%	4/4
4	72.66%	99.90%	46.72%	59.46%	3/3
5	85.10%	99.90%	81.61%	83.67%	2/2
6	70.89%	99.90%	16.27%	30.26%	1/1
7	97.20%	99.90%	18.87%	36.53%	1/1
8	70.86%	99.90%	44.77%	57.47%	3/5
9	94.02%	99.90%	75.32%	85.53%	1/1
10	100.00 %	99.90%	62.04%	80.34%	1/1
11	88.05%	99.90%	52.43%	69.24%	1/1
12	85.14%	99.90%	66.84%	76.73%	3/4
13	78.36%	99.90%	20.96%	37.40%	2/2
14	53.57%	99.90%	49.75%	51.97%	1/1
15	15.05%	99.90%	22.65%	17.40%	2/3
16	87.29%	99.90%	19.19%	36.08%	1/1
17	25.29%	99.90%	37.98%	29.19%	2/2
18	82.68%	99.90%	31.89%	50.50%	1/1
19	69.63%	99.90%	41.51%	54.78%	2/2
20	83.30%	99.90%	50.11%	65.85%	5/5

Table A.1: Results of the first dataset using the single system. The quantitative measures sensitivity, specificity, Dice and F_2-score are measured. The qualitative results show the number of nodules correctly predicted in relation to the total number of nodules.

Model 1 - Single system (S1)	
Patients with no presence of nodules	False positives
21	2949
22	13364
23	1244
24	3494
25	14529

Table A.2: The test dataset contains five patients with no presence of nodules. This table shows the number of false positives predicted by the single system

Model 1 - Sequential system (S2)					
	Quantitative				Qualitative
Patient	Sensitivity	Specificity	Dice	F2-score	nodules
1	60.46%	99.90%	54.56%	57.96%	3/3
2	50.63%	99.90%	16.37%	27.56%	3/3
3	81.90%	99.90%	73.22%	78.19%	4/4
4	70.19%	99.90%	44.40%	56.97%	3/3
5	94.76%	99.90%	69.34%	82.64%	2/2
6	80.94%	99.90%	11.48%	23.67%	1/1
7	92.25%	99.90%	4.40%	10.28%	1/1
8	60.89%	99.90%	19.80%	33.96%	5/5
9	93.65%	99.90%	24.34%	43.77%	1/1
10	100.00 %	99.90%	13.07%	27.31%	1/1
11	90.23%	99.90%	24.23%	43.18%	1/1
12	83.94%	99.90%	29.52%	48.31%	3/4
13	90.65%	99.90%	12.35%	25.63%	2/2
14	64.81%	99.90%	28.93%	43.32%	1/1
15	73.55%	99.90%	27.92%	44.48%	2/3
16	79.84%	99.90%	3.23%	7.62%	1/1
17	16.74%	99.90%	21.24%	18.29%	2/2
18	84.80%	99.90%	13.63%	27.45%	1/1
19	77.33%	99.90%	28.32%	45.70%	2/2
20	54.08%	99.90%	28.06%	39.45%	2/5

Table A.3: Results of the first dataset using the sequential system. The quantitative measures sensitivity, specificity, Dice and F_2-score are measured. The qualitative results show the number of nodules correctly predicted in relation to the total number of nodules.

Model 1 - Sequential system (S2)	
Patients with no presence of nodules	False positives
21	13108
22	12366
23	22861
24	15409
25	59659

Table A.4: The test dataset contains five patients with no presence of nodules. This table shows the number of false positives predicted by the sequential system

