FACULTY OF ENGINEERING, UNIVERSITY OF PORTO

Segmentation of Pulmonary Nodules in CT images

Joana Maria Neves da Rocha

Integrated Master in Bioengineering

Supervisor: Ana Maria Mendonça, PhD

Co-Supervisor: António Cunha, PhD

July 19, 2019

Segmentation of Pulmonary Nodules in CT images

Joana Maria Neves da Rocha

Integrated Master in Bioengineering

July 19, 2019

Resumo

Os nódulos pulmonares estão associados a várias patologias, entre as quais o cancro do pulmão, considerado uma das principais causas de morte a nível mundial. A elevada incidência de nódulos pulmonares reforça a importância de uma monitorização contínua da saúde pulmonar dos pacientes, realçando o papel de uma deteção e caracterização antecipadas, consideradas cruciais para permitir intervenções que possivelmente serão determinantes para a sobrevivência do paciente.

Os sistemas de Diagnóstico Assistido por Computador podem auxiliar os médicos nas suas tarefas diárias, facilitando a interpretação dos resultados da imagiologia médica, tais como *scans* de Tomografia Computorizada (TC). Ao longo do processo de monitorização, existe a necessidade de segmentar os nódulos pulmonares, delimitando as suas bordas de forma a localizá-los de forma precisa, bem como caracterizá-los. Diversas abordagens podem ser usadas para cumprir esta tarefa, enfrentando as numerosas adversidades associadas à segmentação de imagens médicas: entre elas, o ruído, o *blur*, e o baixo contraste das imagens, bem como a complexidade e diversidade das estruturas anatómicas, traduzidas pela variedade em tamanho, forma e textura das mesmas. A presente dissertação pretende superar tais adversidades e segmentar com sucesso nódulos pulmonares, através de métodos exemplificativos das técnicas convencionais e baseadas em *Deep Learning*, de modo a comparar o seu desempenho e interpretar qual das mesmas tem maior probabilidade de contribuir para um diagnóstico preciso e atempado. As técnicas convencionais baseiam-se numa determinada representação do conhecimento, deduzindo nova informação através de regras lógicas e de um conjunto de princípios, enquanto que as técnicas de *Deep Learning* executam uma modelação orientada por uma grande quantidade de dados, através dos quais extraem conhecimento.

A abordagem convencional selecionada baseia-se fundamentalmente num Filtro de Convergência Local, devido à sua habilidade para lidar com imagens ruidosas e de baixo contraste; tal filtro maximiza o grau de convergência dos vetores de gradiente dentro de uma região de suporte, em relação a um píxel central. Especificamente, foi aplicado o *Sliding Band Filter* (SBF), devido à sua região de suporte - uma banda de largura fixa cuja posição é ajustada em cada direção radial -, que lhe garante flexibilidade em termos de forma e uma resposta seletiva ao ignorar o comportamento do gradiente no centro do objeto. O algoritmo convencional estima o centro do nódulo, obtém os respetivos pontos de suporte (coordenadas do bordo), e refina o resultado para obter a máscara de segmentação.

As metodologias baseadas em *Deep Learning* interpretam esta tarefa de segmentação semântica píxel-a-píxel como um exercício de classificação binária, no qual se decide quais as coordenadas da imagem que pertencem ou não à lesão. Neste trabalho, foram implementadas duas redes *fully convolutional*, cujas arquiteturas *encoder-decoder* fazem respetivamente *downsampling* e *upsampling* dos mapas de características, criando simetria nas arquiteturas. O primeiro modelo implementado foi a *U-Net*, desenvolvido inicialmente para segmentação biomédica, e portanto capaz de lidar com o tamanho alargado das imagens, bem como a quantidade limitada de dados de treino anotados. Com as *skip connections*, esta rede permite a combinação de informação contextual capturada no *encoder* com os mapas de características no *decoder*, e portanto obtém um conhecimento detalhado a nível espacial. A segunda rede trata-se de um híbrido entre a *U-Net* e a *SegNet*, designado por *SegU-Net*, que se baseia na estrutura da *U-Net* e substitui o seu método de *upsampling* pelo da *SegNet*, idealmente promovendo a eficiência da rede ao captar mais precisamente a informação espacial.

As técnicas propostas foram testadas e comparadas com base em imagens extraídas de TCs pertences à base de dados LIDC, tendo como referência as respetivas lesões segmentadas por radiologistas. A abordagem baseada no SBF foi testada num conjunto de 2653 nódulos, e as abordagens de *Deep Learning* em subconjuntos de 531 nódulos (20% da base de dados); no entanto, para estabelecer uma comparação justa, o mesmo conjunto de 531 nódulos foi tido em consideração posteriormente. A *U-Net* foi a metodologia mais eficiente para a segmentação de nódulos pulmonares, imediamente seguida da *SegU-Net*, com valores semelhantes, mas ligeiramente inferiores. Nódulos bem circunscritos, com margens definidas e/ou textura sólida são segmentados com sucesso pelas três abordagens, que tendem a falhar com lesões não-sólidas e/ou com forma irregular. A abordagem convencional exibiu o menor desempenho, sendo penalizada pela sua fraca segmentação de nódulos justapleurais, em comparação às segmentações obtidas pelas técnicas de *Deep Learning*. O método de *upsampling* da *SegU-Net* não melhorou os resultados conseguidos pela *U-Net*, cujo modelo superou vários algoritmos do estado da arte apresentados nesta dissertação. Medidas adicionais podem ser incorporadas no futuro para melhorar os algoritmos propostos, nomeadamente desenvolver um pré-processamento e pós-processamento mais adequados, ou adicionar mais exemplos ao conjunto de treino para promover uma aprendizagem superior.

A comparação entre métodos convencionais e baseados em *Deep Learning* explora as vantagens e desvantagens de cada técnica, definindo a *U-Net* como o método mais eficiente na tarefa em questão - particularmente eficiente para lesões óbvias, e capaz de ultrapassar até certo ponto a elevada variabilidade das imagens. Consequentemente, os resultados satisfatórios da segmentação conseguidos nesta dissertação levam a uma perceção mais compreensiva das características dos nódulos, contribuindo assim para o desenvolvimento de um sistema de apoio à decisão, que poderá ser capaz de assistir os especialistas a estabelecer um diagnóstico fidedigno de patologias pulmonares baseando-se na análise dessas mesmas características.

Abstract

Pulmonary nodules can be associated with several pathologies, such as lung cancer, which is the one of the main causes of death globally. The high incidence of pulmonary nodules reinforces the importance of a continuous monitoring of the patients' lung health, thus emphasizing the role of an early detection and characterization, considered crucial to enable possibly life-saving interventions.

Computer-Aided Diagnosis systems can assist the physicians in their daily tasks, by helping them interpret medical imaging results, such as Computed Tomography (CT) scans. Through out the monitoring process, there is a need to segment the pulmonary nodules, thus defining their boundaries as a way to not only precisely locate them, but also characterize them. Several approaches can be implemented to fulfill this task, facing the numerous challenges that come with a medical image segmentation exercise: among them, the noise, blur and low contrast of the images, and the overall complexity and diversity of anatomical structures, expressed by a variability in size, shape, and texture. This work seeks to overcome these challenges and achieve good results for pulmonary nodule segmentation by employing methods which illustrate conventional and Deep Learning based approaches, in order to compare their performance, and find which one is most likely to possibly contribute to a more accurate and early diagnosis. While conventional techniques are based on knowledge representation, deducing new information using logical rules and a set of beliefs, Deep Learning approaches carry out data-driven modeling, by observing a large amount of data to extract knowledge from it.

The selected conventional approach is mostly based on a Local Convergence Filter, due to its ability to deal with noisy and low-contrast images; this filter maximizes the convergence degree of the gradient vectors within a support region toward a central pixel of interest. More specifically, the Sliding Band Filter (SBF) was employed because of its support region - a band of fixed width whose position is adjusted in each radial direction -, that grants this filter shape flexibility and a selective response by ignoring the behavior of the gradient at the center of the object. The conventional algorithm estimates the center of nodule, gets the corresponding support points (matching the border coordinates), and then refines the result to achieve a segmentation mask.

The Deep Learning based methodologies view this pixel-wise semantic segmentation task as a binary classification exercise, deciding which coordinates of the image belong to the lesion. In this work, two Fully Convolutional Networks were used, whose encoder-decoder structures respectively downsample and upsample the feature maps, and yield symmetric architectures. The first model was the U-Net, developed initially for biomedical segmentation, and thus able to tackle the large size of medical images, as well as the limited amount of annotated training data. Through skip connections, this network allows the combination of contextual information captured in the encoder with the upsampled feature maps in the decoder, thus obtaining more comprehensive spatial knowledge. The second implementation is a hybrid between the U-Net and the SegNet, designated by SegU-Net, which takes advantage of the U-Net's structure and replaces its upsampling method (upconvolutions) by the SegNet's max unpooling layers, ideally enhancing the network's performance by improving the acquired spatial knowledge.

The proposed techniques were tested and compared based on images extracted from CT scans which belong to the LIDC database, minding the corresponding segmented annotations made by radiologists as ground truth.The SBF based approach was tested on a set of 2653 nodules, while the U-Net and the SegU-Net were tested on 531 nodules (20% of the data), but in order to establish a fair comparison, the same nodule set was taken into consideration later on. One can conclude that the U-Net was the method with the most efficient pulmonary nodule segmentation masks, immediately followed by the SegU-Net, with similar scores. Well-circumscribed obvious nodules, with sharp margins and/or solid texture are successfully segmented by all three approaches, which tend to fail with non-solid or irregular shaped lesions. The conventional approach exhibited the lowest performance scores, because of its poor segmentation of juxtapleural nodules, in comparison to the segmentation achieved by the Deep Learning algorithms. The SegU-Net's upsampling method did not improve the results achieved by the U-Net, whose model has successfully outperformed state of the art algorithms presented in this dissertation. Additional efforts can be done in future work to improve the algorithms' performance, namely develop more adequate pre-processing and post-processing steps, or add more examples to the training set, promoting a superior and perceptive learning.

The comparison of the conventional and Deep Learning based approaches explored the advantages and disadvantages of each technique, establishing the U-Net as the most efficient method in this case - particularly efficient for obvious lesions, and able to overcome to a certain extent the high variability of the images. Consequently, the satisfactory segmentation results achieved in this work lead to further insights on nodule characterization, contributing to the development of a decision support system, which may be able to assist the physicians to establish a reliable diagnosis of lung pathologies based on the analysis of such characteristics.

Acknowledgements

I am fortunate to say that I love what I do, but I am also very fortunate to have shared my day-to-day with people who made this experience ever more enjoyable, and whom I want to thank deeply.

Foremost, I would like to acknowledge Professor Ana Maria Mendonça for her guidance, knowledge, and permanent availability, and thank her for having an open door to her office since the beginning. I am honored to have her as my supervisor, whose input in this work was essential for me to grow as a researcher.

I would also like to acknowledge Professor António Cunha for his constant willingness to help, and all the words of incentive he transmitted throughout my Master thesis. His enthusiasm about Deep Learning motivates everyone and made me even more curious to learn and develop new skills.

I am grateful to all of those with whom I have had the pleasure to work with at INESC-TEC's Biomedical Imaging Lab, during the past year. In particular, a heartfelt thank you to Carlos Ferreira, Guilherme Aresta, and Tânia Melo, who made me feel very welcome in the lab and were always there to give me great advice.

Last but not least, a very special gratitude to my mother for her moral and emotional support, and for always providing me unfailing continuous encouragement throughout my studies. I would not be where I am without her.

Contents

List of Figures

List of Tables

Abbreviations

2D	Bidimensional
3D	Tridimensional
ADAM	Adaptive Moment Estimation
ANN	Artificial Neural Network
ARF	Adaptive Ring Filter
CAD	Computer-Aided Diagnosis
CF-CNN	Central-Focused Convolutional Neural Network
CNN	Convolutional Neural Network
CT	Computed Tomography
DASL	Deep Active Self-Paced Learning
FC	Fully Connected
FCN	Fully Convolutional Network
FN	False Negative
FP	False Positive
FPN	Feature Pyramid Network
GAP	Global Average Pooling
GCLM	Gray-Level Co-Occurrence Matrix
GGO	Ground Glass Opacity
GPU	Graphics Processing Unit
HU	Hounsfield Unit
IF	IRIS Filter
kNN	k-Nearest Neighbors
LCF	Local Convergence Filter
LIDC	Lung Image Database Consortium
MRI	Magnetic Resonance Imaging
NAM	Nodule Activation Map
PDF	Probability Density Function
R-CNN	Region-based Convolutional Neural Network
ReLU	Rectified Linear Unit
ResNet	Residual Network
R-NAM	Residual Nodule Activation Map
ROI	Region Of Interest
RPN	Region Proposal Network
SBF	Sliding Band Filter
SDAE	Stacked Denoising Autoencoder
SGD	Stochastic Gradient Descent
SOI	Structure of Interest
tanh	Hyperbolic Tangent

TN	True Negative
TP	True Positive
VOI	Volume of Interest

Chapter 1

Introduction

Due to their high clinical importance, the detection and characterization of pulmonary nodules have been a constant focus of attention, and can be a key to assess the patient's health condition. The advances of technology, and imaging techniques in particular, have improved nodule identification - Ost et al. (2003) estimate that 150,000 solitary pulmonary nodules are identified each year. A solitary pulmonary nodule is a rounded well-marginated opacity with a diameter up to 3 cm that is completely surrounded by lung parenchyma and is not related to lymphadenopathy, atelectasis, or pneumonia, as stated in Winer-Muram (2006). These can be classified as benign or malignant. Even though larger nodules tend to be malignant, their size cannot be directly correlated to their classification. This means that after detecting a pulmonary nodule, it is important to consider several aspects, including the patient's clinical history and nodule shape, to establish a diagnosis.

1.1 Pulmonary Nodules and Lung Cancer

Pulmonary nodules can be associated with several diseases, but a recurrent diagnosis is lung cancer, which is the main cause of cancer death in men and the second cause in women worldwide, according to Torre et al. (2016). For this reason, providing an early detection and diagnosis to the patient is extremely crucial, especially considering that any delay in cancer detection might result in lack of treatment efficacy, and therefore decrease the survival probability. Follow-up exams are critical as well - Winer-Muram (2006) states that a nodule is considered benign if the follow-up computed tomography (CT) exams show no growth in 2 years.

Medical imaging has proved to be a valuable asset, as it promotes an accurate early detection of the nodules. There are many medical imaging techniques, among them Radiography, Magnetic Resonance Imaging (MRI), Ultrasound, and Tomography. This work will be implemented and tested using images of lung CT scans. This type of scans is built based on cross-sectional tomographic images (which are often designated as "slices"). A 3D volume of the object (in this case, the lungs) is generated with digital image processing, by stacking all the slices on top of each other.

The use of CT scan images, which are in 3D, can be beneficial in comparison to 2D medical images: they eliminate the superimposition of structures outside the region of interest, as stated in Coche et al. (2011), and they allow the reconstruction of the collected data to be analyzed in three planes (axial, coronal, and sagittal). An example of a CT scan slice is presented in Figure 1.1.

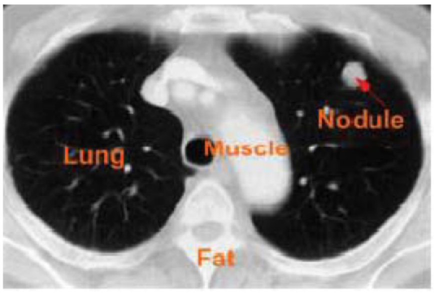


Figure 1.1: Lung CT image slice.
Source: Kamra et al. (2015).

1.2 Pulmonary Nodule Segmentation

Computer-Aided Diagnosis (CAD) systems for pulmonary nodule detection and segmentation can help the physician to make the diagnosis based on the analysis of medical images, such as CT scans. Moltz et al. (2009) explain that manual nodule examinations are error-prone (in part because of their subjectivity) and time-consuming, and for these reasons CAD systems are beneficial and relevant, since they aim to intrinsically automate the segmentation task, and other tasks as well.

In a CAD system with a typical organization, four stages are defined: pre-processing, detection, segmentation, and classification. In order to study lung nodules in CT images, the first step would be a pre-processing stage, where the system intends to improve the medical image quality, reduce noise and identify the region of interest (ROI).

This step is followed by a detection stage, crucial in a CAD system, in which the goal is to enhance the lesions and identify the voxels where they are likely to be in. There are several methods that can be applied to detect lung nodules, but this step usually results in a substantial amount of false positives. A false positive happens when the system indicates that there is a nodule present in a specific lung area, but in fact that area does not contain a lesion. Ideally, the false positive rate should be reduced, but that could mean a decrease in the detection of real nodules as well. Instead, as a nodule cannot remain undetected, considering it would have a very serious negative impact in the patient's health condition, the CAD system focuses more on reducing the false negative rate.

Segmentation is described in Kamra et al. (2015) as the process of differentiating the nodule from other structures, and is one of the main steps in a CAD system. Litjens et al. (2017) add that segmentation aims to select the set of voxels of a CT scan that belong to the borders or the interior of the lesion. However, Wang et al. (2017) confirm that this task is quite complex and can be challenging considering the heterogeneity of the nodule size and shape and the fact that the nodule's intensity can vary within its borders. Besides that, it becomes even more difficult to establish the perimeter of the nodule when it is next to other structures, such as blood vessels or the lung wall. The overall segmentation difficulty increases due to data imbalance - in a CT scan, less than 5% of the voxels belong to these lesions.

There are several techniques when it comes to nodule classification. These can be differentiated into 2D or 3D techniques - the first consider 2D section images ("slice by slice") and the last use 3D images. The 3D techniques are usually better because they avoid loss of information and reduce the number of false positives. Naseem et al. (2017) state that while both nodules and blood vessels appear to be spherical in the 2D images, in 3D the vessels' morphology resembles a tree structure, easing the differentiation between distinct tissues. However, when it comes to segmentation tasks, 2D results are usually easier to visualize and interpret.

1.2.1 Conventional vs. Deep Learning Techniques

When it comes to biomedical image analysis, early methods (generally described as conventional) consisted of following a sequence of image processing steps (e.g. edge/line detectors, region growing) and mathematical models, as described in Litjens et al. (2017). Conventional techniques can also use the training data to elaborate a model (e.g. active shapes). Afterwards, the idea of extracting features and feeding them to a statistical classifier made supervised techniques become a trend. Mohri et al. (2018) explains that these Machine Learning techniques are called supervised because the model infers a function from labeled (annotated) training data, consisting of a set of training examples.

Furthermore, Litjens et al. (2017) add that, more recently, the trend is to develop models that are able to interpret what features better represent the data. Deep Learning is based on these multi-layered models that learn about the input images and predict the outcomes. In particular, Convolutional Neural Networks (CNNs) are often implemented for biomedical image analysis. The flowchart in Figure 1.2 represents an example of a conventional image segmentation technique, where the nodule segmentation is achieved through the use of a Watershed segmentation approach. On the other hand, it is possible to exemplify a Deep Learning technique. Figure 1.3 represents a 3D CNN structure. In this approach, an unsupervised segmentation method is used on a set of images. These segmentation results will then train the CNN, so that the structure is able to predict the probability of a specific voxel belonging to the lesion or not. The top row of Figure 1.3 shows how the ground truth images are obtained (using the unsupervised segmentation method), which will be fed into the multi-layered CNN represented in the bottom row of the figure.

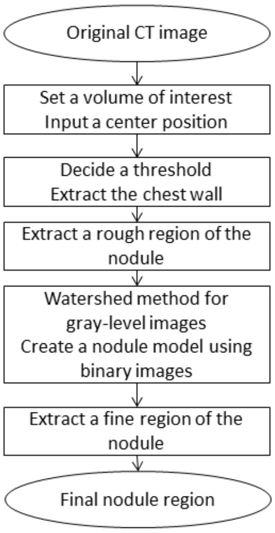

Original CT image
Set a volume of interest
Input a center position
Decide a threshold
Extract the chest wall
Extract a rough region of the nodule
Watershed method for gray-level images
Create a nodule model using binary images
Extract a fine region of the nodule
Final nodule region

Figure 1.2: Example of a conventional approach for pulmonary nodule segmentation. Source: Tachibana and Kido (2006).

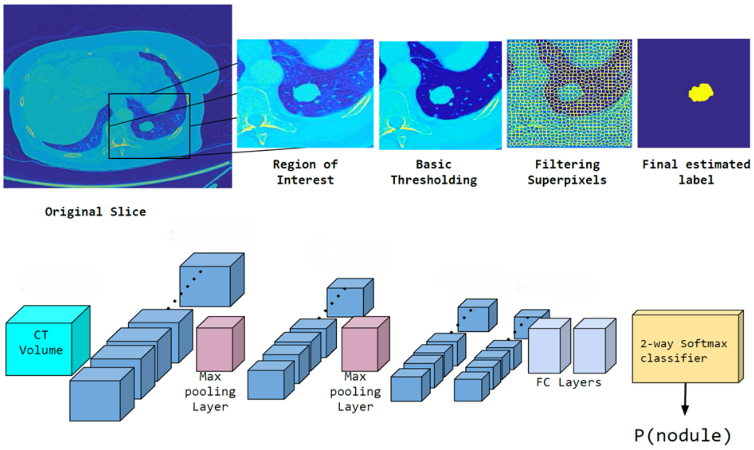


Figure 1.3: Example of a Deep Learning approach for pulmonary nodule segmentation. Source: Anirudh et al. (2016).

Deep Learning approaches require less implementation time in comparison to conventional ones, because, as stated in Cheng et al. (2016), feature extraction and selection are done automatically within the model. Furthermore, the authors mention that Deep Learning is less prone to errors, as it is not influenced by a possible defective image processing or feature extraction. In spite of that, it is important to understand that Deep Learning also has disadvantages. It requires a large amount of annotated data (otherwise it is unlikely to outperform the conventional techniques), it has large computational cost, and it may be difficult to interpret the model's decisions or to tune the hyperparameters. However, recent systems have started to address the interpretation and hyperparameter tuning issues.

1.3 Motivation

The importance of an early lung cancer diagnosis induces the need to develop accurate methods to detect, characterize, and precisely locate pulmonary nodules, thus promoting the high clinical relevance of pulmonary nodule segmentation tasks. For this reason, medical imaging resorts to CT scans, which among other exams help the physicians to give a more accurate early diagnosis, through a clinical decision support system.

Both conventional and Deep Learning based techniques have achieved a good reputation when it comes to solving biomedical imaging problems, and segmentation tasks in particular. They both have advantages and disadvantages, explored in this work, whose goal is to segment pulmonary nodules. Ultimately, such task contributes to the diagnosis of the associated lung pathology through the precise location and characterization of the nodules.