Model 2 - Single system (S1)					
	Quantitative				Qualitative
Patient	Sensitivity	Specificity	Dice	F2-score	nodules
1	84.76%	99.90%	87.21%	85.72%	3/3
2	52.97%	99.90%	50.54%	51.97%	3/3
3	80.66%	99.90%	82.30%	81.31%	4/4
4	68.96%	99.90%	74.08%	70.92%	3/3
5	60.64%	99.90%	69.34%	63.85%	2/2
6	85.46%	99.90%	15.37%	30.26%	1/1
7	82.92%	99.90%	29.27%	47.84%	1/1
8	66.16%	99.90%	25.00%	39.88%	4/5
9	95.30%	99.90%	88.63%	92.51%	1/1
10	97.43%	99.90%	54.35%	73.97%	1/1
11	93.28%	99.90%	56.22%	73.82%	1/1
12	80.06%	99.90%	65.60%	73.57%	4/4
13	68.08%	99.90%	14.60%	27.61%	2/2
14	63.72%	99.90%	58.50%	61.52%	1/1
15	16.30%	99.90%	14.42%	15.49%	2/3
16	82.70%	99.90%	38.59%	56.75%	1/1
17	12.12%	99.90%	17.08%	13.71%	1/2
18	88.32%	99.90%	50.09%	67.67%	1/1
19	55.68%	99.90%	42.54%	49.56%	2/2
20	85.24%	99.90%	63.89%	75.19%	4/5
26	6.44%	99.90%	1.91%	3.31%	1/1
27	90.27%	99.90%	52.80%	70.31%	2/2
28	34.99%	99.90%	25.97%	30.72%	4/5
29	90.07%	99.90%	86.41%	88.57%	3/3
30	0.09%	99.90%	0.06%	0.07%	0/1
31	8.27%	99.90%	13.39%	9.76%	2/2
32	76.83%	99.90%	55.51%	66.60%	1/1
33	79.86%	99.99%	86.42%	82.36%	6/7
34	52.99%	99.90%	31.25%	41.46%	1/1
35	0.00%	99.90%	0.00%	0.00%	0/1
36	68.64%	99.90%	53.15%	61.47%	4/4
37	97.87%	99.90%	75.81%	87.67%	1/1
38	64.39%	99.90%	64.33%	64.36%	5/5
39	5.19%	99.90%	4.52%	4.90%	1/2
40	79.10%	99.90%	18.76%	34.59%	1/1
41	91.85%	99.90%	69.10%	81.16%	1/1
42	79.84%	99.90%	45.03%	60.98%	2/3
43	68.84%	99.90%	40.60%	53.86%	1/1
44	94.06%	99.90%	88.09%	91.57%	1/1
45	0.00%	99.90%	0.00%	0.00%	0/1
46	71.09%	99.90%	14.08%	27.13%	2/2

47	70.27%	99.90%	12.24%	24.25%	1/1
48	4.69%	99.90%	5.53%	4.99%	1/1
49	77.13%	99.90%	27.12%	44.39%	2/2
50	93 %	99.90%	28.12%	48.37%	2/2

Table A.5: Results of the second dataset using the single system. The quantitative measures sensitivity, specificity, Dice and F_2-score are measured. The qualitative results show the number of nodules correctly predicted in relation to the total number of nodules.

Model 2 - Sequential system (S2)					
	Quantitative				Qualitative
Patient	Sensitivity	Specificity	Dice	F2-score	nodules
1	87.63%	99.90%	73.69%	81.47%	3/3
2	82.40%	99.90%	33.08%	51.62%	3/3
3	62.49%	99.90%	44.48%	53.78%	4/4
4	83.50%	99.90%	34.16%	52.92%	3/3
5	92.36%	99.90%	75.47%	84.77%	2/2
6	82.79%	99.90%	11.39%	23.60%	1/1
7	95.67%	99.90%	17.74%	34.70%	1/1
8	73.21%	99.90%	12.14%	24.31%	4/5
9	94.49%	99.90%	22.09%	40.88%	1/1
10	98.12%	99.90%	14.66%	29.93%	1/1
11	91.04%	99.90%	47.88%	66.91%	1/1
12	79.56%	99.90%	26.81%	44.52%	3/4
13	95.67%	99.90%	5.80%	13.29%	2/2
14	78.05%	99.90%	4.01%	9.32%	1/1
15	72.21%	99.90%	20.17%	35.53%	2/3
16	85.80%	99.90%	6.27%	14.13%	1/1
17	19.61%	99.90%	27.10%	22.05%	2/2
18	88.63%	99.90%	21.50%	39.42%	1/1
19	79.44%	99.90%	21.24%	37.90%	2/2
20	48.79%	99.90%	16.38%	27.23%	2/5
26	30.59%	99.90%	1.78%	4.09%	1/1
27	85.20%	99.90%	28.81%	47.78%	2/2
28	82.96%	99.90%	23.58%	41.32%	4/5
29	86.40%	99.90%	68.76%	78.36%	3/3
30	63.81%	99.90%	7.01%	15.04%	1/1
31	63.90%	99.90%	22.78%	37.10%	2/2
32	86.62%	99.90%	6.07%	13.73%	1/1
33	91.21%	99.90%	90.03%	90.73%	7/7
34	83.79%	99.90%	10.60%	22.28%	1/1
35	83.21%	99.90%	20.03%	36.79%	1/1
36	41.42%	99.90%	10.79%	19.40%	4/4
37	100.00 %	99.90%	35.80%	58.23%	1/1
38	86.42%	99.90%	60.48%	73.76%	5/5
39	84.46%	99.90%	18.22%	34.41%	2/2
40	83.47%	99.90%	6.08%	13.70%	1/1
41	86.99%	99.90%	60.76%	74.18%	1/1