In order to define the best approach for the pulmonary nodule segmentation task, conventional and Deep Learning based approaches are implemented, and their performance is evaluated, by comparing each one to a segmentation ground truth reference, given by the specialists.

1.4 Dissertation Structure

This document comprises eight chapters. In the present one, an introduction on the studied subject is made, explaining the motivation behind pulmonary nodule segmentation and the relevance of CAD systems. The goal of this work is defined - it is intended to compare both conventional and Deep Learning based techniques in the mentioned segmentation task. In Chapter 2, the state of the art is described for both topics, including a brief explanation about the characteristics of nodular lesions. Chapter 3 presents the fundamentals about Deep Learning, while Chapter 4 details the LIDC database, whose CT scans are used in the proposed methodologies, and also provides the segmentation ground truth. The implemented techniques are outlined in Chapters 5 and 6, describing the conventional and Deep Learning techniques respectively. The results are presented and discussed in Chapter 7. Chapter 8 is the final chapter, where the main conclusions are drawn.

Chapter 2

State of the Art Review for Pulmonary Nodule Segmentation

CAD systems are useful tools to support the physicians on their diagnosis, and so they have been used for the analysis of several types of biomedical images, such as MRI, X-rays, CT scans, and Ultrasound. They are considered a "second opinion" that helps the physician interpret medical images. Chest radiography and CT scans are commonly used in medical exams to analyze and evaluate the patient's lungs. The results obtained by the CAD scheme help the physician make a more accurate diagnosis in a shorter amount of time, because the system is ideally able to detect and segment pulmonary nodules. Nodule segmentation intends to delineate the nodule's perimeter, which can be difficult to achieve, considering that medical image segmentation has some inherent challenges that might not be easy to overcome: these images are often noisy, blurred and low contrasted. Besides that, the variability and complexity of the anatomic structures, translated by an arbitrariness in shape, size and texture, make this task laborious. Data imbalance also poses a threat to a successful segmentation step, as the ratio between the number of voxels that belong to a lesion and the total amount of voxels in an image is usually low.

2.1 Introduction on Pulmonary Nodule Characterization

It is important to discuss nodule characterization, in order to better understand the obstacles faced by medical segmentation algorithms. The characterization of a nodule can be achieved through the specific visual or quantitative features of a CT scan, as stated in Bartholmai et al. (2015), and ultimately facilitates the classification of that nodule as benign or malignant. The several visual features that may be recognized in a CT scan include size, attenuation, location, morphology, and edge characteristics, as well as other distinctive "signs" that can be highly suggestive of a specific diagnosis. Besides that, controlling any changes in size, attenuation, and morphology in follow-up CT scans is also of crucial importance.

A pulmonary nodule can belong to one of four main classes, according to their location:

(a) Well-circumscribed
(b) Vascularized
(c) Pleural tail
(d) Juxtapleural

Citing Kostis et al. (2003), well-circumscribed nodules have a central location within the lung and have no considerable connection to vasculature, as shown in Figure 2.2a. On the other hand, vascularized nodules also have a central location, but are connected to nearby vessels. If a nodule is near the pleural surface and is connected to it by a thin structure, it is considered a pleural tail nodule. Finally, juxtapleural nodules are attached to the chest wall and pleural surface (Figure 2.2b). Wang et al. (2017) also mention other nodule classes, such as cavitary nodules and calcific nodules. Hansell et al. (2008) describe a cavity as a gas-filled space within the nodule, which means that a cavitary nodule has a lower density area within itself, seen in the CT scan as a black hole (Figure 2.2c). On the contrary, the calcified regions within the nodules have higher intensity values. These regions present different patterns of calcification, consequently having diverse shapes. Khan et al. (2010) state that calcific nodules are likely to be benign, but not all of them are. An example of a calcific nodule is shown in Figure 2.2d.

According to Henschke et al. (2002), nodules can also be differentiated as non-solid, part-solid, and solid, referring to their texture/density (Figure 2.1). The authors define solid nodule as one that "completely obscures the lung parenchyma within them". On the other hand, non-solid nodules, also known as ground glass opacities (GGOs), "do not completely obscure the lung parenchyma within them". A GGO is represented in Figure 2.2e. A sub-solid nodule can be non-solid or part-solid (i.e. the nodule obscures certain spots of the lung parenchyma).

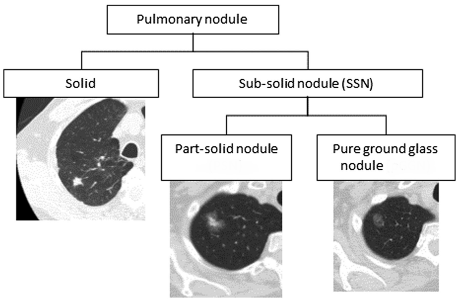


Figure 2.1: Pulmonary nodule classification according to their texture.
Source: Baldwin and Callister (2015).

As mentioned earlier, pulmonary nodule segmentation is a challenging task. This happens partly because nodules often have similar intensity to the neighbor regions - e.g. the intensity of a juxtapleural nodule intensity is identical to the intensity of the lung wall. Furthermore, both cavitary and calcific nodules are difficult to segment due to their lack of structural uniformity (different intensities within their boundaries). Ground glass opacities exhibit low contrast in CT scans, and do not have clear boundaries, which hinders the segmentation.

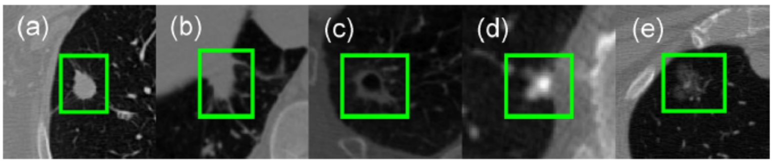


Figure 2.2: Different nodule types present in CT images.
Source: Wang et al. (2017).

In spite of all the difficulties, segmentation is considered to be a crucial and valuable part of the CAD scheme, and that is why efforts have been and are currently being made to reach good nodule segmentation results.

2.2 Conventional Segmentation Techniques

Conventional image analysis techniques have been around longer than Deep Learning, and even though Deep Learning has become a very popular approach recently, conventional image analysis is not considered obsolete. Conventional segmentation techniques can be implemented using diverse methods - among the most common ones are thresholding, histogram-based methods, clustering, region growing, graph partitioning (e.g. Markov Random Fields), and Watershed. These may need longer implementation time in comparison to Deep Learning techniques, but in most situations have proved to perform as expected. This way, some of the most relevant approaches for pulmonary nodule segmentation are described in this section.

Lesion detection and segmentation often imply the use of filters, as suggested in Pereira et al. (2007a), where the Sliding Band Filter (SBF) is used to develop an automated method that identifies lung nodule candidates in chest radiography. It was proposed to multiply the filtered image by a mask, to get the probability of that voxel being part of the nodule or not. Then, to obtain a set of candidates present in non-overlapping areas, the authors applied the Watershed segmentation method. The results of the main steps of this algorithm are represented in Figure 2.3. According to Pereira et al. (2007b), the SBF proved to perform better than other convergence index filters, when it comes to improving image contrast and detecting pulmonary nodules.

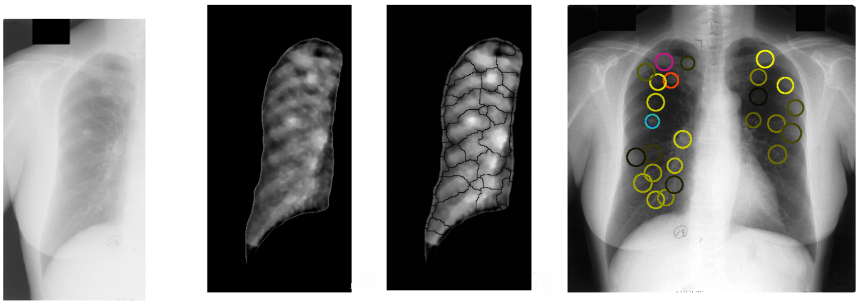


Figure 2.3: Original chest radiograph, and corresponding SBF output, Watershed segmentation result, and final detected nodule candidates.
Source: Pereira et al. (2007a).

Wang et al. (2007) applied a spiral scanning technique to simplify the segmentation task in 3D CT images, by combining 3D and 2D image analysis strategies. Here, the discrete surface of the 3D volume of interest was converted into a 2D image; the goal was to optimize the outline of the nodule. After achieving that, the outline was transformed into a 3D image again, representing the optimal boundaries of the nodule. However, more relevant approaches have been implemented since, some of them which are detailed below.

An example of a 3D semiautomatic segmentation technique is the approach developed by Diciotti et al. (2008), for spiral CT scans, represented in Figure 2.4. First, the user will assign a label to each detection made (nodular, structure or object to reject). This information will then be useful in the following 3D segmentation stage. A threshold is applied to the image, whose value will decrease with each iteration, and as it does so, a morphological opening with a spherical structuring element is employed, so it is possible to disconnect regions that were wrongfully joined. With each iteration, the mean magnitude gradient is calculated. The structure's shape is taken into account because each voxel will be assigned to the nearest connected region, which can be nodular or not (geodesic influence). Gray-level similarity is also considered when it comes to assigning a voxel. This algorithm also generates a different segmentation if the user does not consider the presented one acceptable. Otherwise, the final nodular segmentation will be the one that results in the highest mean magnitude gradient.

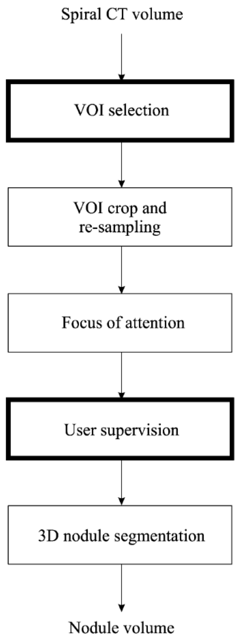


Figure 2.4: Flowchart of the semiautomatic algorithm. VOI stands for Volume of Interest. Source: Diciotti et al. (2008).

In order to obtain segmented pulmonary nodules in 3D CT scan images, Dehmeshki et al. (2008) developed a contrast based region growing method (Figure 2.5), that works based on a mask that finds the optimum seed point inside a nodule. That mask is the result of a local adaptive segmentation algorithm that differentiates between background and foreground points and creates a fuzzy connectivity map of the lesion. The initial seed point inside the nodule can be selected by the user or the CAD system, and the local adaptive segmentation algorithm focuses on local contrast. As seen in Diciotti et al. (2008), previously presented in this section, if the user does not agree with the presented segmentation, other methods are implemented until the result is satisfactory. Having the fuzzy connectivity map, sphericity oriented contrast region growing is applied to generate multiple segmentation results.

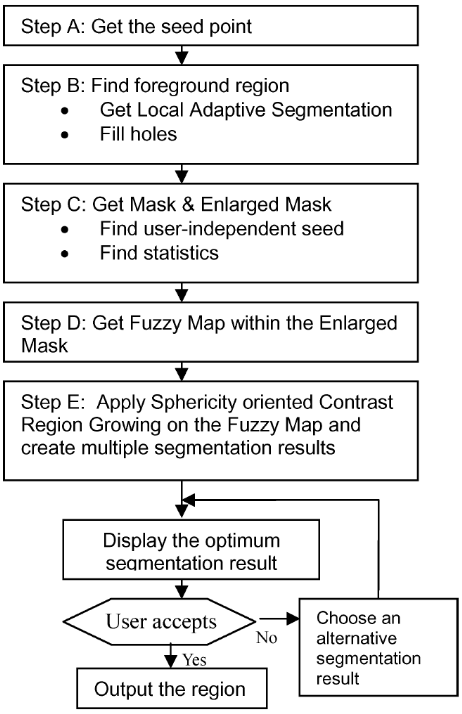

Step A: Get the seed point
Step B: Find foreground region
• Get Local Adaptive Segmentation
• Fill holes
Step C: Get Mask & Enlarged Mask
• Find user-independent seed
• Find statistics
Step D: Get Fuzzy Map within the Enlarged Mask
Step E: Apply Sphericity oriented Contrast Region Growing on the Fuzzy Map and create multiple segmentation results
Display the optimum segmentation result
User accepts
No
Yes
Choose an alternative segmentation result
Output the region

Figure 2.5: Flowchart of the contrast based region growing method.
Source: Dehmeshki et al. (2008).

Other authors also suggest a region growing approach, as it is a popular conventional technique for nodule segmentation. Moltz et al. (2009) dealt with juxtapleural pulmonary nodules in CT scans, by implementing a threshold-based segmentation, followed by 3D region growing. In the end, a morphological opening attempts to eliminate other structures.

After that, Kubota et al. (2011) also used morphological operations and region growing to achieve the segmentation of nodules with diverse densities. In this paper, the algorithm outlined in Figure 2.6 starts by separating the background from the foreground using a threshold, thus separating the wanted structures from the lung parenchyma. Then, it locates the core of the nodule using an Euclidean distance map. Region growing is subsequently applied in this map to isolate the nodule from any other structures. The surface of the nodule is delineated based on the patterns of the region growing and the distance map. Finally, convex hull fills the internal side of the outlined surface voxels previously identified.

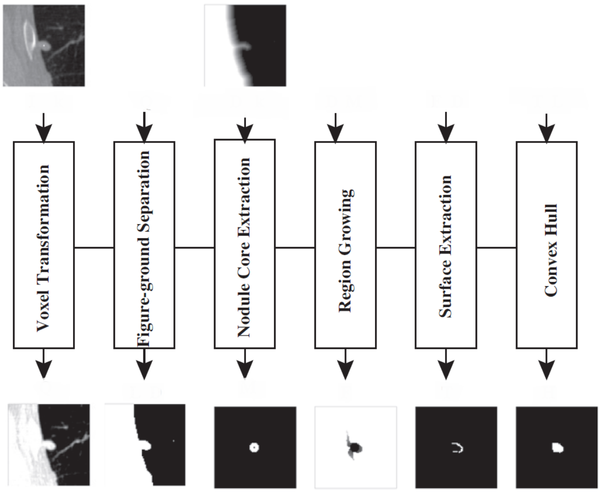


Figure 2.6: Overview of the segmentation algorithm.
Source: Kubota et al. (2011).

Fully automated approaches often take advantage of 3D modeling, as is the case of the deformable model described in Fan et al. (2002). Here, the algorithm works by dynamically initializing and adjusting a 3D template, and analyzing its cross-correlation with the structure of interest (SOI). This work encompasses four stages: first, a threshold is used to extract a structure of interest, defining a nodule candidate region (which may include non-nodular voxels with similar intensity). Then, the 3D template is initialized to get a rough estimation of position, size and shape of that SOI, through the extraction of the core's location. This template is then expanded gradually, computing the cross-correlation at each step and consequently achieving an optimal template of the nodule. The final segmentation results are obtained using logical operations between the optimal template and the SOI, and include a refinement step.

More recently, to tackle GGO segmentation specifically, Jung et al. (2018) employed a similar approach, comprised of three main steps. An intensity-based rough segmentation is achieved using a threshold defined through histogram modeling. This initial segmentation will not include all nodular pixels with lower intensities, and so the result needs to be refined, using an asymmetric multi-phase deformable model. Considering that the previous steps can also lead to the inclusion of pulmonary vessels in the result, the third and final step seeks to remove these structures through a shape-based analysis with a multi-scale approach to remove vessels with different thickness values.

Way et al. (2006) proposed an automatic method in which a 3D active contour follows an automated pulmonary nodule detection, to avoid inconsistent manual detection made by physicians, thus resulting in a segmentation mask. Before applying the active contour, and after detecting the nodule candidates, a 3D k-means clustering algorithm attempts to differentiate nodular and non-nodular regions, followed by a morphological opening which disconnects different structures.

Afterward, another approach that uses active contour to segment several types of nodules was suggested in Li et al. (2016), based on an adaptive local region energy model that uses k-Nearest Neighbors (kNN) clustering. This work faces the segmentation task as an optimization problem and, as such, it adopts an appropriate model to use as cost function. This way, it resorts to Probability Density Function (PDF)-based similarity distance and multi-features dynamic clustering, the last being important for the refinement of the segmentation (particularly in juxtapleural nodules).

Similarly to Way et al. (2006) and Li et al. (2016), the following examples also employ clustering methodologies to achieve a segmentation mask. Zhang et al. (2017) attempted to segment the nodules in CT images using a density-based clustering algorithm on superpixels, more specifically DBSCAN, and therefore identifying the nodule dense regions in 3D space, which revealed to be a fast and effective low computational cost approach.

Yang et al. (2018) resorts to an improved fuzzy c-means clustering algorithm to segment the nodules. The fuzzy c-means typical algorithm aims to set and minimize an objective function, to further identify the center of the cluster and calculate the fuzzy membership matrix. It is possible to make this algorithm more robust by selecting the applicable fuzzy factor, which is done by introducing a weighting factor. This weighting factor is set according to the characteristics of the image pixels and the gray level fluctuations, and is taken into account in the objective function, thus yielding more satisfying results.

Finally, automated segmentation can also be achieved with an unsupervised metaheuristic search, based on evolutionary computation (Figure 2.7), as implemented in Shakibapour et al. (2019). This is a simple unsupervised method that is capable of segmenting different types of nodules. It works with images of the three planes of a lung CT scan (axial, coronal, and sagittal), and starts by enhancing these planes.

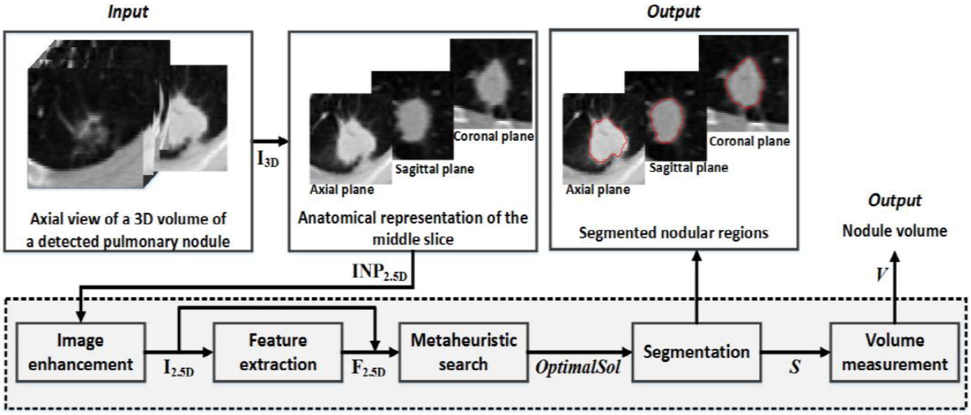


Figure 2.7: Flow chart of the automatic algorithm that uses a metaheuristic search approach. Source: Shakibapour et al. (2019).

The edge and curvature features are extracted, which results in a matrix of feature vectors characterizing every pixel of the three planes. A metaheuristic search approach is used to cluster the feature vectors. Clustering aims to minimize intra-cluster variance and maximize inter-cluster variance. The initial sets of cluster centers are randomly selected and then iteratively evolved and updated by the metaheuristic search method. Once the optimum cluster centers and their respective feature clusters are obtained, segmentation identifies the feature vectors that belong to the same cluster of each plane, outlining the pre-detected nodules. The main advantages of this method are that it does not need annotated images, the metaheuristic search has a fast response time, and that it is based on evolutionary computational searches which approximate solutions by efficiently evolving information.

2.3 Deep Learning Techniques

Deep Learning has been revolutionizing Computer Vision recently. Even though there are aspects that still remain a challenge, major breakthroughs were achieved with these techniques. Deep Learning can have several applications within a CAD scheme. An example of a Deep Learning algorithm for detection of pulmonary nodules and false positive reduction is the one proposed by Setio et al. (2016), in which Multi-view Convolutional Networks were implemented, using a set of nodule candidates as input data. This network has three consecutive convolutional and max pooling layers, whose purpose will be later explained.

The current section aims to briefly describe representative examples of Deep Learning algorithms that intend to segment pulmonary nodules. A much more detailed review on Deep Learning for Image Segmentation is included in Chapter 3.

While CNNs are frequently used in the segmentation of medical images, other types of networks are available. For example, Cheng et al. (2016) intended to develop a Deep Learning based CAD system to discriminate pulmonary nodules present in CT images. To do so, a Stacked Denoising Autoencoder (SDAE) was applied to automatically extract the features, while considering the high variation in shape and size of the lesions. This approach also deals with the inherent noise associated with medical images.

The approach represented in Figure 2.8 was proposed by Wang et al. (2017) and uses a Central-Focused CNN (CF-CNN) to specifically obtain the segmentation of the pulmonary nodules in CT images, which retains much more information due to a novel central pooling layer. The algorithm not only uses 3D images, but also 2D ones, to acquire nodular features. It also considers the neighborhood of the voxel, when attempting to classify it as part of the nodule or part of the healthy structures. The model performs weighted sampling, in the sense that the samples used for training it are chosen based on their segmentation difficulty.

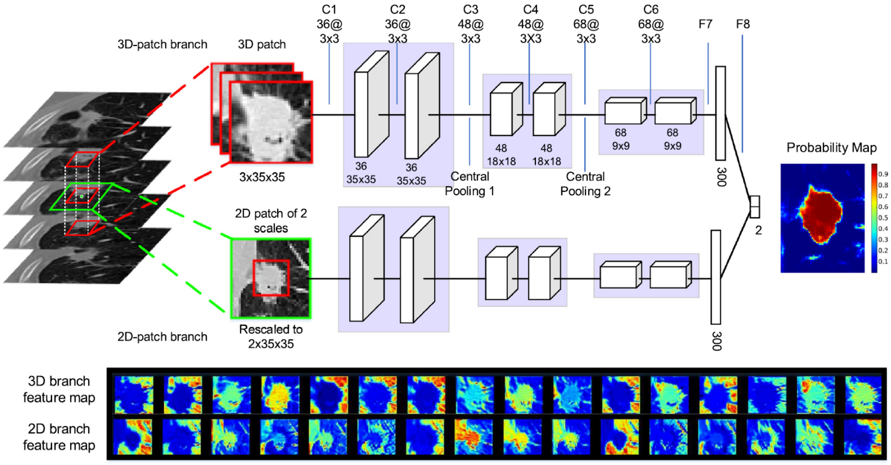


Figure 2.8: Illustration of the CF-CNN architecture.
Source: Wang et al. (2017).

A weakly-supervised CNN-based approach was proposed by Feng et al. (2017). While in a supervised approach the model is trained with fully annotated data (meaning that the training data contains the label to be predicted), in a weakly-supervised strategy both annotated and unannotated data are used for training the model. This strategy achieved accurate voxel-level segmentation with only image-level labels. This means that the proposed CNN model was trained with one label per image, instead of the typical label per voxel - more specifically, a binary slice-level label, which indicates the presence (or not) of a nodule in each slice. The model aims to automate pulmonary nodule segmentation, based on the activation maps of the convolution layers, which determine the discriminative regions of the image. After this step, the model identifies the accurate nodule location with a novel candidate-screening framework. The activation maps correspond to the output activation of a given convolutional layer filter. They are maps that represent where a certain feature is likely to be found in the image (in this case the feature is a nodule). A high activation in a slice image is translated to a preliminary nodule location. This method revealed to have a competitive performance when compared to a standard fully-supervised CNN.

The mentioned weakly-supervised CNN model includes two stages, outlined in Figure 2.9. In the first one, the CNN is trained to attribute a label to each CT slice, which either has a nodule or not. The structure of the CNN combines a fully convolutional stage, that includes a convolutional layer and a global average pooling (GAP) layer, and a final fully-connected (FC) layer. This step not only attributes a binary classification to the images, but also produces a Nodule Activation Map (NAM) that points out the potential nodule locations. The GAP layer enhances the model's performance by promoting a more accurate identification of the discriminative regions and minimizing overfitting through severe dimensionality reduction, as shown in Figure 2.10. The second and final step creates a coarse segmentation, guided by the NAM. A fine segmentation is later achieved by masking each nodule candidate and then feeding each masked image to the same network, originating a residual NAM (R-NAM), which is used to select the final nodules. The model can use a one-GAP CNN structure, or a multi-GAP CNN structure. The first case allows better discriminative power, while the second provides higher resolution NAMs and consequently more accurate location of the nodules.

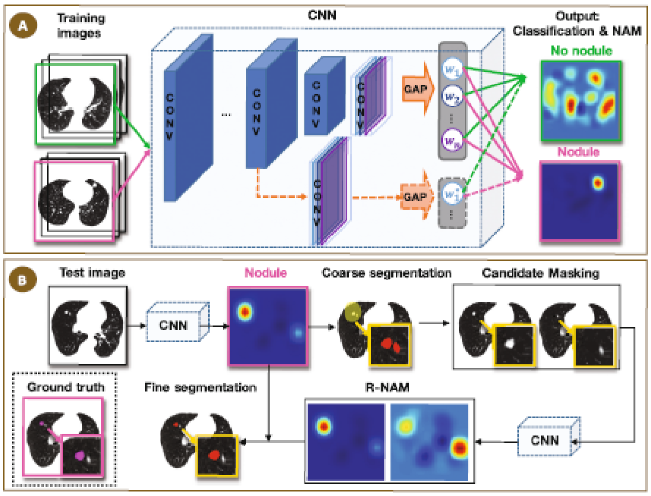


Figure 2.9: a) Training of the CNN model results in classified CT slices and nodule activation maps; b) Coarse segmentation of test slices classified as "nodule slice", followed by fine segmentation guided by Residual NAMs, which are generated from images with masked nodule candidates.

Source: Feng et al. (2017).

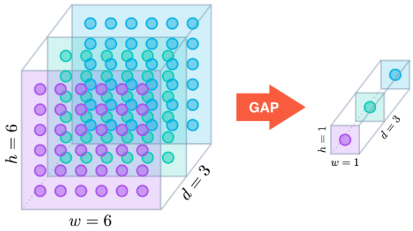


Figure 2.10: Schematic of a general global averaging pooling layer's input and output.
Source: Cook (2017).

Afterward, Wang et al. (2018) dealt with the lack of annotated data by creating another novel weakly-supervised strategy. This strategy (outlined in Figure 2.11) starts with a deep region-based network named Nodule Region-based Convolutional Neural Network (R-CNN), that takes the annotated samples and not only detects the pulmonary nodules but also proceeds to segment them using a 3D region-based network. Then, Deep Active Self-Paced Learning (DASL) aims to use unannotated data by labeling it, and feeding it back to the Nodule R-CNN model training set. DASL, as the name implies, takes advantage of Active Learning and Self-Paced Learning schemes. Active Learning implies that the model interacts with the Nodule R-CNN structure to obtain the labels for the unannotated data, which is useful because of the amount of unlabeled data. Manually labeling would take a lot of effort (time and expertise wise). By annotating it, Active Learning takes advantage of unannotated data that by itself could be considered uninformative for the model. With Self-Paced Learning, the model works according to its current abilities, in the sense that it starts with straight-forward tasks, and then moves on to more complex ones. Self-Paced Learning labels the samples minding prior knowledge and knowledge acquired during the training task. This is a pioneer approach on the 3D pulmonary nodule segmentation problem.

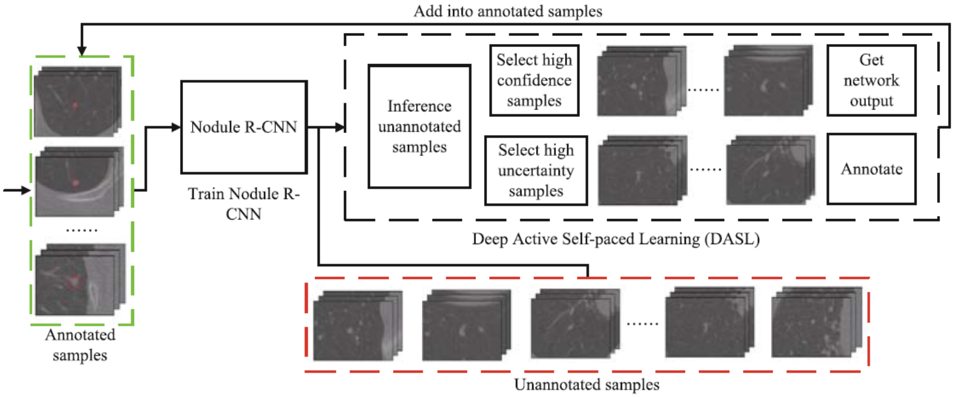

Add into annotated samples
Annotated samples
......
Nodule R-CNN
Train Nodule R-CNN
Inference unannotated samples
Select high confidence samples
Select high uncertainty samples
Get network output
Annotate
Deep Active Self-paced Learning (DASL)
Unannotated samples

Figure 2.11: Weakly-supervised pulmonary nodule segmentation model.
Source: Wang et al. (2018).

Aresta et al. (2018) address the problem at hand with a very common approach, the U-Net, known for its fast and accurate segmentation of medical images, and initially proposed in Ronneberger et al. (2015). For this specific pulmonary nodule segmentation task, the authors implemented minor modifications to the original network in order to increase its performance. On the other hand, Tong et al. (2018) applied more accentuated changes to the original U-Net structure, thus developing an improved U-Net, which introduces a residual network to get even better results. The improved algorithm is outlined in Figure 2.12 and the corresponding model is in Figure 2.13. The V-Net, proposed in Milletari et al. (2016), is also meant to segment medical images, and was inspired in the U-Net, having a similar architecture. However, while the U-Net works with 2D images, the V-Net uses 3D inputs.

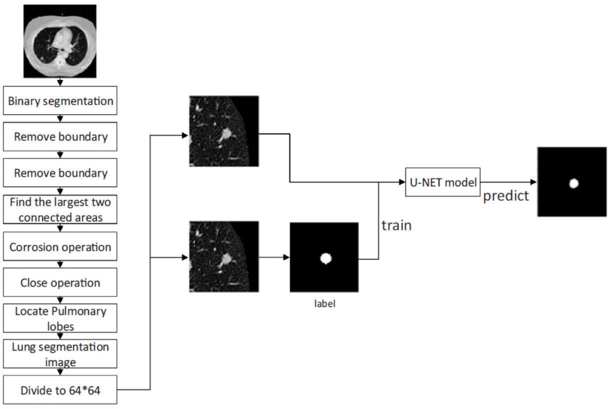


Figure 2.12: Flow chart of the algorithm that implements improved U-Net.
Source: Tong et al. (2018).

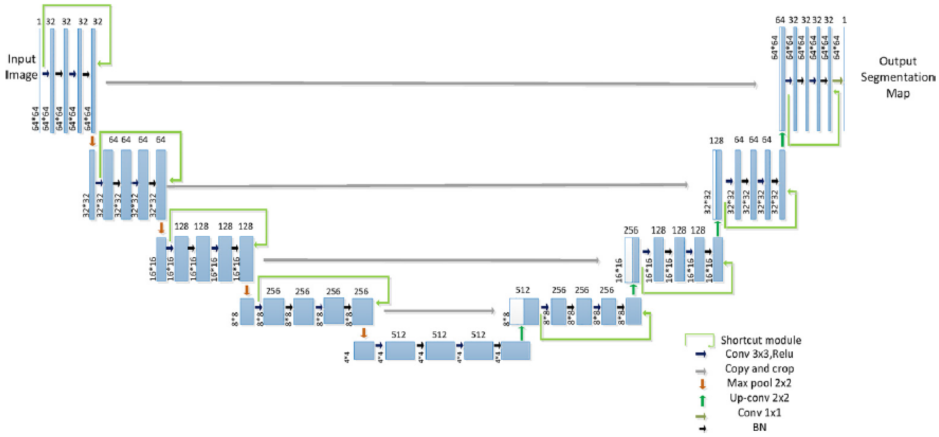


Figure 2.13: Structure of the improved U-Net.
Source: Tong et al. (2018).

Another very popular approach for image segmentation is the Mask Region-based CNN (Mask R-CNN), initially presented by He et al. (2017). Liu et al. (2018b) use this approach for the nodule segmentation task, arguing that the Mask R-CNN is one of the current best instance segmentation models, because it not only performs target detection, but is also able to output the predicted segmentation mask for each detected target. This way, the outcome of the implementation of such algorithm results in the prediction of the nodule's position in the CT image (detection), as well as its boundary (segmentation). To reduce possible interference, the authors segmented the lung region in advance, and then implemented this flexible generic segmentation framework with a Residual Network (ResNet101), He et al. (2015), and Feature Pyramid Network (FPN), Lin et al. (2017), backbone, which yields high-quality segmentation results. The Mask R-CNN was equally implemented by Kopelowitz and Englehard (2019), being a recurrent method to perform this task.

More recently, Huang et al. (2019) proposed a fast and fully-automated approach for pulmonary nodule segmentation, using a customized fully convolutional network, which learns based on CT scans. Considering that the high number of false positives after the detection step hinders the use of CAD systems in clinical practice, this approach starts by taking detected nodule candidates and reducing the number of false positives. To do so, the authors employ a traditional three-layered 2D CNN, as outlined in Figure 2.14. Then, using the geometric centers of the remaining detected nodules, a precise segmentation in achieved through a modified fully convolutional network, which is guided by those centers to get a refined segmentation.

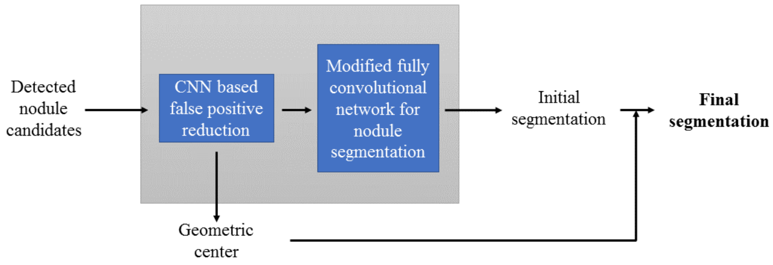

Detected nodule candidates
CNN based false positive reduction
Modified fully convolutional network for nodule segmentation
Initial segmentation
Final segmentation
Geometric center

Figure 2.14: .
Source: Huang et al. (2019).

As mentioned before, Chapter 3 will present in detail the fundamentals on Deep Learning for Image Segmentation, further explaining some of the methods mentioned in this section, and approaching several other networks which are often implemented for this purpose.

2.4 Summary

This chapter starts by highlighting the difficulty of the task at hand, since the segmentation of pulmonary nodules faces several challenges; among them, the inherent noise and low-contrast of medical images, as well as the variability and complexity of the nodules and their surrounding structures. To elaborate on this subject, the diverse nodular classes are explored and described, e.g. the nodules can be characterized based on their location, intensity variations within their borders, and even texture/density. However, in spite of all the challenges, nodule segmentation is valuable for a CAD system. For this reason, the state of the art was studied and the most relevant techniques used for this purpose are outlined in this chapter.

After introducing the types of pulmonary nodules, the chapter then starts by introducing conventional techniques, involving e.g. thresholding, histogram-based methods, clustering, region growing, Watershed segmentation, etc. The state of the art is presented minding former methodologies, as well as more recent ones, and seeks to accompany how these have evolved throughout the years.

Deep Learning based techniques are a recent approach for image segmentation, and have achieved very interesting and promising results in this field. For this reason, Chapter 3 is dedicated to the fundamentals of Deep Learning for Image Segmentation, and further explores the main methodologies used for this purpose. However, the current chapter presents a few relevant examples of Deep Learning techniques which were proposed to deal specifically with pulmonary nodule segmentation, while at the same time minding the inherent disadvantages of such techniques. Convolutional neural networks are likely to be the most frequent approach for medical image segmentation, and so this chapter focus mainly on the studies which employ these relevant architectures. However, other common and/or more innovative approaches are also mentioned.

Chapter 3

Deep Learning for Image Segmentation

Deng (2014) describes Deep Learning as a type of Machine Learning that:

(a) is comprised of several layers of nonlinear processing units for feature extraction and interpretation; the output of one layer is used as input in the next layer

(b) can be considered supervised (e.g. classification) or unsupervised (e.g. pattern analysis)

(c) learns multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts.

This means that the representation of the input data is built initially from low-level features, progressing to high-level ones, as the data moves from layer to layer inside the Deep Learning network. This concept will be further explained later in this chapter.

Supervised learning is the most common learning approach in Deep Learning, according to LeCun et al. (2015), and it refers to methods that make predictions based on a large training set, where each training example is labeled with the corresponding ground truth output. A training example consists of a feature vector describing the object, and a label indicating the object's class. However, it is difficult to get these ground truth labels due to the effort required in the data-labeling process when it comes to the required time and expertise. This led to the need for new learning approaches, such as weak supervision. Zhou (2018) describes three types of weak supervision: incomplete supervision (only a subset of the training data is labeled), inexact supervision (image-level labels instead of pixel-level ones), and inaccurate supervision (the given labels are not correct, so they cannot be considered ground truth). The term weakly-supervised encompasses these three types, and implies the lack of valid and/or sufficient annotated data.

3.1 Artificial Neural Networks

Biological neural networks have inspired researchers to make numerous advances when it comes to developing structures that are able to learn from a given dataset, as stated in Jain et al. (1996). Artificial Neural Networks (ANNs) are systems that perform classification or regression tasks, by learning from a training set, meaning they are able to identify the characteristics to recognize from that set of data. Both classification and regression can be considered predictive exercises: the first aims to predict a class label output for each sample, while the second predicts a continuous quantity output.

A neural network is constituted by an input layer, hidden layer(s), and output layer (e.g. Figure 3.1). A neuron is an unit that computes a number; more specifically, a grayscale value of the corresponding pixel. The degree of activation of a neuron is proportional to the number it holds. In a classification problem, the number of neurons in the last layer is equal to the number of classes being evaluated, and the activation values of those neurons correspond to the probabilities of that object belonging to a certain class.

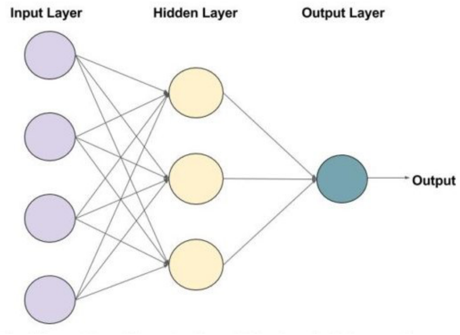


Figure 3.1: Neural network with one hidden layer, containing 3 neurons.
Source: Gupta (2017).

Different images cause different neurons to fire, which means their activation values will also vary. Consequently, the activation values of one layer influence the next. The network has a set of parameters that allow it to detect patterns. A weight (w) is assigned to each connection between the neurons from two distinct layers. Considering the activation values (a) of the previous layer, a weighted sum of these activation values is computed, as shown in Equation 3.1, where n is the number of neurons present in the previous layer.

$$\sum_{i=1}^{n} a_i w_i \tag{3.1}$$

The result of the weighted sum does not fit within a specific range, so in order for the activation to take values from 0 to 1, an activation function is used (e.g. sigmoid function, Figure 3.2). The sigmoid causes very negative inputs to fall close to 0, and very positive ones to fall close to 1, while assuring a steady increase around null inputs.

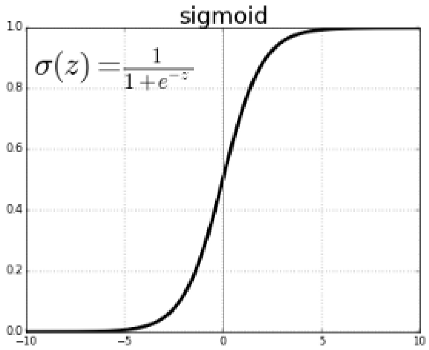


Figure 3.2: Sigmoid function.
Source: Sharma (2017).

This function, represented by σ, is applied to the weighted sum, as shown in Equation 3.2. A bias b is added to the equation - consequently, a neuron is only meaningfully activated when their weighted sum is higher than b. In other words, the bias is a threshold from which a neuron becomes meaningfully activated.

$$\sigma\left(\sum_{i=1}^{n} a_i w_i\right) \tag{3.2}$$

All connections between neurons have their own weights and biases. When it is mentioned that the network is able to learn, it means that it is able to find and adjust the appropriate weights and biases. The resulting equation for each neuron's output is represented in 3.3, where k is the number of neurons present in the current layer.

$$\sigma\left(\begin{bmatrix} w_{0,0} & w_{0,1} & \dots & w_{0,n} \\ w_{1,0} & w_{1,1} & \dots & w_{1,n} \\ \dots & \dots & \dots & \dots \\ w_{k,0} & w_{k,1} & \dots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \dots \\ a_n \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \dots \\ b_k \end{bmatrix}\right) \tag{3.3}$$

Equation 3.4 exemplifies the output (meaning the activation value of the neuron) of a layer preceded by the input layer.

$$a^{(1)} = \sigma\left(wa^{(0)} + b\right) \tag{3.4}$$

3.1.1 Feedforward Neural Networks

Feedforward neural networks are the most primitive type of ANNs, as stated in Schmidhuber (2015), and their connections do not form a cycle (acyclic), meaning that the information flows from the first layer to the last one. According to Haykin (1999), a neural network is considered fully connected when every node in each layer is connected to every node in the next layer; if any connection is missing, then the network is partially connected.

Haykin (1999) identifies two types of feedforward networks:

(a) Single-layer feedforward network — This network, as well as the next, are considered strictly feedforward, as the output of the input layer feeds the next layer, and not the other way around. The term "single-layer" implies there is an unique hidden layer (the input layer is not counted, as no computations are performed there).

(b) Multi-layer feedforward network — This class of feedforward networks has one or more hidden layers. Their nodes are termed hidden nodes, and they process the input, preparing it for the last layer of the network, as will be explained later in this chapter. By adding more hidden layers, their neurons are able to extract higher-order statistics, which is particularly useful when dealing with a large input layer. The final layer of the network outputs the network's response to the activation pattern caused by the input.

3.2 Convolutional Neural Networks

CNNs are considered a specific type of multi-layer neural networks, comprised of learnable weights and biases. They are often applied for image classification, as well as object detection and recognition, where they identify visual patterns directly from the image pixels with minimal preprocessing, as stated in LeCun (2015). However, these networks can also be implemented to solve regression problems.

Their great efficacy has put Deep Learning under the spotlight. The overall success of these networks is partly due to the fact that, when fed with a set of images, CNNs take advantage by shaping their architecture based on their input data, and so the layers of a CNN have neurons rearranged in 3D (*height* × *width* × *depth*), the *depth* being the feature or channel dimension. The images move along the network, passing through its several layers as input and output volumes, which represent the images as matrices whose dimensions change from layer to layer.

Figure 3.3 shows that unlike the regular Neural Networks, CNNs have 3D layers, one or more being convolutional ones. The hidden layer that is present in a regular network is replaced by different types of layers that extract features from the training set.

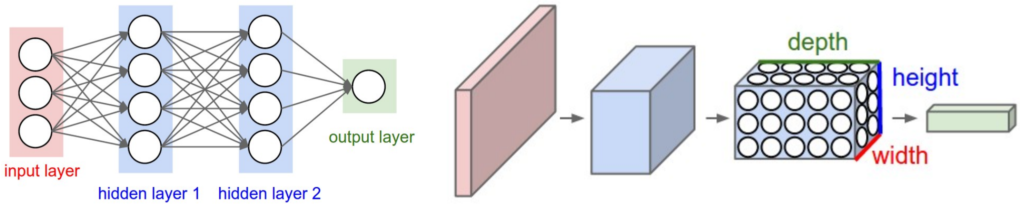


Figure 3.3: A regular 3-layer Neural Network, and a CNN, respectively.
Source: Karpathy (2015).

It is possible to split a CNN into two main components, represented in Figure 3.4:

(a) Convolutional component — CNNs use convolutional layers to look for distinctive features present in the input data. As images go through the 3D layers, the network first looks for low-level features (local features), and then moves on to high-level features, thus performing effective feature learning. In other words, multiple convolutional layers filter all the input volumes from small to greater levels of abstraction.

(b) Fully Connected (FC) component — corresponds to a Fully Connected Feedforward ANN, which, in a classification problem, outputs the predicted classification of a pixel, based on the knowledge captured in the previous layers.

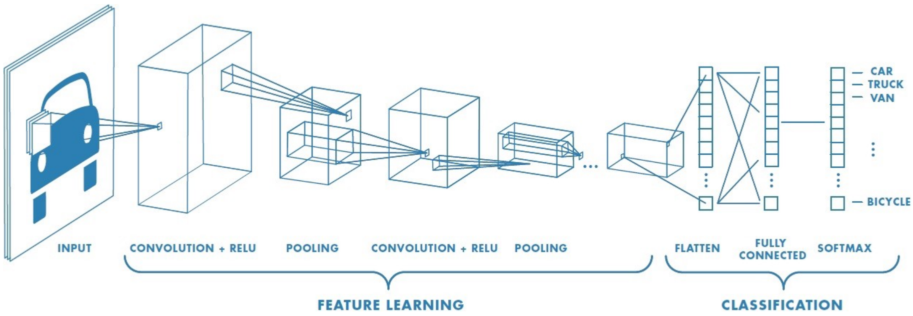


Figure 3.4: Architecture of a CNN, where the hidden layers perform feature learning.
Source: Patel (2017).

In order to understand how CNNs work, it is important to understand what layers are present in their architecture and their role. Below, the details about the most common layers of a CNN are presented.