42	91.21%	99.90%	15.09%	30.22%	2/3
43	89.20%	99.90%	8.05%	17.73%	1/1
44	94.08%	99.90%	82.07%	88.88%	1/1
45	70.17%	99.90%	6.70%	14.65%	1/1
46	73.92%	99.90%	10.81%	22.16%	2/2
47	81.08%	99.90%	7.91%	17.24%	1/1
48	11.28%	99.90%	6.84%	8.95%	1/1
49	90.51%	99.90%	40.97%	61.00%	2/2
50	93.20%	99.90%	18.78%	36.05%	2/2

Table A.6: Results of the second dataset using the sequential system. The quantitative measures sensitivity, specificity, Dice and F_2-score are measured. The qualitative results show the number of nodules correctly predicted in relation to the total number of nodules.

Model 2 - Sequential system (S2)	
Patients with no presence of nodules	False positives
21	10816
22	35730
23	26359
24	22170
25	6728
51	34473
52	31038
53	38037
54	25902
55	72625

Table A.7: The test data contains ten patients with no presence of nodules. This table shows the number of false positives predicted by the sequential system.

Model 3 - Single system (S1)					
	Quantitative				Qualitative
Patient	Sensitivity	Specificity	Dice	F2-score	nodules
1	75.26%	99.90%	79.94%	77.07%	3/3
2	53.57%	99.90%	52.81%	53.26%	3/3
3	75.56%	99.90%	72.92%	74.48%	4/4
4	79.19%	99.90%	72.50%	76.37%	3/3
5	54.49%	99.90%	66.31%	58.68%	2/2
6	68.96%	99.90%	19.31%	34.00%	1/1
7	88.96%	99.90%	30.50%	50.36%	1/1
8	74.10%	99.90%	39.99%	55.25%	4/5
9	92.22%	99.90%	93.24%	92.63%	1/1
10	99.80%	99.90%	41.83%	64.21%	1/1
11	88.20%	99.90%	66.66%	78.11%	1/1
12	84.20%	99.90%	75.19%	80.35%	4/4
13	77.90%	99.90%	21.63%	38.17%	2/2