Input Layer — Images are fed into the CNN through the input layer. In fact, the input layer has the same dimensions as the images, *height* × *width* × *depth*, where the third dimension *depth* matches the number of color channels. For this reason, it is necessary to establish a fixed size for the input images, which might require a resize of the original training set, if they have different dimensions. The input layer holds the raw pixel values, so its dimensions will be *height* × *width* × 1, if the training set contains grayscale images, or *height* × *width* × 3, if it contains RGB images. Figure 3.5 exemplifies an input layer for an RGB image with 4 × 4 × 3 pixels, where each color channel is represented by a 4 × 4 matrix. No computations are performed in this layer.

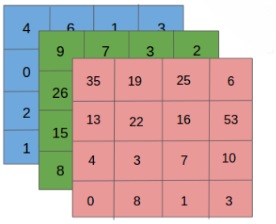


Figure 3.5: Representation of an RGB image.
Source: Gilleman (2018).

Convolutional Layer — A convolution operation preserves the relations between the image pixels, by learning its features. A filter is convolved across the image's width and height, computing the dot product at each position of the filter, as exemplified in Figure 3.6. This way, as the filter moves along the input volume, a 2D feature activation map is generated, matching the responses of the filter at every spatial position of the input image.

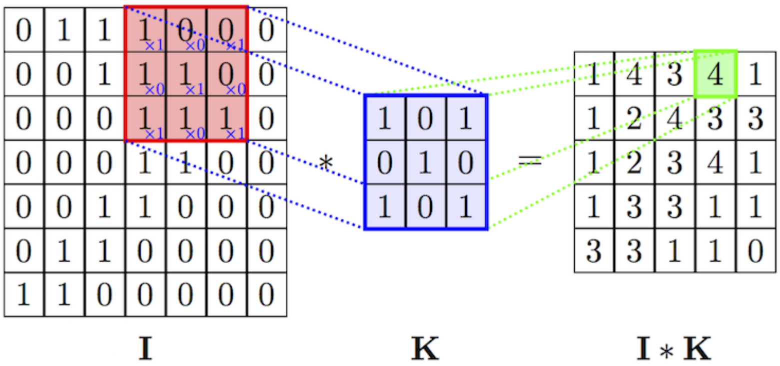


Figure 3.6: Example of a convolution operation between a filter K and a section of an image I. Source: Gilleman (2018).

The response of the filter is high when a certain feature is detected, which means that a CNN can learn from these activation maps, because they indicate the location of a feature that activated the filter (feature detection and extraction). For this reason, the convolution output can be interpreted as a measure of similarity - if the output values are high, the mask is very similar to the equivalent spatial position in the input volume, which allows the model to identify specific features, based on that measure. In the first convolutional layers, local features are identified, such as edges with a defined orientation, or certain colored spots in the image. Higher layers look for more complex features (e.g. patterns).

A typical convolutional filter can be represented by $5 \times 5 \times 3$ pixels, for example. If the input image has $32 \times 32 \times 3$ pixels, then each neuron of the CNN will have 75 weights ($5 \times 5 \times 3$), plus one bias parameter. The number of parameters can be reduced in the network through parameter sharing, thus increasing computational efficiency. As the same filter is used at every spatial position of the input volume, a set of parameters is defined, instead of learning a parameter for each position. However, the output of the convolution does not depend exclusively on these parameters; it also depends on hyperparameters (i.e. their values are set usually by the user before the learning process begins). These hyperparameters described below:

(a) Depth of output volume, k — can be compared to the number of filters to be used, as each one will look for a different feature.

(b) Spatial extent, f — matches the size of the filter, also known as receptive field size.

(c) Stride, s — If the stride is 1, the filter is moved pixel by pixel; if it is 2, then the filter is moved 2 pixels at once (as exemplified in Figure 3.7), etc. A higher stride value results in smaller outputs.

(d) Zero-padding size, p — Padding controls the spatial size of the output volume. Not padding the image results in the loss of information present in the borders after each convolutional layer, which reduces the model's performance. An example of this operation is shown in Figure 3.8.

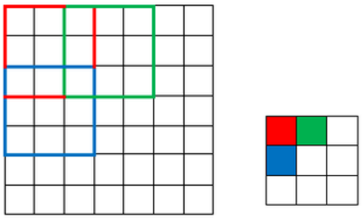
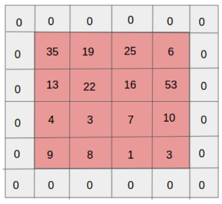


Figure 3.7: On the left, the 7×7 input volume is convolved with a filter whose spatial extent is 3×3 pixels, and whose stride is equal to 2, resulting in the 3×3 output volume represented on the right.
Source: Gilleman (2018).



0	0	0	0	0	0
0	35	19	25	6	0
0	13	22	16	53	0
0	4	3	7	10	0
0	9	8	1	3	0
0	0	0	0	0	0

Figure 3.8: Example of zero-padding.
Source: Gilleman (2018).

Several filters are applied in each convolutional layer, and each one produces a separate 2D activation map. All the maps are stacked along the depth dimension to yield the output volume, whose dimensions are described in Equations 3.5, 3.6, and 3.7 (w_1 and h_1 are the width and height of the previous layer, respectively).

$$\text{output width} = \frac{w_1 - f + 2p}{s} + 1 \tag{3.5}$$

$$\text{output height} = \frac{h_1 - f + 2p}{s} + 1 \tag{3.6}$$

$$\text{output depth} = k \tag{3.7}$$

Activation Function — An activation function is a node that defines the final activation value of a neuron (Figure 3.9). The neuron's output is a linear combination of the input volumes; by adding an activation function, non-linear outputs are permitted, which means that it introduces non-linearity to the convolutional network. By doing so, the model approximates universal functions, as it helps the model to generalize and adapt itself to the data, thus enhancing the model's performance and discriminative capability. The most common activation function is the Rectified Linear Unit (ReLU), but there are other types, such as sigmoid and hyperbolic tangent (tanh). It is important that the activation functions are differentiable, as will be explained in section 3.2.1.

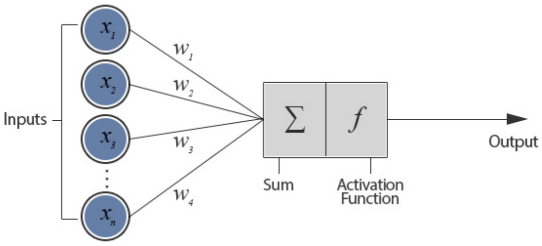


Figure 3.9: Output of a neuron.
Source: Jacobson (2013).

The sigmoid function ranges from 0 to 1, which means that it is possible to find the slope of the curve at any two points. The tanh function ranges from -1 to 1, and performs better than the sigmoid because the negative inputs become very negative and null inputs are mapped near 0, which does not happen with the sigmoid function, as it is possible to observe in Figure 3.10.

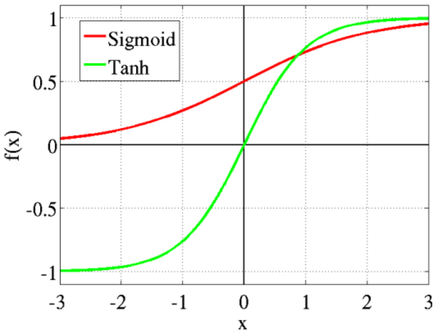


Figure 3.10: Sigmoid and tanh functions.
Source: Sharma (2017).

The ReLU layer is implemented element-wise, using a function $f(z) = max(0,z)$ that ranges from 0 to infinity, as the output of the neuron will be 0 when z is a negative value. However, the instant conversion of all negative values to 0 decreases the efficacy of the model's training, and so a new version of this activation function, named Leaky ReLU, was developed to solve the "dying ReLU" problem. The leak, present in Figure 3.11, broadens the range ($-\infty$ to $+\infty$) with $f(z) = a.z$, where $a \approx 0.01$.

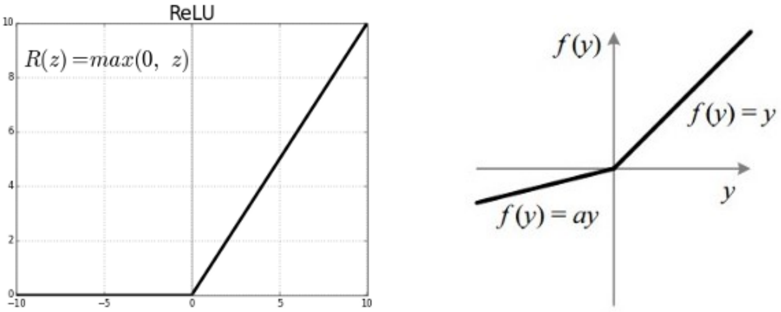


Figure 3.11: ReLU function and Leaky ReLU, respectively.
Source: Sharma (2017).

All activation functions preserve the size of the layer, and do not have any associated hyperparameters.

Pooling Layer — Pooling layers are often implemented between successive convolutional layers in a CNN, in order to decrease the size of their output (Figure 3.12) and consequently decrease the number of parameters and computational power required by the network. This measure also controls overfitting, which happens when a model is specifically conformed to its training set, and does not generalize well when dealing with validation and testing sets.

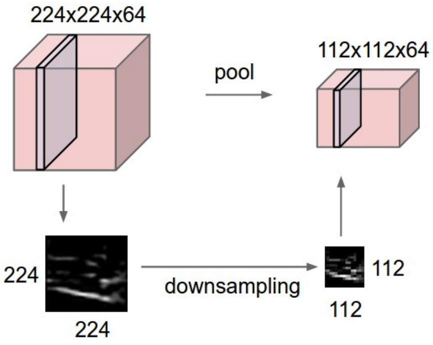
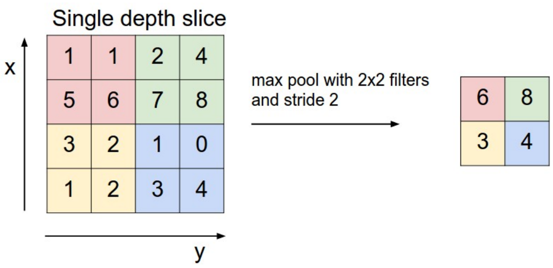


Figure 3.12: Downsampling operation.
Source: Karpathy (2015).

The pooling layer resizes the input's width and height, while keeping the depth unchanged. Such operation can be performed, for example, with max pooling, selecting the maximum value from each cluster of neurons at the previous layer, as outlined in Figure 3.13. However, other pooling operations exist, such as average pooling, L2-norm pooling, and sum pooling.



1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

Figure 3.13: Example of a max pooling operation, with a 2×2 filter and stride equal to 2.
Source: Karpathy (2015).

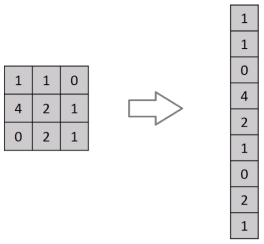
The downsampling operation does not harm the model's performance: once a specific feature is detected in the original input volume (high activation values), its relative location in comparison to other features is more important than its exact location. Given the previous layer's width w_1, height h_1, and depth d_1, the output of the pooling layer is represented in Equations 3.8, 3.9, and 3.10.

$$\text{output width} = \frac{w_1 - f}{s} + 1 \tag{3.8}$$

$$\text{output height} = \frac{h_1 - f}{s} + 1 \tag{3.9}$$

$$\text{output depth} = d_1 \tag{3.10}$$

Dense Layer — A dense layer, also known as a fully connected layer, consists of a linear operation performed on the layer's input, which needs to be flattened. For this reason, the feature map matrix becomes a vector, as exemplified in Figure 3.14. Neurons in a fully connected layer are connected to all neurons in the previous layer (Figure 3.15), as presented in the regular neural networks section. If needed, the fully connected layers can be converted into convolutional layers, and by doing so, the classification model becomes a screening model, containing exclusively convolutional layers.



1	1	0
4	2	1
0	2	1

1
1
0
4
2
1
0
2
1

Figure 3.14: Transformation of a matrix into a vector.
Source: Escontrela (2018).

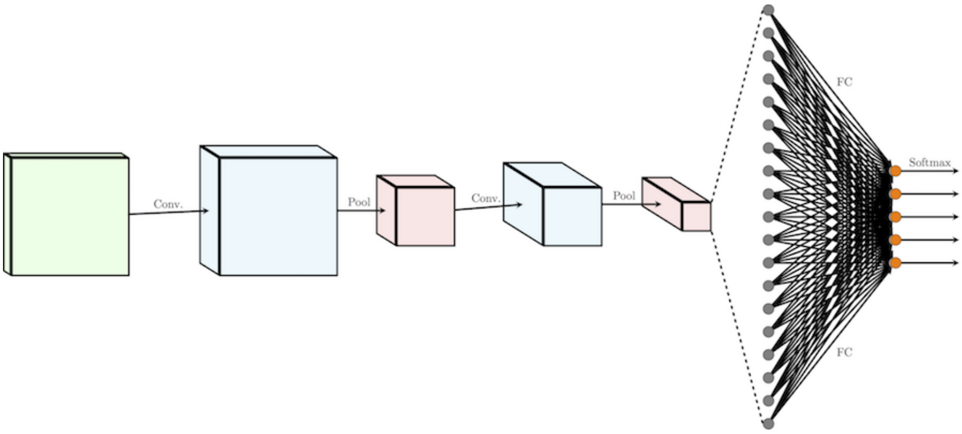


Figure 3.15: Example of a CNN, where a fully connected layer is present.
Source: Escontrela (2018).

Output Layer — The output of the fully connected layer is modified by an activation function, to attribute a pre-defined class to the network's input, in the case of a classification exercise. This way, the result of the activation function determines the output of the CNN. A sigmoid function can be used as an activation function, mapping the output between 0 and 1. However, the softmax is used more often, as it is a generalized logistic function used for multi-class classification. A class is attributed to the initial input based on the extracted features, because the softmax function outputs the probability of that object belonging to each of the pre-defined classes. Figure 3.16 represents a CNN with four classes.

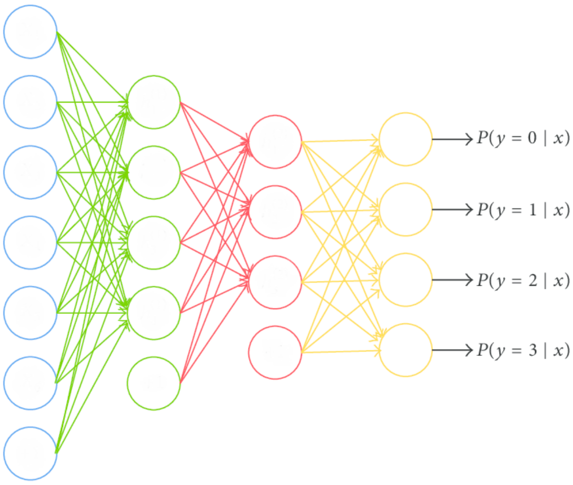


Figure 3.16: CNN with a softmax classifier.
Source: Varma (2017).

LeCun et al. (2015) point out the advantages of CNNs by stating that they are much easier to train, and generalize better in comparison to other networks.

3.2.1 Gradient Descent & Backpropagation

As presented in the previous sections, the activation value of a neuron is influenced by several parameters, namely weights and biases, outlined in Figure 3.17. The weights and biases of the network are initialized with random values, so at first the model exhibits a poor performance. The cost function guides the network towards a better performance, based on the expected classification. More specifically, the gradient descent allows the network to learn, which means adjusting its weights and biases, by finding the minimum of the cost function.

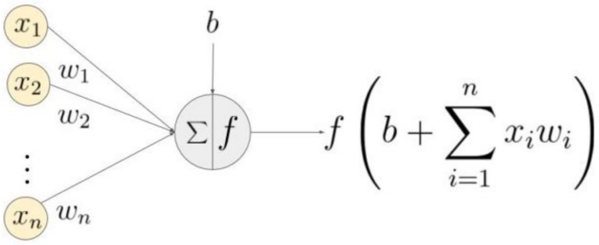


Figure 3.17: Activation value of a neuron.
Source: Goku (2018).

The cost function C considers the average cost of all training data, having the weights and biases as input, and the cost values for each set of input parameters as output. The gradient ∇C finds the steepest increase in the cost function, which means that its negative $-\nabla C$ finds the steepest decrease (the length of this vector is proportional to how steep the function gets). The algorithm computes ∇C (Equation 3.11), takes a small step in the direction of $-\nabla C$, and repeats these steps, until the minimum of the cost function is achieved, as shown in Figure 3.18.

$$\nabla C = \begin{bmatrix} \frac{\partial C}{\partial w^1} \\ \frac{\partial C}{\partial b^1} \\ \dots \\ \frac{\partial C}{\partial w^L} \\ \frac{\partial C}{\partial b^L} \end{bmatrix} \tag{3.11}$$

The gradient descent represents how sensitive the cost function is to each weight and bias, advising how their values should be changed in order to reach the minimum of that function - the absolute value of each element tells how much that value should be increased (if positive) or decreased (if negative).

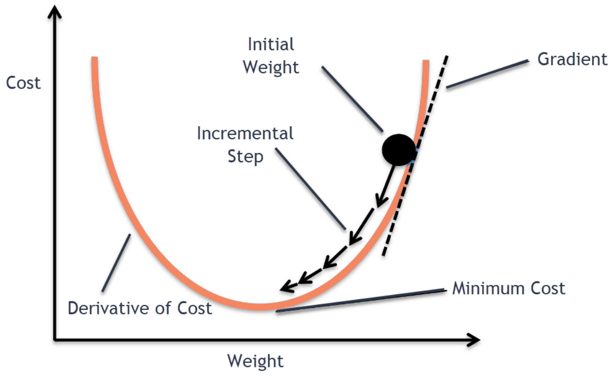


Figure 3.18: Gradient descent method.
Source: Lanham (2018).

In other words, the gradient expresses how much the outputs (activation values of the last layer) are influenced by a change in the inputs. The learning rate determines how fast the gradient descent will move towards the optimal weights, and consequently the minimum of the cost function; for this reason, the learning rate should not be very high (otherwise it will bounce back and forth, as happens in Figure 3.19), nor very low (as it will require too much time).

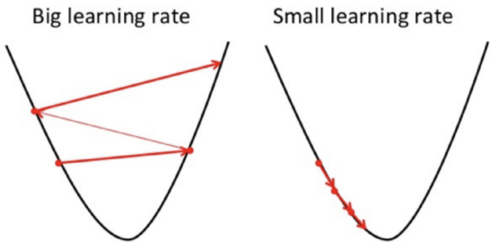


Figure 3.19: Learning rate examples.
Source: Donges (2018).

There are several approaches for the calculation of the gradient descent (such as batch gradient descent, stochastic gradient descent, and mini-batch gradient descent), but these approaches are not guaranteed to find the global minimum of the function, only a local minimum.

During a classification problem, it is intended for an input object to be classified correctly, meaning that the output of that class should have a higher activation value than the others'. In order to get this high activation value, the cost function is calculated during the forward pass (yellow arrows in Figure 3.20), comparing the achieved output and the expected one, and then the weights and biases are adjusted during the backward pass to approximate both values as much as possible (purple arrows in Figure 3.20). Equation 3.12 shows how the cost function compares the activation values $a_i^{(L)}$ of the last layer L to the expected activation value y_i, for the n training examples, as presented in Maini and Sabri (2017).

$$C = \frac{1}{2n} \sum_{i=1}^{n} \left(a_i^{(L)} - y_i \right) \tag{3.12}$$

The backpropagation algorithm calculates the gradient based on two partial derivatives, $\frac{\partial C}{\partial w}$ and $\frac{\partial C}{\partial b}$, in order to find the values of the weights and biases that decrease the cost.

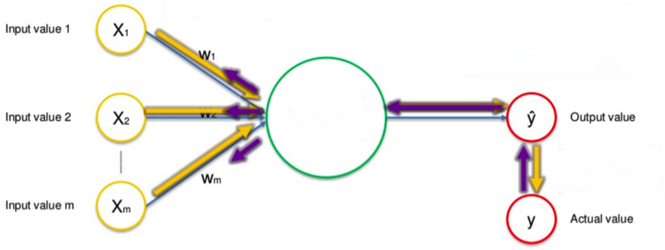


Figure 3.20: Forward and backpropagation represented in yellow and purple, respectively. Source: Eremenko (2018).

3.2.2 Regularization

The model performs poorly on unseen data, whether it is a validation or a test set, if it is too adjusted to the training data (overfitting). The training error decreases as the model's complexity increases; however, if the model is overfitting, the test error starts increasing after a certain point. Figure 3.21 shows how the error changes based on the model's complexity.

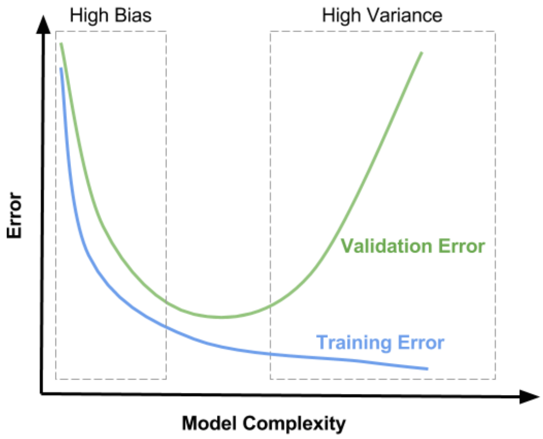


Figure 3.21: Error as a function of the model's complexity. Source: Ziganto (2018).

The complexity of the network makes it prone to overfitting. In order for the model to generalize better, regularization techniques can be implemented, and consequently the model will perform better when dealing with unseen data. Regularization reduces overfitting by penalizing the weight matrices of the network's neurons. This will lead to a simpler model, with a higher bias and a lower variance. If the test error is still high after applying a regularization technique, the model is underfitting, meaning that it can neither perform well with training data not generalize with new data (Figure 3.22), which means the model is too simple and already has a high bias. The solution is to add complexity to the model and then use a regularization technique, as presented below.

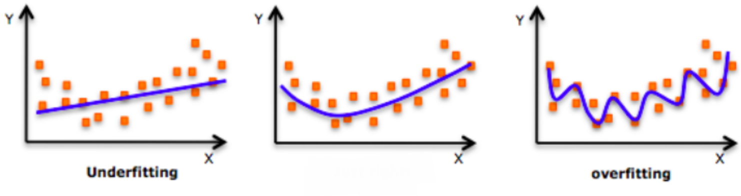


Figure 3.22: From left to right: underfitting, well adjusted, and overfitting model.
Source: Jain (2018).

3.2.2.1 Early Stopping

A validation set is a fraction of the training set (as seen in cross-validation), that can be used to evaluate the performance of the trained model. When the performance on the validation set starts decreasing, the training should be stopped; otherwise the model will overfit to the training data. The optimal model complexity happens just before the training error begins to increase, as presented in Figure 3.23.