14	53.91%	99.90%	58.07%	55.50%	1/1
15	16.65%	99.90%	14.96%	15.93%	2/3
16	83.85%	99.90%	18 %	34.04%	1/1
17	11.92%	99.90%	19.85%	14.19%	1/2
18	86.84%	99.90%	43.72%	62.14%	1/1
19	61.31%	99.90%	43.82%	52.87%	2/2
20	81.15%	99.90%	64.20%	73.40%	5/5
26	5.90%	99.90%	2.21%	3.54%	1/1
27	81.59%	99.90%	45.85%	62.20%	2/2
28	59.03%	99.90%	37.16%	47.78%	5/5
29	90.17%	99.90%	87.66%	89.15%	3/3
30	0.00%	99.90%	0.00%	0.00%	0/1
31	4.24%	99.90%	5.92%	4.78%	1/2
32	83.02%	99.90%	89.04%	85.33%	1/1
33	70.95%	99.90%	80.89%	74.62%	6/7
34	62.76%	99.90%	58.40%	60.94%	1/1
35	1.45%	99.90%	1.14%	1.31%	1/1
36	54.27%	99.90%	53.81%	54.08%	3/4
37	92.23%	99.90%	82.18%	87.93%	1/1
38	74.69%	99.90%	70.66%	73.02%	5/5
39	8.13%	99.90%	8.32%	8.20%	1/2
40	94.95%	99.90%	74.17%	85.38%	1/1
41	82.06%	99.90%	78.33%	80.53%	1/1
42	86.06%	99.90%	54.89%	70.13%	3/3
43	38.64%	99.90%	28.78%	33.98%	1/1
44	92.59%	99.90%	94.25%	93.25%	1/1
45	0.00%	99.90%	0.00%	0.00%	0/1
46	70.73%	99.90%	27.56%	43.49%	2/2
47	53.38%	99.90%	17.81%	29.68%	1/1
48	0.89%	99.90%	1.37%	1.04%	0/1
49	73.57%	99.90%	43.66%	57.75%	2/2
50	92.20%	99.90%	31.20%	51.74%	2/2
56	69.98%	99.90%	15.25%	28.73%	1/1
57	64.32%	99.90%	61.46%	63.14%	2/2
58	46.71%	99.90%	41.96%	44.69%	2/2
59	36.23%	99.90%	51.51%	41.11%	1/1
60	0.12%	99.90%	0.04%	0.07%	0/1
61	70.99%	99.90%	70.55%	70.81%	1/1
62	15.31%	99.90%	25.57%	18.24%	1/2
63	40.52%	99.90%	49.35%	43.64%	1/3
64	6.15%	99.90%	10.07%	7.29%	1/2
65	16.38%	99.90%	24.89%	18.98%	2/2
66	14.67%	99.90%	21.41%	16.78%	1/2
67	17.04%	99.90%	1.67%	3.64%	1/3
68	88.73%	99.90%	52.22%	69.34%	1/1
69	1.74%	99.90%	1.42%	1.60%	1/3
70	36.12%	99.90%	10.18%	17.89%	1/2
71	39.83%	99.90%	40.97%	40.28%	1/1
72	55.05%	99.90%	10.53%	20.46%	1/2
73	47.88%	99.90%	61.54%	52.54%	1/2

74	0.00%	99.90%	0.00%	0.00%	0/2
75	51.94%	99.90%	44.32%	48.60%	1/1
76	4.70%	99.90%	1.78%	2.84%	1/1
77	50 %	99.90%	10.17%	19.48%	1/1
78	77.44%	99.90%	80.66%	78.70%	7/7
79	91.32%	99.90%	48.24%	67.28%	2/2
80	95.51%	99.90%	44.57%	65.55%	1/1
81	1.27%	99.90%	2.32%	1.55%	1/4
82	63.65%	99.90%	72.84%	67.03%	5/6
83	69.77%	99.90%	32.57%	47.89%	1/1
84	97.73%	99.90%	17.09%	33.85%	1/1
85	96.57%	99.90%	13.89%	28.57%	1/1
86	76.88%	99.90%	78.32%	77.45%	1/1
87	69.21%	99.90%	30.54%	45.94%	4/4
88	74.67%	99.90%	13.89%	27.14%	2/2
89	81.65%	99.90%	81.03%	81.40%	4/4
90	0.00%	99.90%	0.00%	0.00%	0/1
91	77.35%	99.90%	84.23%	79.96%	3/3
92	0.00%	99.90%	0.00%	0.00%	0/1
93	23.60%	99.90%	37.66%	27.75%	4/6
94	74.23%	99.90%	82.75%	77.42%	2/2
95	28.42%	99.90%	35.11%	30.76%	1/1
96	39.36%	99.90%	11.71%	20.25%	2/2
97	50.54%	99.90%	31.55%	40.73%	2/2
98	0.00%	99.90%	0.00%	0.00%	0/1
99	41.77%	99.90%	6.47%	13.12%	1/2
100	0 %	99.90%	0 %	0 %	0/1
101	10.97%	99.90%	11.90%	11.33%	1/2
102	63.40%	99.90%	2.56%	6.03%	2/2
103	32.08%	99.90%	32.84%	32.38%	2/4
104	82.86%	99.90%	54.38%	68.50%	1/1
105	78.12%	99.90%	14.52%	28.39%	1/1
106	46.99%	99.90%	3.69%	8.25%	1/1
107	17.44%	99.90%	11.41%	14.40%	2/2
108	39.05%	99.90%	48.71%	42.42%	1/1
109	90.77%	99.90%	3.20%	7.60%	1/1
110	79.70%	99.90%	12.30%	24.97%	1/1

Table A.8: Results of the third dataset using the single system. The quantitative measures sensitivity, specificity, Dice and F_2-score are measured. The qualitative results show the number of nodules correctly predicted in relation to the total number of nodules.

Model 3 - Single system (S1)	
Patients with no presence of nodules	False positives
21	936
22	1771
23	5504
24	10889

25	3135
51	8791
52	166
53	1655
54	2031
55	1151
111	3246
112	25563
113	15156
114	19
115	5

Table A.9: The test data contains fifteen patients with no presence of nodules. This table shows the number of false positives predicted by the single system.

Model 3 - Sequential system (S2)					
	Quantitative				Qualitative
Patient	Sensitivity	Specificity	Dice	F2-score	nodules
1	75.12%	99.90%	80.90%	77.33%	3/3
2	83.96%	99.90%	51.97%	6.37%	3/3
3	71.78%	99.90%	47.52%	59.61%	4/4
4	75.64%	99.90%	47.71%	61.29%	3/3
5	92.80%	99.90%	81.45%	87.90%	2/2
6	77.14%	99.90%	13.63%	26.93%	1/1
7	94.45%	99.90%	8.34%	18.41%	1/1
8	61.79%	99.90%	8.38%	17.42%	3/5
9	93.52%	99.90%	77.77%	86.51%	1/1
10	99.90%	99.90%	7.41%	16.66%	1/1
11	87.73%	99.90%	64.11%	76.46%	1/1
12	85.97%	99.90%	30.08%	49.31%	3/4
13	98.30%	99.90%	12.36%	26.00%	2/2
14	81.34%	99.90%	30.42%	48.72%	1/1
15	71.33%	99.90%	39.99%	54.30%	2/3
16	86.71%	99.90%	8.74%	18.98%	1/1
17	18.26%	99.90%	25.46%	20.59%	2/2
18	88.80%	99.90%	20.21%	37.67%	1/1
19	79.52%	99.90%	26.85%	44.56%	2/2
20	64.44%	99.90%	35.87%	48.87%	3/5
26	80.14%	99.90%	20.47%	37.00%	1/1
27	84.39%	99.90%	35.63%	54.54%	2/2
28	88.83%	99.90%	33.08%	53.06%	5/5
29	91.83%	99.90%	81.39%	87.35%	3/3
30	3.17%	99.90%	0.56%	1.10%	0/1
31	59.43%	99.90%	51.06%	55.77%	2/2
32	88.35%	99.90%	47.14%	65.46%	1/1
33	88.96%	99.90%	88.99%	88.97%	7/7
34	81.03%	99.90%	39.34%	56.91%	1/1
35	86.39%	99.90%	12.17%	25.11%	1/1