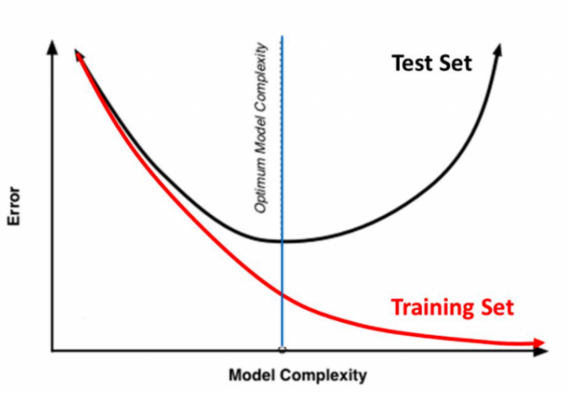


Figure 3.23: Optimum model complexity.
Source: Jain (2018).

3.2.2.2 L2 & L1 regularization

Both L2 and L1 are common types of regularization; they update the cost function by adding a regularization term to it, that causes the values of the weight matrices to decrease, as smaller weights lead to simpler models. The regularization term includes a regularization hyperparameter λ, controlling how much the weights are going to be penalized, and whose value should be optimized. A large λ will approximate the weight values to zero, and a null λ does not allow any regularization, which means that optimized λ should be in between those values.

The L2 regularization is also known as weight decay, because it forces the weights to be very close to zero, but not exactly null. According to this technique, the cost function becomes as represented in Equation 3.13 taken from Maini and Sabri (2017), where the second portion of the sum is the regularization term.

$$C = \frac{1}{2n}\sum_{i=1}^{n}((a_i w_i + b) - y_i)^2 + \lambda\sum_{i=0}^{n} w_i^2 \tag{3.13}$$

On the other hand, the L1 regularization penalizes the absolute values of the weights, and can reduce them to zero, thus performing feature selection by eliminating neurons from the network, which becomes simpler. Equation 3.14 represents the cost function when the L1 regularization is implemented.

$$C = \frac{1}{2n}\sum_{i=1}^{n}((a_i w_i + b) - y_i)^2 + \lambda\sum_{i=0}^{n} \|w_i\| \tag{3.14}$$

3.2.2.3 Dropout

The dropout is another regularization technique where randomly selected neurons of all layers are shut down on each iteration, so they are not used during the forward and backpropagation. Such action causes the network to not focus on specific neurons, as they may be ignored, thus training a "different" model on each iteration. By reducing the number of neurons, the dropout leads to a smaller and simpler model, as shown in Figure 3.24, being the most common regularization technique and leading to great results.

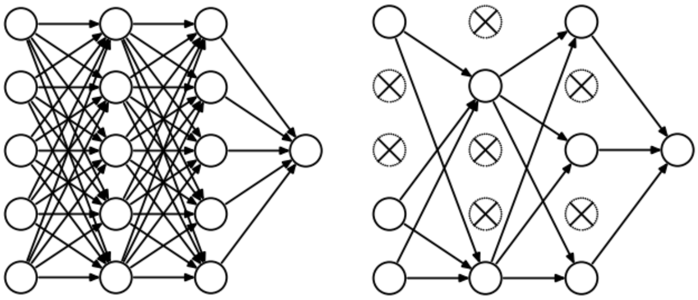


Figure 3.24: The original network, and the same network after dropout.
Source: Srivastava et al. (2014).

As each iteration selects different neurons, the model's output will also be different with each iteration. The hyperparameter of the dropout function controls how many neurons should be dropped (they might belong to the input or hidden layers). The dropout is usually preferred when dealing with a large network.

3.2.2.4 Data Augmentation

A simple and effective way to enhance the performance of a Deep Learning model is to add more examples to the training set. However, collecting and labeling data is time consuming, and may require a certain proficiency, as is the case of biomedical image labeling. Data augmentation tackles the need for more training data, without adding new images to it. It encompasses all techniques that artificially generate more training examples to feed the model, based on the existing ones.

Mikolajczyk and Grochowski (2018) state that affine image transformations and color modifications are the most common data augmentation techniques. Berger (1994) defines an affine transformation as a function between spaces which preserves points, straight lines and planes, and Weisstein (2018) adds that an affine transformation preserves not only collinearity (i.e. all points lying on a line initially still lie on a line after transformation), but also distance ratios (e.g. the midpoint of a line segment remains the midpoint after transformation). Affine image transformations include rotation, reflection, scaling (both zoom in and zoom out), and shearing (Figure 3.25). On the other hand, color modifications include histogram equalization, contrast and brightness enhancement, white balancing, sharpening, and blurring (Figure 3.26).

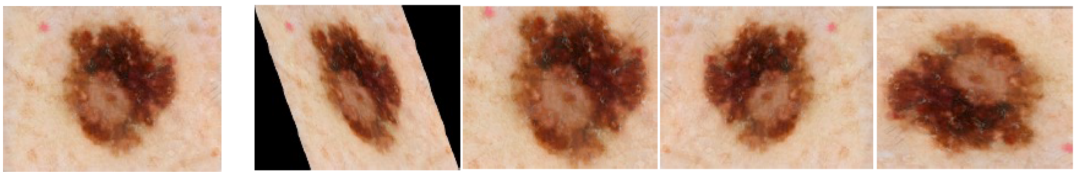


Figure 3.25: Original image and the results of different affine transformations (shear, zoom in, reflection, and rotation, respectively).
Source: Mikolajczyk and Grochowski (2018).

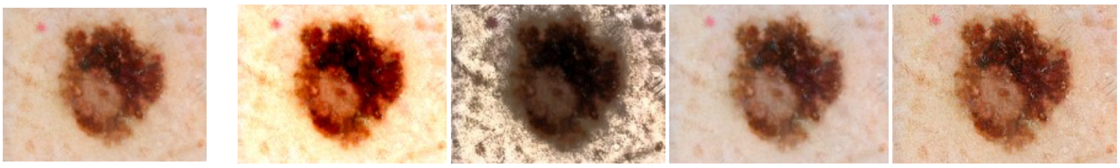


Figure 3.26: Original image and the results of different color transformations (contrast enhancement, histogram equalization, white balancing, and sharpening, respectively).
Source: Mikolajczyk and Grochowski (2018).

Rotations, scaling, and translations are often used in data augmentation for medical images, as is the case of Castro et al. (2018), that used data augmentation to increase the number of mammogram images in the training set. The U-Net, which aims to segment medical images, also resorts to data augmentation, as will be further explained in Chapter 6.

Perez and Wang (2017) prove the effectiveness of data augmentation, even when using simple techniques (such as cropping, rotating, and flipping the input images), thus increasing the accuracy of classification tasks.

3.2.3 Batch Normalization

Batch normalization was developed by Ioffe and Szegedy (2015), and its purpose is to adjust and scale the activation values of the hidden layers, and therefore reduce the training time of the network, as well as increase its stability. The batch normalization is associated with mini-batch statistics, including a mean μ_B and a variance σ_B^2, causing the output of the hidden layers (meaning their activation values) to have a null mean, and variance equal to 1. Two trainable weights, γ and β, are added to each layer to maintain that mean and variance, represented in Figure 3.27. However, after modifying the activation values, the weights in the next layer may not remain optimal, and so the Stochastic Gradient Descent (SGD) undoes the normalization, if it causes the minimization of the loss function.

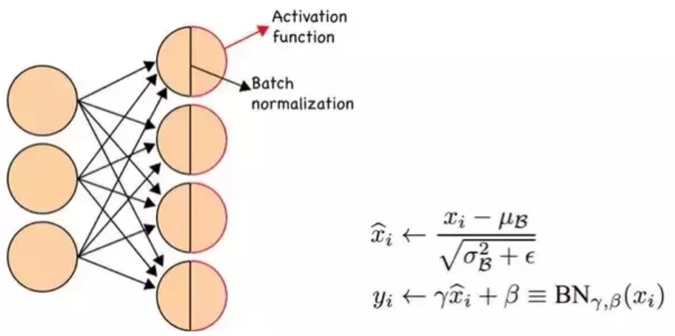

Activation function
Batch normalization
$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$
$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i)$

Figure 3.27: Batch normalization: $\hat{x}_i$ represents the normalized input, while y_i represents the scaled and shifted batch normalization output.
Source: Deng (2017).

The batch normalization solves the co-variance shift problem, which slows down training and demands careful weight initialization. Because of that normalization, higher learning rates can be used, as no activation value will be really high or really low. By having a slight regularization effect, this operation also reduces the chances of overfitting. However, the regularization should not depend exclusively on batch normalization - it should be used with dropout as well. Furthermore, it is possible to conclude that batch normalization causes a layer to learn more independently from the other layers.

3.3 Deep Learning for Image Segmentation

CNNs are often used for image analysis, and image segmentation tasks in particular, where it is intended to partition an image into multiple segments. Medical image segmentation commonly relies on CNNs, as stated in Jiang et al. (2018).

Image segmentation attains the mask of the object(s) of interest, by selecting the portions in the image that belong to it. This task of grouping pixels into classes can be achieved through isolated pixel classification (pixel-wise), or region-based classification techniques.

Two distinct types of segmentation can be interpreted, which are represented in Figure 3.28. In instance segmentation, each distinct object will be delineated individually, as explained in Romera-Paredes and Torr (2016). This means that if there are, for example, two nodules in a single image, they will have their distinct labels and masks - nodule 1 and nodule 2. On the other hand, semantic segmentation splits the image into significant portions, in order to understand its content. In other words, it divides the image into semantically meaningful regions, so if there are two nodules in it, they would be attributed to the same category, and the segmentation mask would encompass both of them.

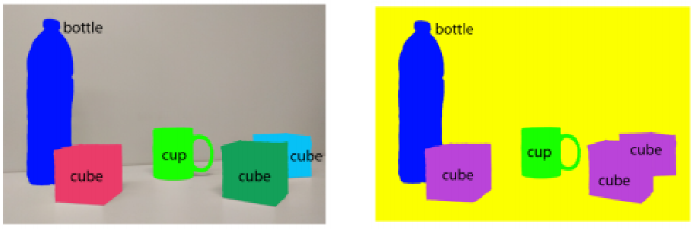


Figure 3.28: Instance and semantic segmentation, respectively.
Source: Garcia-Garcia et al. (2017).

At first, Deep Learning based segmentation relied on a sliding window method, that split the input image into equally sized sections. These sections were fed into a CNN to be classified, being considered a pixel-wise classification if each section was formed by a single pixel. However, such method was very computationally inefficient, and there was a need to reduce the time spent on training the model. Currently, CNNs achieve good segmentation performance through the use of Fully Convolutional Networks (FCNs), and Encoder-Decoder Networks.

The fully connected layers of a CNN cause the loss of information acquired during the convolutional layers. Since spatial information is crucial for segmentation tasks, Long et al. (2014) proposed a network where the fully connected layers are converted into convolutional ones. This originated the FCNs, commonly applied for semantic segmentation, whose typical architecture is displayed in Figure 3.29. They can process the entire input image at once, and the output will have the same dimensions as the input images.

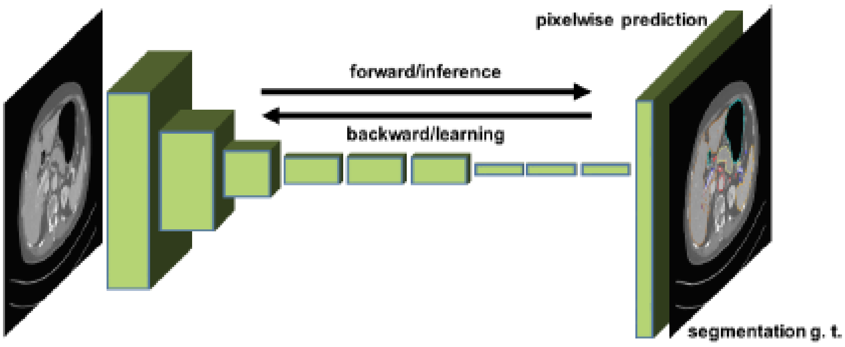


Figure 3.29: Fully convolutional network applied in medical imaging. The input image is a CT slice.
Source: Roth et al. (2018).

This way, FCNs do not keep any fully connected layer, encompassing only locally connected layers (such as convolutional, pooling, and upsampling ones), which results in a smaller amount of required parameters, and less computation time. The pooling layers downsample the information, as explained before, so the final upsampling layers are used to output a classification map with the same dimensions as the input, using e.g. a softmax probability function. The mentioned final upsampling is achieved with transposed convolution, also known as upconvolution.

Encoder-Decoder networks can also be employed in semantic segmentation tasks and proved to be more memory efficient than the FCN proposed in Long et al. (2014). Such networks comprise two parts: downsampling (where the semantic contextual information is achieved and features are extracted), and upsampling (where spatial information is restored and the features are located). The encoder aims to get feature representations from the input images, while the decoder reaches an output prediction based on those representations. A frequent fully convolutional encoder-decoder architecture is the SegNet, proposed in Badrinarayanan et al. (2015) and shown in Figure 3.30, whose knowledge is built upon the analysis of the low-dimensional representation of the input image. The decoder stage is usually symmetrical to the encoder one, creating a symmetrical architecture that does not happen in the FCN previously presented in Figure 3.29. The downsampling and upsampling processes can also be addressed as convolution and upconvolution, and they are exemplified in Figure 3.31. Besides the SegNet, the U-Net is also a popular fully convolutional encoder-decoder network; both will be explored in detail in Chapter 6.

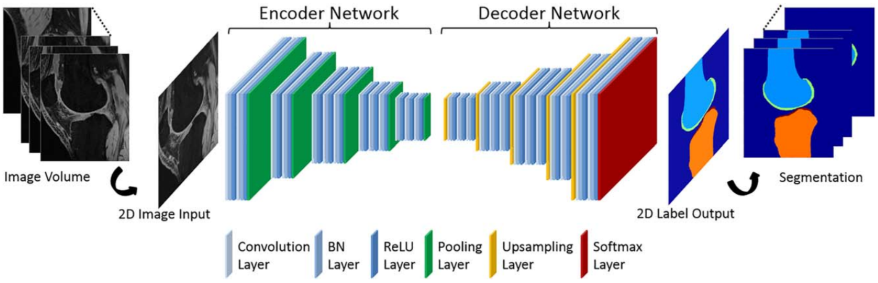


Figure 3.30: Illustration of the SegNet's encoder-decoder architecture for semantic segmentation. The left half of the scheme is the convolution stage, while the right half is the upconvolution stage.
Source: Liu et al. (2018a).

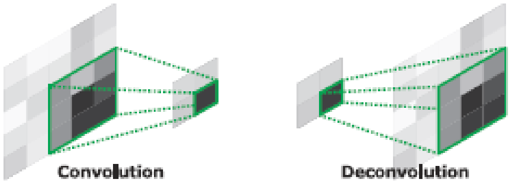


Figure 3.31: The representation on the left exemplifies the image downsampling process during the convolution. The one on the right exemplifies the upsampling (upconvolution) of the convolution result.
Source: Noh et al. (2015).

More recently, He et al. (2017) developed the Mask Region-based CNN (Mask R-CNN) for instance segmentation, incorporating an FCN in its algorithm (Figure 3.32). In the first stage, a Region Proposal Network (RPN) proposes the candidate object bounding boxes (object detection, Figure 3.33), and then, in the second stage, while the class and the bounding box are being predicted for the object of interest, a binary mask is also computed. This way, classification and regression are computed in parallel. Such network requires fewer parameters and gets high accuracy values.

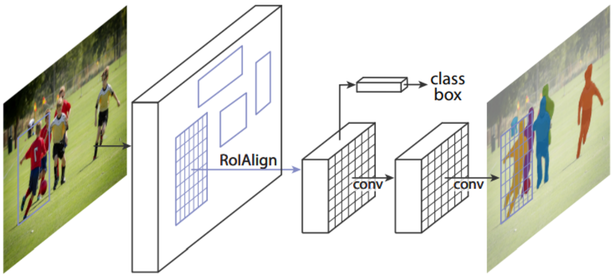


Figure 3.32: Mask R-CNN.
Source: He et al. (2017).

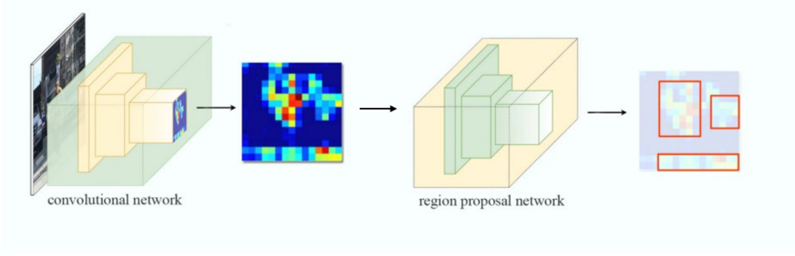


Figure 3.33: Region Proposal Network.
Source: Hui (2018).

As stated in He et al. (2017), labels and bounding boxes are collapsed into fully connected layers. However, a mask that encodes the object's spatial layout can result from a pixel-wise classification using convolutions. For this reason, an FCN is incorporated into the algorithm to predict the mask for each object of interest.

3.4 Summary

This chapter presents the fundamentals of Deep Learning, starting with a brief definition. Artificial Neural Networks are introduced as a primitive type of Deep Learning, and then the chapter moves on to more complex structures like Convolutional Neural Networks, giving a brief explanation about their most common layers (e.g. convolutional, ReLU, pooling, and fully connected). Each layer may or may not have parameters (e.g. convolutional and fully connected do, ReLU and pooling do not), and may or may not have additional hyperparameters (e.g. convolutional, fully connected, and pooling do, ReLU does not). Backpropagation is described as an algorithm used for training the model, based on gradient descent. Furthermore, regularization techniques are explored, including dropout and data augmentation. Batch normalization is also referred.

Finally, the chapter explains how Deep Learning can be used for image segmentation (whether instance or semantic), by implementing diverse approaches. Fully Connected Networks are often associated with this task, being used in Encoder-Decoder Networks, which are more memory efficient and produce satisfying segmentation results. Among these Encoder-Decoder Networks, it is possible to find the SegNet and the U-Net. More recently, other approaches have taken place, as is the case of the Mask R-CNN.

Chapter 4

LIDC Database

The Lung Image Database Consortium (LIDC) is publicly available and consists of lung cancer screening thoracic CT scans, in which the lesions are annotated. This database is often used for training and validating CAD systems that aim to provide lung nodule detection, and ultimately cancer diagnosis. Armato et al. (2011) characterize this database as a heterogeneous set of 1018 CT images from 1010 different patients, acquired by seven academic centers and eight medical imaging companies. The composition of a region in a CT scan can be identified based on differences in radiodensity, which is measured in Hounsfield Units (HUs), as indicated in Motley et al. (2001). In other words, the HU is an attenuation unit to interpret CT scans by characterizing the relative density of a substance. This way, according to Bolliger et al. (2009), each pixel is assigned a value between -1000 (air) and 30000 HUs (foreign body, e.g. gold). For the lung tissue, the HU value is approximately -700 HU, as stated in Kazerooni and Gross (2004).

4.1 LIDC Annotation Process

A XML file contains the label for each lesion detected in the CT scans. The proposed annotations were achieved in two phases. A total of twelve radiologists were involved in this process, and for each CT scan, four of them were randomly selected to annotate the lesion. In the first phase of the annotation process, the four radiologists assigned one of the following labels to independently characterize the lesion, referring to the greatest in-plane dimension:

(a) Nodule greater than or equal to 3mm
(b) Nodule with less than 3mm
(c) Non-nodule greater than or equal to 3mm.

The third label is attributed to a structure identified as a pulmonary lesion whose characteristics are not consistent with the ones of a nodule (e.g. an apical scar). An example of each label can be analyzed in Figure 4.1. This first phase is known as “Blinded Read Phase”, as the lesions are independently analyzed by each specialist, without any influence from the other three radiologists.

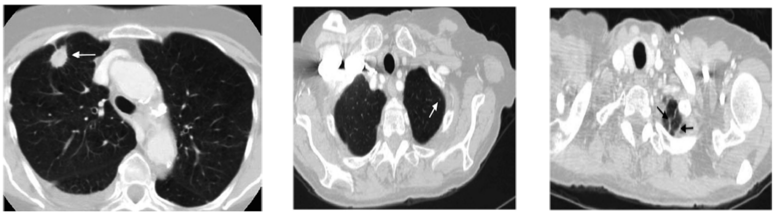


Figure 4.1: Examples of lesions that represent a nodule greater than or equal to 3mm, a nodule with less than 3mm, and a non-nodular structure greater than or equal to 3mm, respectively.
Source: Armato et al. (2011).

The second and final phase was the "Unblinded Read Phase". Here, the annotations obtained in the first phase were sent to all four radiologists involved, so they could analyze the lesion again, minding their colleagues' opinions. The radiologists reviewed all the classifications, with which they agreed with or not, and after that they could choose to keep their original annotations or change them. For this reason, the LIDC does not hold a consensual annotation list. Besides that, it is possible that a nodule is not annotated by one or more radiologists, if they believe the nodule does not fit in any category. This means that a lesion can have one, two, three, or four annotations. Furthermore, in this phase, the specialists also characterized the nodules $\geq$ 3 mm, as will be explained in the next section. The LIDC database includes all the DICOM (and respective XML) files for each of the 1018 CT scans.

4.2 Annotation Results

Considering that different categories could be attributed to the same lesion, as the database accepts different opinions among the four radiologists, it is important to realize how divergent (or not) the labelings of the lesions were. One example of conflicting and non-conflicting classifications can be found in Figure 4.2. The four radiologists agreed that in 1.9% of the CT scans containing a lesion were non-nodular. As 1.9% represents a small portion of scans, it was considered that the dataset is mostly comprised of nodular lesions. 7371 were considered to be nodular by at least one of the four radiologists. A summarized description of the radiologists' opinions regarding nodular lesions can be found in Table 4.1.

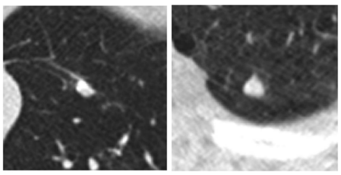


Figure 4.2: Two lesions annotated as nodules $\geq$ 3 mm. The lesion on the left was marked by only one radiologist, while the other specialists labeled it as non-nodular $\geq$ 3 mm. On the right, all radiologists agreed the lesion was nodular $\geq$ 3 mm.
Source: Armato et al. (2011).

The database emphasizes the importance of labeling nodules ≥ 3 mm, because they have a higher probability of being malignant and are often used by CAD developers. 2669 of the 7371 nodular lesions were classified as a nodule ≥ 3 mm, by at least one radiologist.

Not every lesion was attributed a category - for example, a certain lesion could be assigned a label by three radiologists, while the fourth could consider it doesn't belong to any category. 10.1% (744 of the 7371 nodular lesions) were labeled as nodular by only one radiologist. On the other hand, 46.1% (3396) were labeled as nodular by all four specialists (either nodule ≥ 3 mm or < 3 mm). Only 26.3% (1940) of the nodular lesions were assigned the same category by all the radiologists.