36	79.56%	99.90%	42.05%	58.64%	4/4
37	100.00 %	99.90%	53.03%	73.84%	1/1
38	82.81%	99.90%	68.93%	76.64%	5/5
39	95.14%	99.90%	20.50%	38.73%	2/2
40	85.53%	99.90%	10.63%	22.41%	1/1
41	91.98%	99.90%	66.76%	79.91%	1/1
42	68.56%	99.90%	23.75%	39.08%	3/3
43	83.38%	99.90%	7.04%	15.63%	1/1
44	93.90%	99.90%	87.28%	91.13%	1/1
45	0.00%	99.90%	0.00%	0.00%	0/1
46	68 %	99.90%	16.47%	30.20%	2/2
47	76.35%	99.90%	17.75%	32.90%	1/1
48	55.55%	99.90%	39.40%	47.73%	1/1
49	91.05%	99.90%	11.35%	23.90%	2/2
50	44.40%	99.90%	10.98%	20.03%	2/2
56	0 %	99.90%	0 %	0 %	0/1
57	85.45%	99.90%	22.35%	40.14%	2/2
58	90.32%	99.90%	51.17%	69.16%	2/2
59	92.84%	99.90%	82.84%	88.56%	1/1
60	97.55%	99.90%	15.29%	30.94%	1/1
61	95.35%	99.90%	74.05%	85.51%	1/1
62	56.20%	99.90%	54.70%	55.59%	1/2
63	46.65%	99.90%	26.56%	35.82%	1/3
64	90.70%	99.90%	72.31%	82.33%	2/2
65	0.00%	99.90%	0.00%	0.00%	0/2
66	8.38%	99.90%	5.79%	7.10%	1/2
67	16.40%	99.90%	0.40%	0.97%	1/3
68	97.07%	99.90%	23.45%	43.03%	1/1
69	0.00%	99.90%	0.00%	0.00%	0/3
70	72.34%	99.90%	8.45%	17.98%	2/2
71	77.75%	99.90%	67.48%	73.29%	1/1
72	67.05%	99.90%	14.36%	27.17%	1/2
73	91.34%	99.90%	88.19%	90.05%	1/2
74	0 %	99.90%	0 %	0 %	0/2
75	67.96%	99.90%	13.36%	25.80%	1/1
76	73.50%	99.90%	11.37%	23.07%	1/1
77	77.27%	99.90%	2.74%	6.51%	1/1
78	49.56%	99.90%	32.03%	40.66%	6/7
79	98.06%	99.90%	34.40%	56.35%	2/2
80	97.14%	99.90%	75.80%	87.31%	1/1
81	6.15%	99.90%	10.04%	7.28%	2/4
82	86.02%	99.90%	86.49%	86.20%	6/6
83	87.33%	99.90%	22.04%	39.97%	1/1
84	97.09%	99.90%	7.15%	16.09%	1/1
85	99.43%	99.90%	3.76%	8.89%	1/1
86	89.94%	99.90%	80.74%	86.02%	1/1
87	87.63%	99.90%	21.16%	38.83%	4/4
88	97.19%	99.90%	8.13%	18.05%	2/2
89	71.83%	99.90%	22.82%	38.64%	4/4
90	49 %	99.90%	12.76%	22.94%	1/1

91	89.50%	99.90%	74.07%	82.61%	3/3
92	93.12%	99.90%	2.37%	5.72%	1/1
93	84.64%	99.90%	75.56%	80.76%	5/6
94	89.14%	99.90%	68.04%	79.30%	2/2
95	0.00%	99.90%	0.00%	0.00%	0/1
96	0.00%	99.90%	0.00%	0.00%	0/2
97	65.54%	99.90%	49.49%	58.04%	2/2
98	100.00 %	99.90%	13.67%	28.37%	1/1
99	46.95%	99.90%	5.95%	12.50%	1/2
100	52.87%	99.90%	14.30%	25.44%	1/1
101	17.88%	99.90%	7.75%	11.74%	1/2
102	87.63%	99.90%	3.43%	8.11%	2/2
103	51.69%	99.90%	27.78%	38.45%	2/4
104	98.10%	99.90%	19.22%	37.13%	1/1
105	90.57%	99.90%	19.16%	36.37%	1/1
106	80.00%	99.90%	4.74%	10.88%	1/1
107	16.49%	99.90%	7.08%	10.77%	1/2
108	89.35%	99.90%	42.06%	61.63%	1/1
109	58.46%	99.90%	0.70%	1.72%	1/1
110	90.98%	99.90%	2.35%	5.65%	1/1

Table A.10: Results of the third dataset using the sequential system. The quantitative measures sensitivity, specificity, Dice and F_2-score are measured. The qualitative results show the number of nodules correctly predicted in relation to the total number of nodules.

Model 3 - Sequential system (S2)	
Patients with no presence of nodules	False positives
21	2122
22	22998
23	6983
24	19110
25	14359
51	29120
52	11117
53	10567
54	11525
55	28184
111	3846
112	25201
113	21268
114	5960
115	2028

Table A.11: The test dataset contains fifteen patients with no presence of nodules. This table shows the number of false positives predicted by the sequential system.