Table 4.1: Summary of the radiologists' annotations in the 1018 CT scans.

Description	**Number of Lesions**
At least one radiologist assigned either a nodule ≥ 3 mm mark or a nodule < 3 mm mark.	7371
At least one radiologist assigned a nodule ≥ 3 mm mark.	2669
All four radiologists assigned a nodule ≥ 3 mm mark.	928
All four radiologists assigned a nodule ≥ 3 mm mark or all four radiologists assigned a nodule < 3 mm mark.	1940
All four radiologists assigned either a nodule ≥ 3 mm mark or a nodule < 3 mm mark.	2562

Source: Armato et al. (2011).

Considering the 2669 lesions annotated as a nodule ≥ 3 mm by at least one radiologist, 29.1% (777) were classified as such by a single specialist. 34.8% (928) were annotated as nodules ≥ 3 mm by all four radiologists.

4.2.1 Characteristics of the Annotated Nodules

As mentioned before, the LIDC focuses on nodules ≥ 3 mm due to their high risk of malignancy. For this reason, during the unblinded reading phase, each radiologist is asked to characterize (in a subjective way) the lesions labeled as nodules ≥ 3 mm. In this analysis, a set of descriptors is defined, which includes calcification, internal structure, lobulation, malignancy, margin, sphericity, spiculation, subtlety, and texture.

The nodule properties are described in McNitt-Gray et al. (2007), as follows:

(a) Calcification — pattern of calcification, if present.
(b) Internal Structure — internal composition of the nodule (soft tissue, fluid, fat, air).
(c) Lobulation — the degree of lobule-resembling appearance, ranging from none to marked.
(d) Malignancy — subjective assessment of the likelihood of malignancy, assuming the scan originated from a 60-year-old male smoker.
(e) Margin — description of how well-defined the margin is.
(f) Sphericity — the three-dimensional shape of the nodule in terms of its roundness.
(g) Spiculation — the extent of spiculation present.
(h) Subtlety — difficulty of detection.
(i) Texture — internal texture (solid, ground glass, or mixed).

In order to facilitate the characterization of the nodules by the radiologists, a standard scoring code was defined (explicit in Table 4.2), where a number corresponds to a specific property of the nodule. This way, a number was attributed to each characteristic, by the four radiologists involved (e.g. when it comes to malignancy, a nodule is likely to be benign if it is attributed 1 or 2, and malignant when attributed 4 or 5).

The results of this characterization were obtained for 2669 lesions. Table 4.3 shows, for each specific property, the percentage (of the 2669 lesions) that is characterized as 1, 2, 3, 4, 5, or 6. This information is valuable to understand what types of nodules are present in this database and how many belong to that class. For example, it is possible to conclude that most nodules:

(a) are not calcified (85%)
(b) are made of soft tissue (98%)
(c) have no lobulation (54%)
(d) are inconclusive about their malignancy (47%)
(e) have a sharp margin (39%)
(f) assume an intermediate shape between ovoid and round (46%)
(g) have no spiculation (62%)
(h) are moderately obvious (35%)
(i) are solid (70%)

Table 4.2: Scoring for each nodule characteristic. Radiologists are allowed to use intermediate values for Lobulation, Margin, Sphericity, Spiculation, and Texture. N/A stands for not applicable.

Scoring	1	2	3	4	5	6
Calcification	Popcorn	Laminated	Solid	Non-central	Central	Absent
Internal Structure	Soft Tissue	Fluid	Fat	Air	N/A	N/A
Lobulation	None				Marked	N/A
Malignancy	Highly Unlikely	Moderately Unlikely	Indeterminate	Moderately Suspicious	Highly Suspicious	N/A
Margin	Poorly				Sharp	N/A
Sphericity	Linear		Ovoid		Round	N/A
Spiculation	None				Marked	N/A
Subtlety	Extremely Subtle	Moderately Subtle	Fairly Subtle	Moderately Obvious	Obvious	N/A
Texture	Non-solid/ GGO		Part-Solid/ Mixed		Solid	N/A

Source: Yip et al. (2017).

Table 4.3: Percentage of nodules (in a sample set of 2669 nodular lesions) that are characterized as 1, 2, 3, 4, 5, or 6, for each property assessed.

LIDC (%)	1	2	3	4	5	6
Calcification	0,00	0,00	7,16	3,07	3,94	85,83
Internal Structure	98,99	0,60	0,30	0,07	0,04	
Lobulation	54,46	29,69	11,02	3,49	1,35	
Malignancy	10,79	22,94	47,04	14,99	4,24	
Margin	3,97	8,51	15,07	32,83	39,62	
Sphericity	0,07	4,87	25,90	46,78	22,38	
Spiculation	62,56	24,48	7,35	3,90	1,72	
Subtlety	4,35	9,71	21,55	35,08	29,31	
Texture	6,56	3,67	5,88	12,97	70,91	

Source: Armato et al. (2011).

More information about these results can be found in Appendix A.

4.3 LIDC for Pulmonary Nodule Segmentation

Chapter 1 points out that manual nodule examinations are error-prone partly due to their subjectivity. In this database, the subjectivity is noticeable because of the divergence of opinions given by the specialists - not only because of the labels they choose to assign, but also by how they outline the nodules' shape. Figure 4.3 exemplifies how radiologists may trace the contour of the same nodule differently. In this case, the outlines are different because three radiologists consider that the image represents a single nodule, while the fourth one believes these are two distinct nodules.

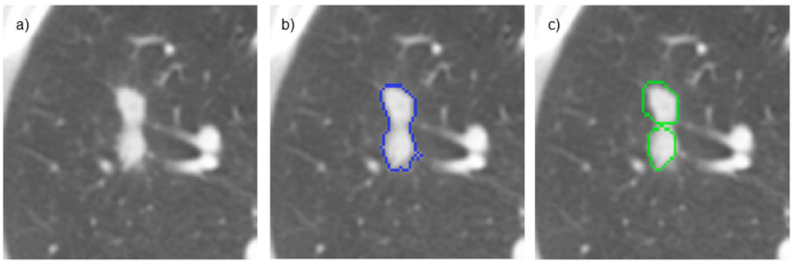


Figure 4.3: Example of lesion considered a single nodule $\geq$ 3 mm by three radiologists and two individual nodules $\geq$ 3 mm by the fourth specialist. Figures b) and c) represent the outlines according to each point of view.

Source: Armato et al. (2011).

Overall, the LIDC is considered a reference database and will be applied in this work to develop pulmonary nodule segmentation algorithms, using conventional and Deep Learning based approaches. A set of 2669 nodular lesions taken from the LIDC database was used, where a single detected nodule can be found per 3D image. Each file also includes the 3D coordinates of the lesion's centroid, its binary mask, and its characteristics. The LIDC segmentation results achieved by the specialists will be used as ground truth reference in this project, so the conventional and Deep Learning based results can be evaluated and compared.

4.4 Data Preparation

The imported files are in a *pickle* format (.pkl), which is meant for object serialization, and each file contains the 3D nodule image, and four binary segmentation masks (one per radiologist). The following procedure was applied in *Python* to all the images in the set; however, for clarity and brevity reasons, the process will be explained for the first nodule in particular.

As mentioned in Chapter 1, the CT scans are built based on cross-sectional tomographic images, often designated as "slices", and yield 3D images. Then, one can analyze the three anatomical planes individually (axial, coronal, and sagittal, exemplified in Figure 4.4), consequently working in 2.5D. This approach simplifies the visualization and interpretation of the images and the results.

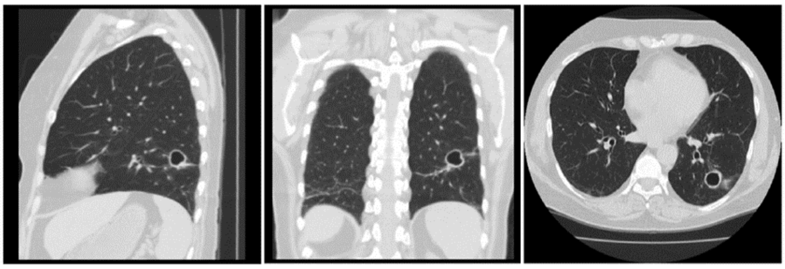


Figure 4.4: Example of a sagittal, coronal, and axial sections of a CT scan.
Source: Malo et al. (2014).

The 3D nodule image was imported, having size 80 × 80 × 80 voxels, along with four binary masks with the same size. The nodule's 3D image was split into the three anatomical planes, resulting in three 2D images, with size 80 × 80 pixels, in Figure 4.5. The intensity values of the nodule's image correspond to HUs. These units were truncated, meaning that only the intensities between -1000 and 400 were considered. By doing so, foreign bodies (such as gold, steel, and copper) are eliminated from the image, and thus it is easier to analyze the nodular structures, which are the focus of this work. Besides truncating the HUs, the intensity values were normalized, and are now within a 0 to 1 range.

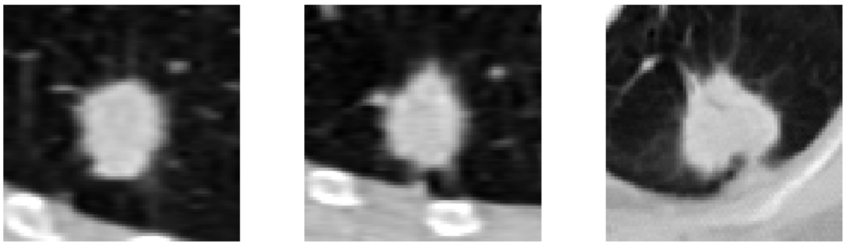


Figure 4.5: Sagittal, coronal, and axial planes of the first nodule.

Since there are four 3D binary masks per nodule, an average of those masks was calculated. However, as the result of that operation was a grayscale image whose intensities ranged from 0 to 1, a threshold of 0.5 was used to convert that image into a binary scale. Once this step was completed, the 3D mask was split into the same three planes, meaning that each nodule has three ground truth binary masks, as shown in Figure 4.6.

The nodule's ground truth center for each plane was calculated by averaging the coordinates of the high intensity values of the mask. Figure 4.7 proves that the nodule's center of mass is near the center of the image, as expected, since the nodules are already detected. All the lesions which did not have a significant size in at least one of the planes (diameter less than 1 pixel) were eliminated, as they generated null ground truth masks, meaning they were not properly centered. For this reason, only 2653 from the set of 2669 nodules were employed in the implemented methodologies.

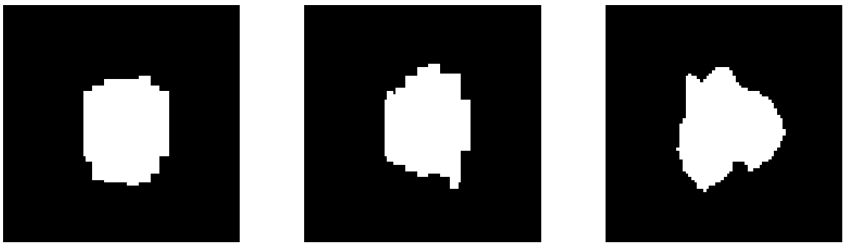


Figure 4.6: Averaged binary masks of the sagittal, coronal, and axial planes of the first nodule, which correspond to the ground truth segmentation.

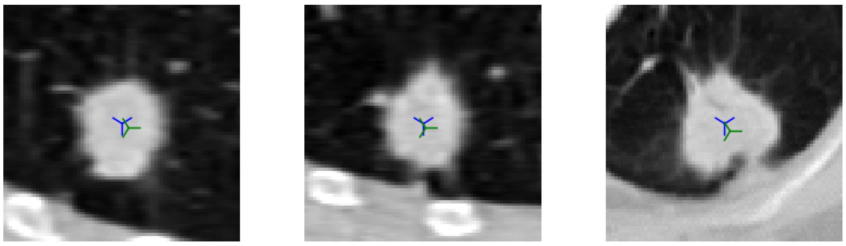


Figure 4.7: Ground truth center of the nodule (green marks), and image center (blue marks), for the sagittal, coronal, and axial planes of the first nodule.

4.5 Summary

This work is based on the LIDC database, which contains thoracic CT scans and their corresponding annotated lesions. Four randomly selected radiologists annotated each lesion and characterized it following a set of common guidelines explicit in this chapter. The annotation process is described in order to understand the database.

There are three types of lesions, nine characteristics analyzed for each one, and the annotations may or may not be consensual. Only the lesions annotated as a nodule greater than or equal to 3mm (greatest in-plane dimension) are used in this work; more specifically, a total of 2653 nodules are used for implementing the methods described in the following chapters, which will be evaluated with the LIDC segmentation results (ground truth). Finally, this chapter also details how the data was prepared in order to be used in such methodologies.

Chapter 5

Conventional Methodology for Nodule Segmentation

Medical image segmentation is able to extract clinically relevant information from images such as CT scans. This work aims to segment pulmonary nodules, a goal which can be interpreted as a classification problem - the classes being nodular or non-nodular. However, segmenting pulmonary nodules can be a challenging task due to the nodule's distinct position, lighting, texture, shape, and background.

As mentioned before, a conventional and two Deep Learning based methods will be implemented. By comparing their results with a segmentation ground truth, this work seeks to evaluate both the performance of the approach based on knowledge representation, as well as the performance of the data-driven model, in this particular segmentation exercise. The theoretical fundamentals and implementation of the conventional approach are presented in this chapter, explaining the Local Convergence Filters, and particularly the Sliding Band Filter.

5.1 Local Convergence Filters

Conventional techniques are based on knowledge representation, which deduces new information using logical rules and a set of beliefs, as defined in Dubois et al. (2000). Their performance does not depend on the amount of data, and so these methods may be successful even when dealing with a small dataset.

Local Convergence Filters (LCFs) estimate the degree of convergence of the gradient vectors within a support region, toward a central pixel of interest, assuming that the studied object has a convex shape and limited size range. The local convergence of a gradient vector at a particular pixel is given by the cosine of the angle between its orientation and the line connecting that point to a central pixel. LCFs aim to maximize the convergence index at each image point, and the overall convergence is obtained by summing all the individual convergences within the support region of the filter.

According to Pereira et al. (2007a), LCFs exhibit a good performance when applied to low contrast images that may or may not have noisy characteristics. Ying-Lun Fok et al. (1996) and Tek et al. (2005) argue that LCFs perform better than other filters because LCFs are not influenced by gradient magnitude (they rely on the gradient vectors' direction), and they are ideal to detect any convex shape, independently of the contrast with the surrounding structures. Other filters only detect a limited range of object shapes, and their output level is influenced by contrast.

LCFs have been previously used in several algorithms for medical image analysis, which aimed, for example, to detect cell nuclei, as proposed in Esteves et al. (2012), detect lung nodules, as proposed in Pereira et al. (2007a), detect cancerous tumours, as stated in Jun Wei et al. (1999), etc. Yet, LCFs can also be applied to achieve segmentation results, as is the case of the optic disc segmentation developed in Dashtbozorg et al. (2015), and the cell segmentation task presented in Quelhas et al. (2010).

The LCFs include several filters with different approaches to estimate the convergence degree of the gradient vectors, such as the Iris Filter (IF), Adaptive Ring Filter (ARF), and Sliding Band Filter (SBF), characterized by their distinct support region, shown in Figure 5.1. The IF maximizes the convergence index by adjusting the radius of its support region in each radial direction, while the ARF uses a ring-shaped support region with fixed width, whose radius changes adaptively but has the same value in each direction. However, the filter chosen for this work is the SBF, which is detailed in the next section.

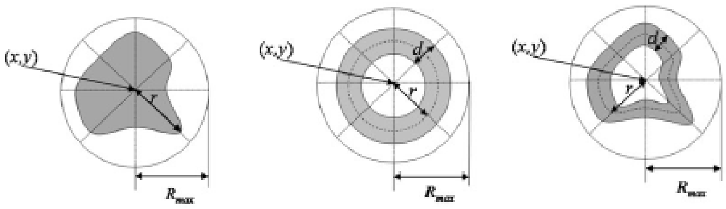


Figure 5.1: Schematics of the support regions for the Iris Filter, Adaptive Ring Filter and Sliding Band Filter, respectively (represented in gray).
Source: Esteves et al. (2012).

5.1.1 Sliding Band Filter

Being a member of the LCFs, the Sliding Band Filter outputs a measure which estimates the degree of convergence of gradient vectors. As Esteves et al. (2012) explain, this particular type of LCFs incorporates principles from the IF and the ARF: the position of the support region is adapted in each radial direction according to the gradient orientation, but, on the other hand, the calculation of the convergence indices is performed minding a support region which is a band of fixed width. In other words, the SBF combines the shape flexibility of the IF with the limited band search of the ARF, while looking for the maximization of the convergence index at each point.

The SBF studies the convergence along a band, ignoring the gradient's behavior at the center (x,y) of the object, when it is considered irrelevant for shape estimation. For this reason, Pereira et al. (2007a) highlight that the SBF takes advantage of a more selective response for nodules which exhibit a distinctively random degree of convergence in their central region - in this case, only the nodule's band with the highest convergence index is considered. By adjusting the band's position, the filter is able to detect different shapes, because the support region can be molded in each direction. Figure 5.2 compares the performance of the three LCFs, in a pulmonary nodule detection task.

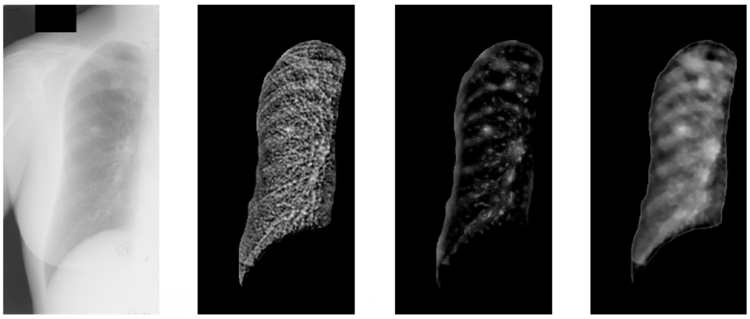


Figure 5.2: Original lung field image, and enhanced images using the IR, ARF, and SBF, respectively.

Source: Pereira et al. (2007a).

This way, it is possible to conclude that the SBF is efficient in the detection of a wide range of shapes, being ideally able to perform well even when not dealing with the anticipated round shape of the nodular structure. This is useful because, as seen before, the shape of the nodules can be very diverse. The SBF is applied in this work as a conventional pulmonary nodule segmentation method, which obtains a set of pixels that maximize the response of the filter, and consequently estimates the boundary of each nodule. The maximization of the response is achieved by adjusting the support region, and therefore deducing the coordinates of the nodule's border.

5.1.2 Mathematical Formulation

Considering an image I as a 2D discrete space, LCFs assess the convergence degree for each coordinate (x,y). To do so, it is necessary to calculate the orientation of the image gradient, represented in Equation 5.1 as $\alpha(x,y,\theta_i,m)$. Such calculation is done within the support region R of the filter, discretized into a set of equally distributed N radial lines emerging from the central pixel of interest. The image should be previously convolved with a Gaussian 2D filter, in order to smooth its noisy characteristics. In this work, a 7×7 kernel was used for this purpose.

$$\alpha(x,y,\theta_i,m) = \tan^{-1}\left(\frac{\partial I(x,y)/\partial x}{\partial I(x,y)/\partial y}\right) \tag{5.1}$$

The polar coordinates within the support region are represented as (θ_i,m); m being measured in pixels, and θ_i resulting from radial sampling. This angle is calculated as shown in Equation 5.2, minding the N radial lines where convergence is evaluated, and where i represents the direction of the reference line.

$$\theta_i = \frac{2\pi}{N}(i-1), \quad i=(1,2,...,N) \tag{5.2}$$

Therefore, according to Esteves et al. (2012), the convergence index for a certain pair of coordinates (θ_i,m) is determined in Equation 5.3 as the cosine between the polar direction θ_i and direction of the gradient; in other words, it can be described as the cosine of the orientation angle $\theta_{i,m}$ of the gradient vector at a certain point, with respect to a line with direction i. This way, if the polar direction and the gradient orientation are superimposed, the angle will be null and the cosine will be maximal.

$$C(x,y,i,m) = \cos\left(\theta_i - \alpha(x,y,\theta_i,m)\right) \tag{5.3}$$

The overall convergence index results from the sum of all convergence values within the support region, considering the fixed width of the band as it moves along a radial length that varies from a minimum (*Rmin*) to a maximum (*Rmax*) values. Consequently, the SBF response for a pixel of interest is given by Equation 5.4, taken from Esteves et al. (2012), where r is the radial length of R, d is the band width (both represented in Figure 5.3), and *Cmax* is calculated as shown in Equation 5.5.

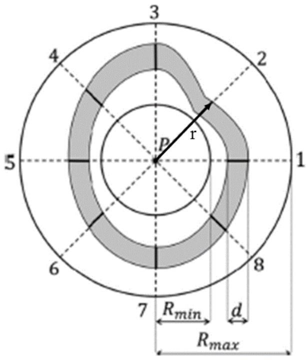


Figure 5.3: Schematics of the Sliding Band filter with 8 support region lines (dashed lines, N=8). Source: Dashtbozorg et al. (2015).

As observed in Equation 5.5, the position of the band is adjusted between *Rmin* and *Rmax*, in a way that the convergence degree is maximal in each of the N directions.

$$SBF(x,y) = \frac{1}{N}\sum_{i=0}^{N-1} Cmax(i) \tag{5.4}$$

$$Cmax(i) = \max_{Rmin \leq r \leq Rmax} \left[\frac{1}{d} \sum_{m=r-\frac{d}{2}}^{r+\frac{d}{2}} cos\theta_{i,m} \right] \tag{5.5}$$

The support points (X,Y) of the filter correspond to the coordinates of the support region and can be interpreted as the boundary of an object. They are estimated for each radial direction using Equations 5.6 and 5.7, described in Dashtbozorg et al. (2015). Both equations assume there is a center candidate (x_c,y_c), and take into consideration the relative coordinates of the support region, $(x_{annulus},y_{annulus})$, whose reference in order to calculate the boundary coordinates for a direction i is the center (x_c,y_c). The radius r_{shape} which maximizes the response of the filter in each direction i is calculated as shown in Equation 5.8.

$$\begin{aligned} X(\theta_i) &= x_c + x_{annulus} = \\ &= x_c + r_{shape}(i)\cos(\theta_i) \end{aligned} \tag{5.6}$$

$$\begin{aligned} Y(\theta_i) &= y_c + y_{annulus} = \\ &= y_c + r_{shape}(i)\sin(\theta_i) \end{aligned} \tag{5.7}$$

$$r_{shape}(i) = \underset{Rmin \leq r \leq Rmax}{argmax} \left[\frac{1}{r} \sum_{m=r-\frac{d}{2}}^{r+\frac{d}{2}} cos\theta_{i,m} \right] \tag{5.8}$$

5.2 Implementation

The algorithm starts with three 80×80 images per nodule in which it is centered, and whose intensities range from 0 to 1. As explained in section 4.4, the blue marks represent the center of the image, while the green marks correspond to the ground truth center of the nodule. For clarity and brevity reasons, two planes from different nodules were selected to explain and exemplify each step of this methodology (Figures 5.4a and 5.4b).

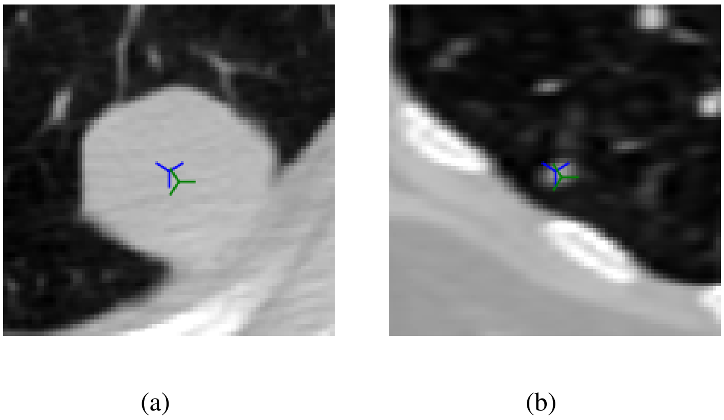

(a) (b)

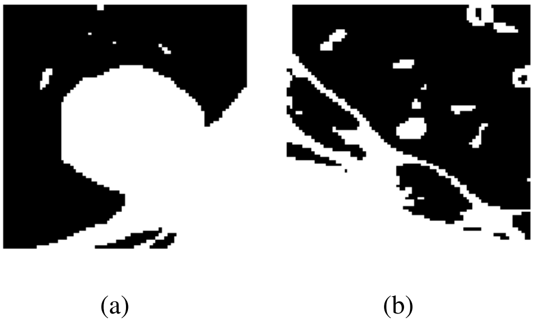
Figure 5.4: Ground truth center of the nodule (green marks), and image center (blue marks).

In order to identify the coordinates which have similar intensity to the lesion's, the nodule's average intensity is determined by calculating the mean of a set of pixels close to the center of the image, more specifically by calculating the mean of the intensity values of a matrix whose center matches the center of the image, as exemplified in Figure 5.5. However, as the nodules have different sizes, several matrices are taken into consideration: a 5×5, 7×7, 9×9, 15×15, and 21×21 matrices. The highest mean corresponds to the average nodule intensity: if the nodule is large, then the largest matrix encompasses more nodular pixels and achieves a more accurate average value; if the nodule is small, the larger matrices encompass background pixels (which have lower intensity values), and so that value will not be selected as the nodule's average intensity.

The selected value is then taken into account to truncate the intensities which are much higher and lower than that nodule's average intensity, originating the truncated masks in Figure 5.6. A closing morphological operation was applied to these masks, using a squared kernel of 1×1 to preserve the edges of the nodules, as their intensity tends to fade. This way, a mask is created for each plane, containing all the pixels with similar intensity to the nodule, and therefore eliminating unwanted structures in the images (e.g. vessels). With these pre-processing steps, there is already a very primitive segmentation, involving a low computational cost, which now needs substantial refinement.

$v_{x-2,y-2}$	$v_{x-2,y-1}$	$v_{x-2,y}$	$v_{x-2,y+1}$	$v_{x-2,y+2}$
$v_{x-1,y-2}$	$v_{x-1,y-1}$	$v_{x-1,y}$	$v_{x-1,y+1}$	$v_{x-1,y+2}$
$v_{x,y-2}$	$v_{x,y-1}$	$v_{x,y}$	$v_{x,y+1}$	$v_{x,y+2}$
$v_{x+1,y-2}$	$v_{x+1,y-1}$	$v_{x+1,y}$	$v_{x+1,y+1}$	$v_{x+1,y+2}$
$v_{x+2,y-2}$	$v_{x+2,y-1}$	$v_{x+2,y}$	$v_{x+2,y+1}$	$v_{x+2,y+2}$

Figure 5.5: Schematics of a 5×5 matrix centered in the image center (blue mark), whose intensities v were averaged to get the nodule's average intensity.



(a) (b)

Figure 5.6: Corresponding truncated masks.

The SBF is first applied to get an estimation of the nodule's center coordinates, matching the position of the maximum filter's response. It is possible to limit this search using a binary mask which indicates the SBF where to look for the center. Such mask, shown in Figure 5.7, includes a 7×7 square centered in the image, and encompasses the nodule's ground truth center.

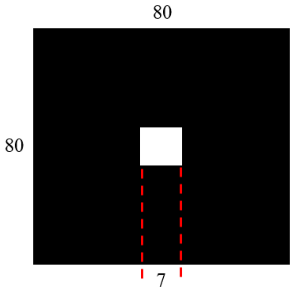


Figure 5.7: Schematics of the search mask used to estimate the nodule's center.

The original nodule image (Figure 5.4) and the corresponding truncated mask (Figure 5.6) are analyzed by the SBF, along with a set of input parameters d, N, $Rmin$, and $Rmax$, defined in section 5.1.2. The filter assumes that the nodule exhibits higher intensities than the background.

As it is intended for this algorithm to be as versatile as possible and able to work with all types of nodules, *Rmin* takes the value of the smallest radius in the LIDC database, while *Rmax* takes the value of the largest radius. It is also convenient to mention that N should be a multiple of 4, as it represents radial directions. Higher values of N and d increase the required computational power, but allow higher sensitivity to the neighboring changes. For this reason, these values were empirically selected in order to maximize the algorithm's performance while keeping the computational cost as low as possible. The selected values for the input parameters are described below:

- $d = 7$
- $N = 64$
- $Rmin = 1$
- $Rmax = 25$

This operation returns the filter's response $SBF(x,y)$ within the area defined by the search mask, and the maximum value of the filter's response in each plane corresponds to the estimated nodule's center. Figure 5.8 indicates the coordinates which maximize the response (red mark).

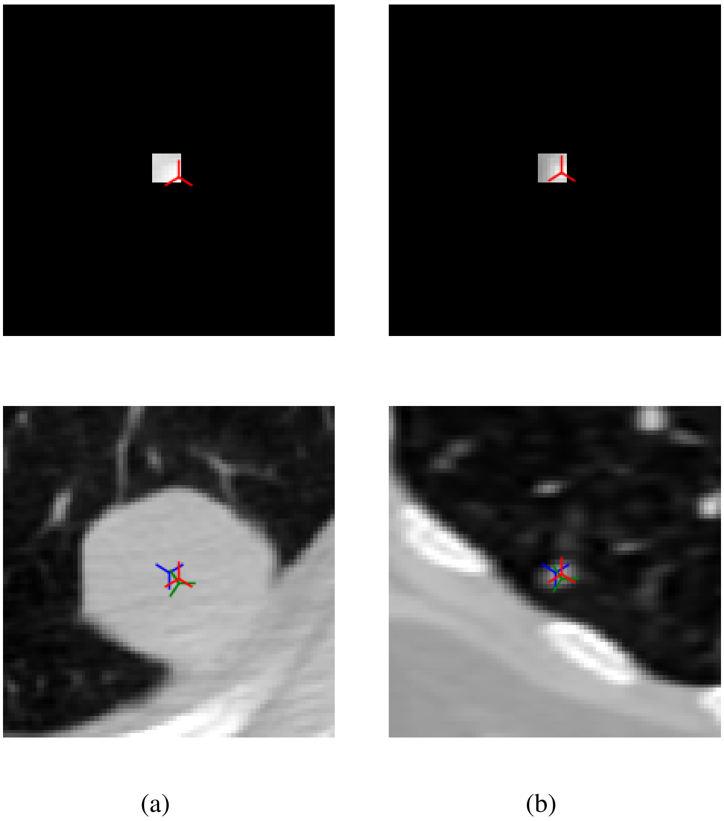


Figure 5.8: Estimated coordinates of the nodule's center in each plane (red marks). On the top row, the center is plotted within the search mask, where the intensities correspond to the filter's response values. On the bottom row, the same coordinates are plotted in the original nodule image.

The SBF then evaluates the corresponding set of N support points, and gets an initial nodule boundary. This step requires the same set of input parameters, nodule image and truncated mask, as well as the coordinates of the estimated center of the nodule.

The truncated mask is of extreme importance for the employment of the SBF, as it forces the cosine of the gradient vector's orientation angle to be zero when the pixel which is being evaluated in a certain direction is null in the mask. Such condition keeps the SBF from encompassing in the segmentation non-nodular regions (e.g. vessels) within the *Rmin* and *Rmax* limits which were successfully eliminated in the mask, thus contributing to a more precise segmentation.

In part of the nodules, the initial SBF segmentation has outliers (abnormal boundary positions). To deal with these, the distance between each boundary coordinate and the previous one was calculated: if that distance is greater than 3 pixels, then that coordinate is replaced by the mean of its six closest neighbors. Figure 5.9 shows the result of the SBF support points (yellow trace), after this outlier removal/attenuation step. The region encompassed by the yellow trace then originates a mask with the corresponding segmentation result.

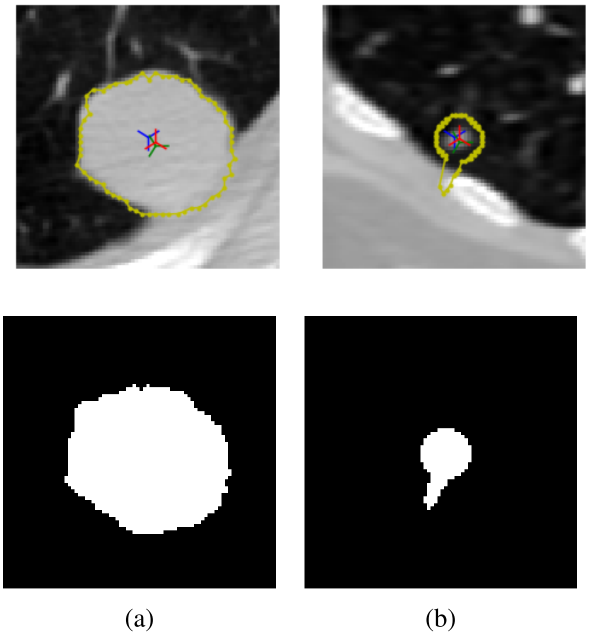

(a) (b)

Figure 5.9: SBF segmentation after outlier removal/attenuation, and corresponding masks.

While the SBF is able to achieve a satisfying segmentation for the nodule in Figure 5.9a, in the nodule in Figure 5.9b part of the pleura is included in the segmented area, since the pleural wall was not eliminated in the corresponding truncated mask and is within the *Rmin* and *Rmax* limits. For this reason, and to further refine the segmentation by specifically selecting the nodular area, only the intersection of the SBF segmentation mask and the truncated nodule mask is considered, thus eliminating unwanted regions such as the pleural wall. Any cavities within the intersected binary masks are filled, and the outcome of this process is exemplified in Figure 5.10.

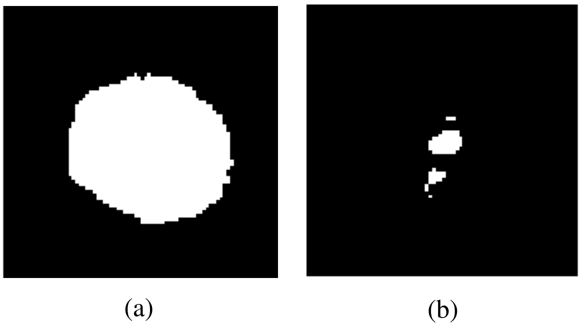


Figure 5.10: Intersected masks.

By labeling all the different regions present in the intersected masks, which are identified by their connected components, it is possible to eliminate any region that has no connection to the nodule, as demonstrated in Figure 5.11. This is done by eliminating from the mask all regions that do not encompass the center of the image. After doing so, the final segmentation masks are achieved. A flowchart of this method can be found in Appendix A.

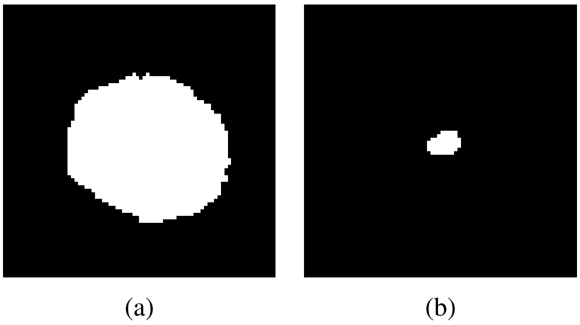


Figure 5.11: Final conventional segmentation result, after deleting non-nodular regions.

5.3 Summary

Local Convergence Filters estimate the convergence degree of the gradient vectors withing a support region, toward a central pixel of interest, while aiming to maximize the convergence degree at each point of an image. The local convergence of a gradient vector at a certain pixel is given by the cosine of the angle between its orientation and a line connecting that pixel to a central one. Local Convergence Filters are applied in the conventional methodology because of their ability to deal with noisy and low-contrast images. Their satisfying performance is partly due to the fact that they analyze the direction of the gradient vectors, not the magnitude, and consequently they perform better than other filters which are influenced by contrast levels.

The Sliding Band Filter, which is a specific type of Local Convergence Filters exhibits a support region which is a band of fixed width and whose position is adjusted in each radial direction to maximize the convergence degree, and therefore, the filter's response. This filter was applied in this work to achieve the segmentation of pulmonary nodules as part of the conventional method, considering its shape flexibility, and the fact that it ignores the gradient's behavior at the center of the nodule. The theoretical fundamentals and mathematical formulation behind the Sliding Band Filter are presented in detail in the current chapter.

The conventional approach was also described, and it consists of four main steps. First, the pre-processed nodule images (three 2D images per nodule) were imported. The Sliding Band Filter aims to estimate an accurate center for the nodule, based on the values of the filter's response in a set of pixels. To get these values, the filter assumed the nodule has brighter intensities than the background. The Sliding Band Filter then used the estimated center to get the corresponding support points, which can be translated as the nodule's border coordinates. Finally, the algorithm seeked to refine this initial segmentation, in order to improve its performance, and resulting in a mask containing the segmented region for each imported image.

Chapter 6

Deep Learning Based Methodology for Nodule Segmentation

The state of the art has been greatly improved when it comes to object detection and segmentation, and overall region recognition thanks to Deep Learning, as mentioned in LeCun et al. (2015). Deep Learning based techniques can be interpreted as data-driven modeling, in the sense that they observe a large amount of data to extract knowledge from it. This work involves semantic segmentation as a pixel-wise segmentation task, and deals with a binary classification problem: a pixel is either nodular or non-nodular. To complete the task, two encoder-decoder style architectures are implemented: the U-Net, and a hybrid between the SegNet and the U-Net, designated as SegU-Net.

6.1 U-Net

CNNs have recently become a trend. The U-Net is a particular CNN developed in Ronneberger et al. (2015), specifically for biomedical image segmentation, whose implementation results in a precise location of the studied subject. The original paper suggested two initial applications for this network: the segmentation of neuronal structures and cell tracking. Other U-Net inspired networks have been proposed since, which tackle e.g. the segmentation of the optic disc, Sevastopolsky (2017), and the segmentation of brain tumors, Dong et al. (2017). Several papers have also used the U-Net to segment pulmonary nodules, namely Aresta et al. (2018) and Tong et al. (2018).

The U-Net can be considered an FCN, in the sense that it does not have any fully connected layers, and so it requires a smaller amount of parameters and computing time. However, it is more accurately described as an improved FCN, as it is able to work with less training data and still achieve a great performance. In other words, the U-Net deals with the need for a large amount of annotated data through data augmentation, which is frequently used in biomedical segmentation tasks because it allows to simulate realistic deformations in tissue. More specifically, this network applies elastic deformations to the available training images, and consequently enhances its performance even with a limited set of labeled training samples.

The U-Net, which is an example of an encoder-decoder architecture for semantic segmentation, includes a contracting path and an expansive path. The contracting path (represented with red frames in Figure 6.1) is also known as encoder, and can be seen as a common convolutional network with several convolutional layers, each followed by a ReLU and a max pooling layer. This path downsamples the input image into feature representations with different levels of abstraction (low to high level features), designated as feature maps. In other words, the encoder produces coarse contextual information, as it gathers knowledge about the features of the input, while at the same time shortening spatial information. Dropout layers are included throughout the network to improve the robustness of the algorithm by avoiding overfitting.

On the other hand, the expansive path (blue and green frames in Figure 6.1) receives the information gathered by the contracting path and takes advantage of a series of upconvolutions and concatenations to perform upsampling. This way, the feature and spatial knowledge from the contracting path is associated with the higher resolution layers of the expansive path, by the means of skip connections. This causes the expansive path to be symmetrical to the contracting one, resulting in a U-shaped network.

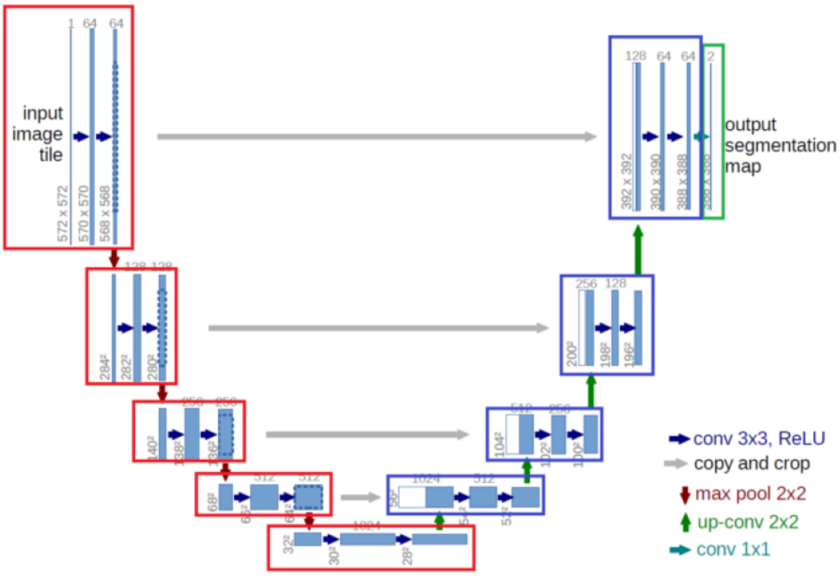


Figure 6.1: U-net architecture. Each blue block resembles a multi-channel feature map (the number of channels is denoted on top of the block), while the white blocks are copied feature maps. The x and y dimension values are written at the lower left edge of the block. Each arrow represents an operation. The red frames indicate the contracting path, and the blue and green frames represent the expansive one. The green frame is the last layer of this network, which gives the output segmentation map.

Source: Ronneberger et al. (2015).

It is essential to highlight the skip connections present in the U-Net (grey arrows in Figure 6.1), through which the output of one layer is fed to another, skipping intermediate layers. This is necessary for the concatenations, which use information captured in the initial layers and the result of the upconvolution layers to finally obtain comprehensive knowledge combining localization and context. The skip architecture makes sure that the initial information is not lost (i.e. does not become more abstract as the network moves on to deeper layers), and that the concatenation joins lower and higher level features.

The efficiency of the U-Net is mainly due to that combination of the coarse contextual information from the downsampling part with more precise spatial information achieved in the upsampling part, leading to satisfactory segmentation maps.

6.1.1 Specifications

Ronneberger et al. (2015) propose a network in which the contracting path includes four sets of two 3×3 unpadded convolutions, each followed by a ReLU with batch normalization, a 2×2 max pooling layer with stride equal to 2, and a dropout layer. The number of feature channels is doubled with each downsampling operation, repeated four times as mentioned above. Two 3×3 convolutions with ReLU and batch normalization are present at the end of this path, creating a bottleneck before the following path.

The expansive path is comprised of four blocks which repeatedly have an upconvolution layer with stride equal to 2 (reducing the number of feature channels by half), a concatenation of that output with the corresponding cropped feature map from the contracting path, dropout and two 3×3 convolutions, each followed by ReLU with batch normalization. The decoder layers start with a large number of feature channels, which promote an accurate segmentation. In order to achieve the segmentation result (classification task with two classes), the endmost layer includes a 1×1 convolution, using pixel-wise softmax activation.

The network was originally trained with the Stochastic Gradient Descent (SGD) as optimizer, minimizing the loss function by computing the gradient of a subset of samples, and thus requiring less computational power in comparison to the standard Gradient Descent, which uses all samples - this is particularly useful when dealing with large data sets. In this paper, a cross-entropy loss function was employed, also known as Log Loss, evaluating the performance of a classification model whose output is a probability value between 0 and 1 per pixel.

6.1.2 Implementation

The Deep Learning algorithms presented in this work were implemented using *Keras*, with a *TensorFlow* backend. The data was prepared as mentioned in section 4.4, resulting in 7959 images from a total of 2653 nodules, which are imported and randomly split into training, validation, and test sets. A condition was added to ensure the same nodule is not present in multiples sets, meaning that the three nodular planes must belong exclusively to one, as it would not make sense to train and test the model using the same nodules. This way, the set of 2653 nodules was first split into training and test sets (80% - 20%, respectively), and then 20% of the training set was used as a validation set. Real-time data augmentation is applied to the training set, replicating tissue deformations through affine transformations (e.g. 0.2 degrees of shear and random rotation within a 90 degree range), and generating more training data with horizontal and vertical flips. The mentioned values are considered standard and were selected after testing several options. Grayscale variations were also initially considered, but discarded since they proved to reduce the model's performance, which became maximal with soft data transformations.

The architecture of the U-Net was kept as proposed in Ronneberger et al. (2015), whose contracting and expansive paths are described in section 6.1.1. In this work, the pixel-wise probabilities that resulted from the sigmoid activation faced a 50% threshold to decide whether a pixel is nodular or not. The sigmoid function was selected as a more adequate option, in opposition to the softmax, because this is a binary exercise; the softmax would be more suitable for a multi-class task.

The network was trained with the ADAM (Adaptive Moment Estimation) optimizer. Despite being faster than the standard gradient descent, the SGD leads to oscillations in the loss function due to a high variance in the frequent parameter updates. Such fluctuations hinder the convergence to the best local minima. On the other hand, the ADAM uses adaptive learning to compute individual learning rates for each parameter. It takes advantage of the *momentum* technique, which accelerates and navigates the optimizer along a relevant direction, by accumulating the gradient of the past steps to determine which direction to follow. The parameter updates are only done for relevant examples, and by reducing the frequency of these updates, it is possible to achieve faster and stable convergence, and consequently reduce the fluctuations of the loss function.

The loss function used in this work is also different than one proposed in section 6.1.1. It was necessary to take into consideration the class unbalance within a sample (generally, there are more non-nodular pixels in an image than nodular ones), and so a Dice based loss function was selected instead of the cross-entropy loss function. The training stage of the model is guided with two evaluation metrics: accuracy and Jaccard Index. Such metrics are explained in detail in Chapter 7.

While training the model, callbacks were included. First, early stopping ensures the training ends when the validation loss stops improving. At the same time, the learning rate also reduces on plateau, meaning that it is reduced when the validation loss cannot reach a lower value. More specifically, the initial learning rate value, 0.001, is the default provided in the original paper for the ADAM optimizer, Kingma and Ba (2014), and will be reduced by a factor of 0.1 (new learning rate $=$ learning rate $\times$ 0.1), having a minimum accepted value of 0.00001. The value of the *momentum* is also the default provided in the paper mentioned above, i.e. 0.9.

The model is fit on batches with real-time data augmentation, using a batch size of 64 samples and allowing a maximum of 100 epochs. The model's weights that maximize the evaluation metrics and minimize the loss are stored, to get the predictions of the test set.

6.1.2.1 Remarks on Training Conditions and Hyperparameters

In a Deep Learning process, it is necessary to set the hyperparameter values before the learning stage starts. Among them, it is necessary to establish the batch size and the number of epochs. These hyperparameters are important because most of the times it is not possible to feed the whole training set to the model, due to memory limitations. This way, it is common to split the data into batches which are able to fit the memory of the computer at once. Then, it becomes intuitive to feed each of these batches individually to the model as it is trained. When all batches are fed once, one epoch is completed. In other words, one epoch means the model has seen the whole training set exactly once.

In general, the model tends to improve with a higher number of training epochs. However, after a certain number of epochs, its accuracy will plateau and this is why callbacks such as early stopping are so important. The value described in the previous section - 100 epochs - is a standard value which in this case allows the model to converge early on.

The batch size value is usually a power of two, in order to align the virtual processors and the physical processors of the Graphics Processing Unit (GPU). Without getting into finer details on this subject, considering the number of physical processors is frequently a power of two, not using a power of two for the virtual processes may lead to a worse performance.

According to Keskar et al. (2016), large batches are associated with a degradation of the model's quality, meaning they are associated with poorer generalization. On the other hand, small batches do not work well with complex data, and may impose longer computational time. For this reason, it is important to find an adequate value which reduces the variance of the model, thus avoiding overfitting to the training data, while at the same time also avoiding a long training stage. This way, three power of two values were studied: 32, 64, and 128. As a batch size of 64 increased the model's performance, it was defined as the final value.

Dropout is another simple way to prevent neural networks from overfitting, and usually assumes a value between 0.25 and 0.50, depending on how much the network is likely to overfit. As the model does not have a great amount of data to be trained with, considering the total dataset is only comprised of 2653 nodules, all efforts were made to avoid such situation, and so a dropout of 0.50 was selected after testing several values (0.25, 0.40, and 0.50).

6.2 SegU-Net

The SegNet is an FCN proposed in Badrinarayanan et al. (2015) for pixel-wise semantic segmentation. Its architecture, presented in Figure 6.2, was designed to be efficient in terms of memory and computational time during inference, in comparison to other FCNs, as is the case of the structure proposed in Long et al. (2014), introduced in Chapter 3.

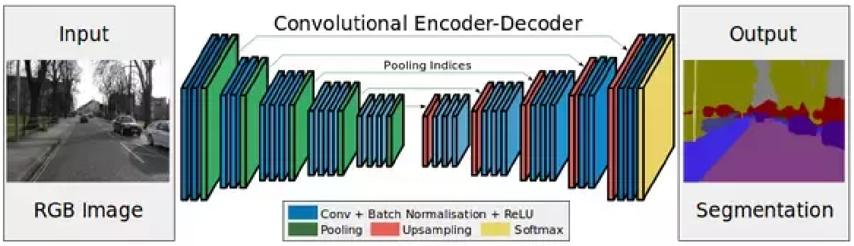


Figure 6.2: Illustration of the SegNet's architecture.
Source: Badrinarayanan et al. (2015).

The architecture of the SegNet can be described by its encoder network, and its decoder network, ending in a final pixel-wise classification layer, whose output is a segmentation mask. As mentioned earlier, the encoder network extracts low-to-high level features, thus estimating what objects are present in the input image, and roughly locating them. The encoder contains thirteen 3×3 convolutional layers from the VGG16 network in Simonyan and Zisserman (2014), each followed by batch normalization and a ReLU activation layer. Several non-overlapping 2×2 max pooling layers are present in this path.

Then, the decoder network receives the high-level features and attributes a precise pixel location to the object(s). Each encoder layer has a corresponding decoder layer, hence resulting in thirteen decoder layers, which are also comprised of 3×3 convolutional layers followed by batch normalization and a ReLU activation layer. This way, the decoder takes the output of the final encoder layer and runs it through certain sets of upsampling and convolutional layers, which add detail to the feature map to compensate for the loss that happens at the encoder's pooling layers.

The non-overlapping (i.e. stride equal to 2) 2×2 max pooling operations in the encoder path cause only the maximum value within that 2×2 region to be selected and propagated to the next layer, as exemplified in Figure 6.3. When the decoder performs an upsampling operation, it is basically performing the opposite operation. In other words, it is "restoring" a 1×1 region into a 2×2 region again. Furthermore, in this restored 2×2 region, one pixel will take the value of the 1×1 region, while the others will be null. However, where to allocate that value poses a relevant question, as assigning it to a fixed position or in a random way would introduce error in the consecutive layers, therefore reinforcing the importance of proper placement.

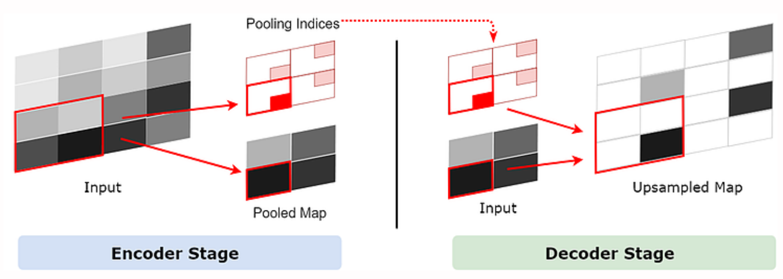


Figure 6.3: Downsampling and upsampling using max pooling indices.
Source: CyberAILab (2018).

The novelty of the SegNet is related to the way it places that max pooled value while upsampling. During the encoder stage, the index (i.e. location) of the maximum valued feature in each 2 × 2 area is stored. Consequently, and considering the symmetric architecture, the decoder receives those max pooling indices to perform the upsampling and the value in the 1 × 1 region is assigned to the corresponding initial max pooling index, thus being assigned to the same initial position. This process is exemplified in Figures 6.3 and 6.4.

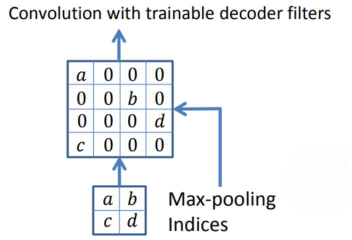


Figure 6.4: Example of an upsampling operation in SegNet.
Source: Badrinarayanan et al. (2015).

After the non-linear upsampling, denser maps are obtained by convolving the sparse upsampled maps with trainable filters. Finally, after a 1 × 1 convolutional layer, a softmax classifier takes the information collected by the previous layers and outputs the segmentation results. The proposed network was trained using a cross-entropy loss function and a SGD optimizer.

6.2.1 SegNet vs. U-Net

Both the U-Net studied in the previous section and the SegNet are FCNs with encoder-decoder structures meant for pixel-wise semantic segmentation. They use convolutional and activation layers to extract local features, pooling to downsample feature maps and propagate spatially invariant features to deeper layers, and batch normalization to normalize the distribution of the training data and therefore accelerate the learning process.

While the U-Net was initially developed for biomedical applications, the SegNet was primarily motivated by outdoor and indoor scene understanding. However, it has also been employed in several biomedical applications, as it is the case of the pulmonary tuberculosis detection system evaluated in Lee et al. (2017), the segmentation of cross-sectional brain MRI in Khagi and Kwon (2018), and the brain tumor segmentation on multi-modal MRI in Alqazzaz et al. (2019).

The skip connections at the same depth level in the U-Net are an effective way to transfer low-level information between the encoder and the decoder, which is receiving that coarse contextual information. Besides that, the fact that the number of feature maps doubles at each max pooling layer in the contracting path also contributes to capturing better spatial context of the input images. Since the entire feature maps are transferred from the encoder to the decoder to be concatenated, the U-Net's model becomes larger, meaning it requires more memory. On the other hand, while the SegNet also doubles the number of feature maps, it does not have any skip connections; instead, it uses the max pooling indices to perform max unpooling. Hence, the main difference between both architectures can be briefly described by the way they perform the upsampling: the U-Net uses upconvolutions and concatenations, while the SegNet performs max unpooling, using the indices from the max pooling stages (this way, it does not need to employ upconvolutions or concatenations). Citing Badrinarayanan et al. (2015), the main advantages that come with this second upsampling approach are an improvement of boundary delineation, associated with a memory-wise efficiency and good time performance. Nevertheless, it is also known that the benchmarks for the SegNet are not satisfactory anymore.

Recently, hybrid networks have appeared, where the SegNet and the U-Net are combined to achieve a more memory-efficient model, which is also able to capture fine details thanks to the skip connections inspired by the U-Net. That is the case of the hybrid networks proposed in Do Nhu et al. (2019) for knee bone tumor segmentation, and in Kumar et al. (2018) for brain tissue segmentation - both these algorithms performed better than the SegNet and the U-Net individually. Do Nhu et al. (2019) add that such architecture prevents overfitting during training due to the batch normalization and the upsampling technique from the SegNet, as it is based on local information received from the encoder (i.e. indices). Minding the reasons presented above, a hybrid network designated as SegU-Net was developed and applied to the same data as the U-Net, expecting to possibly achieve higher performance scores.

6.2.2 Specifications

The SegU-Net takes advantage of the SegNet and the U-Net, combining both into a new architecture dedicated to restoring pixel position information, and ideally achieve finer edge details. The contracting path has four blocks, and starts off with the original input images. Each of these blocks is comprised of two 3×3 convolutional layers, each followed by batch normalization and ReLU; after the convolutions, there is a 2×2 max pooling layer which stores the max pooling indices, and dropout. The number of feature channels doubles when a new block begins. After the four blocks, there are two convolutional layers with batch normalization and ReLU, in which the number of feature channels remains the same as in the last block, to ensure the consistency of the layers' dimensions when upsampling.

The expanding path is comprised of four blocks and a final convolutional layer. In each block of the decoder, a max unpooling layer is concatenated with the result of the corresponding layer from the encoder. The 2×2 max unpooling is done using the indices stored early on, and the concatenated result is followed by dropout and two 3×3 convolutional layers with batch normalization and ReLU. In the decoder, the number of feature channels in the convolutional layers of each block matches the number of channels of the corresponding encoder layer - for this reason, this number is consecutively halved after each block. The final 1×1 convolutional layer has a sigmoid activation to output a segmentation map.

Based on this description, one can infer that the SegU-Net is based on the U-Net structure presented in the previous sections, but at the same time incorporating the SegNet's upsampling approach. In other words, instead of upsampling using upconvolutions, the SegU-Net uses max unpooling (i.e. minding the max pooling indices) and keeps the U-Net's concatenations to gather more spatial information about the features. Figure 6.5 outlines the SegU-Net's model, and a more detailed scheme can be found in Appendix A.

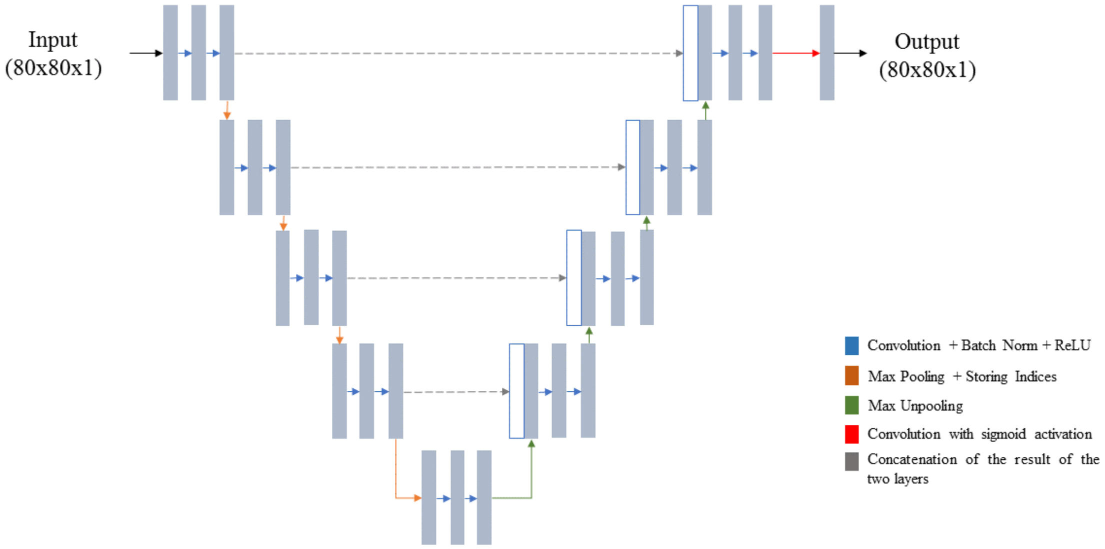

Input
(80x80x1)
Output
(80x80x1)
🟦 Convolution + Batch Norm + ReLU
🟧 Max Pooling + Storing Indices
🟩 Max Unpooling
🟥 Convolution with sigmoid activation
⬛ Concatenation of the result of the two layers

Figure 6.5: Scheme of the SegU-Net's model.

6.2.3 Implementation

Similarly to the procedure described in section 6.1.2, the SegU-Net was trained using 80% of the dataset (from which 20% is kept as a validation subset), and tested with the remaining 20%. This split was done randomly and ensuring no nodule belongs to more than one set.

The model complies with the architecture detailed in section 6.2.2, and was trained minding a Dice based loss function, guided by the ADAM optimizer, and the selected evaluation metrics (Jaccard Index and accuracy). In comparison to the U-Net, the same training conditions were included, when it comes to callbacks and associated hyperparameters. A batch size of 32 was selected after a brief study revealed it to be the most suitable batch size among a list of three possible values (32, 64 or 128), considering it maximizes the SegU-Net's performance. For the same reason, a dropout of 0.50 was used, after testing several values (0.25, 0.40, and 0.50). This way, the model was fit on batches of 32 samples, with real-time data augmentation using the same criteria disclosed in section 6.1.2 (affine transformations and flips), and allowing up to 100 epochs. The predictions were then made based on the weights that minimized the loss function and maximized the evaluation metrics.

6.3 Summary

This chapter presents in detail the two Deep Learning methodologies employed in this work, so they can be later compared to the conventional technique presented before. The Deep Learning approaches look at this semantic segmentation task as a pixel-wise classification, where a pixel is either nodular or non-nodular (binary classification), and they require a large amount of input data to learn from it.

Several Deep Learning models could have been implemented to achieve this goal. However, this work focuses on FCNs with encoder-decoder architectures. Here, the encoder takes the input (in this case, a grayscale image) and outputs a feature map, thus capturing the context of the input image. Along the encoder network, the image and the feature maps are progressively downsampled using pooling layers, and so the spatial information is reduced. The decoder network is usually symmetric to the encoder (i.e. same number of layers), and performs the opposite operation: it receives the feature maps and consecutively upsamples them, until the output results in a segmentation map with the same resolution as the initial input image. To do so, it includes a classifier in the last layer, to attribute each pixel a probability of it being nodular, which is then thresholded to make a final decision.

The first model to be employed was the U-Net, an FCN specifically developed for biomedical image segmentation, able to deal with the large size of medical images, as well as the limited amount of available data. It uses data augmentation to simulate the elastic deformations of the tissues, and at the same time generate more training data. Its contracting path is constituted by convolutions, ReLU, batch normalization, max pooling (to downsample), and dropout. The expansive path then uses upconvolutions (to upsample), concatenations, convolutions, ReLU, batch normalization, and dropout. The skip connections in the U-Net are one of its most important aspects: they allow the output of one layer to be fed to a non-consecutive layer. This is essential for the concatenations, where the contextual information captured in the contracting path is combined with the result of the upconvolutions in the expansive path, thus obtaining comprehensive information from the association of localization and contextual knowledge and contributing to the U-Net's efficiency.

Similarly to the U-Net, the SegNet is also an encoder-decoder network, whose novelty lies in the way it performs upsampling, which differs from the U-Net. When the contracting path consecutively downsamples the images, the indices of the max pooling values are saved. When upsampling a 1×1 region to a 2×2 one in the expanding path, the value within that 1×1 region is placed in the same index/location which was previously stored. The SegU-Net model proposed in this work starts off with the U-Net structure and replaces the original upsampling method (up-convolutions) by the max unpooling method suggested in the SegNet paper, while keeping the U-Net's concatenations. This way, the U-Net is upgraded, and it is expected that by doing so the model's knowledge on localization is improved, consequently enhancing its performance. All the details on the U-Net and SegU-Net's implementation, training conditions, and hyperparameters are described in this chapter.

Chapter 7

Results and Discussion

The implementation of segmentation methods in a medical scope can be a great contribution to assess the patient's health, as e.g. they determine the detection and monitoring of a tumor. This work aims not only to segment pulmonary nodules using several techniques, but also compare the performances of the implemented algorithms, thus drawing conclusions about their quality and effectiveness.

It is necessary to compare the results achieved by each technique to the ground truth images. Such task can be seen as a way of measuring the similarity between both segmentation masks, which by itself is considered a challenging exercise, as explained in Monteiro and Campilho (2012). Furthermore, Taha and Hanbury (2015) enumerate some of the adversities encountered when it comes to evaluating medical segmentation: not only choosing which metrics to use, but also the fact that occasionally there are multiple definitions for the same metric in the literature. Additionally, metrics can be sensitive to outliers, class imbalance, and the number of segmented objects.

In this chapter, the results for each algorithm are presented and discussed using quantitative evaluation metrics, resulting in a score which translates to how well the algorithm performed, and thus providing a means of comparison. As mentioned in Chapter 4, all outcomes are compared to the ground truth segmentation, obtained from the average of the four segmentation masks given by the specialists.

7.1 Evaluation Metrics

In order to compare the results achieved by both the conventional and the Deep Learning based segmentation techniques, it was necessary to define a specific evaluation system, explained below, which standardizes the metrics that will be used and their definition.

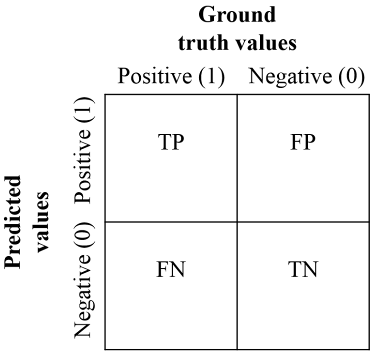
Considering there is a ground truth reference, the confusion matrix represented in Figure 7.1 can be used to evaluate the performance of the algorithm. It comprises the number of True Positives (TPs), False Positives (FPs), False Negatives (FNs), and True Negatives (TNs), defined in Esteves et al. (2012) as follows:

(a) True Positive — region detected as nodular for which there is a corresponding ground-truth.

(b) True Negative — region detected as non-nodular for which there is a corresponding ground-truth.

(c) False Positive — region detected as nodular for which there is no corresponding ground-truth.

(d) False Negative — region detected as non-nodular for which there is no corresponding ground-truth.



		Ground truth values	
		Positive (1)	Negative (0)
Predicted values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 7.1: Confusion matrix.

From these values, other evaluation metrics can be deduced, such as Sensitivity and Precision (Equations 7.1 and 7.2, respectively). In the mentioned equations, taken from Shakibapour et al. (2019), G represents the ground truth segmentation and S represents the segmentation being evaluated. Sensitivity is also known as True Positive Rate and Recall, and measures the proportion of positive pixels in the ground truth which are correctly identified as positive in the segmentation being analyzed. On the other hand, Precision (or Positive Predictive Value) analyzes the proportion of the values predictive as positive that are equally positive in the ground truth. While Precision expresses the portion of positive/relevant samples among the retrieved set of occurrences, Sensitivity is interpreted as the portion of relevant samples correctly assigned from a set of all truly relevant occurrences. As stated in Shakibapour et al. (2019), Sensitivity and Precision close to 1 imply a high level of agreement between the segmentation result and the ground truth.

$$\text{Sensitivity} = \frac{|G \bigcap S|}{|G|} = \frac{TP}{TP+FN} \tag{7.1}$$

$$\text{Precision} = \frac{|G \bigcap S|}{|S|} = \frac{TP}{TP+FP} \tag{7.2}$$

Although Sensitivity and Precision are very common evaluation metrics used in several studies, more elaborated metrics can be used. That is the case of the Dice Coefficient (also designated as F1 Score), a very common metric for validating medical image segmentation, which not only establishes a direct comparison between the achieved and the ground truth segmentation, but is also able to measure reproducibility/repeatability, according to Taha and Hanbury (2015). The Dice Coefficient takes into consideration both Precision and Sensitivity in a harmonic mean, defined as written in Equation 7.3. It ranges from 0 to 1, the last corresponding to a perfect segmentation.

$$\begin{aligned}\text{Dice Coefficient} = 2\,\frac{\text{Precision} \times \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} = \\ = 2\,\frac{|G \bigcap S|}{|G| + |S|} = \frac{2TP}{2TP + FP + FN}\end{aligned} \tag{7.3}$$

The Jaccard Index, or Intersection Over Union (IOU), also ranges from 0 to 1 and is defined as the intersection between two sets divided by their union (Equation 7.4). Through Equation 7.5, it is possible to understand how the Jaccard Index and the Dice Coefficient can be related. Both metrics take into account incorrectly classified pixels (False Positives and False Negatives), but the Jaccard Index penalizes those instances more than the Dice Coefficient. For this reason, it was decided to employ both, as well as Precision and Sensitivity, considering that the Dice Coefficient resembles more of an "average score", while the Jaccard Index translates to a "worst case performance". In the next sections, the results of the algorithms are presented and discussed.

$$\text{Jaccard Index} = \frac{|G \bigcap S|}{|G \bigcup S|} = \frac{TP}{TP + FP + FN} \tag{7.4}$$

$$\text{Jaccard Index} = \frac{\text{Dice Coeff}}{2 - \text{Dice Coeff}} \tag{7.5}$$

7.2 Results

Once the evaluation metrics were established, each method was evaluated using the ground truth segmentation masks from the LIDC database. Table 7.1 presents the maximum, minimum, and mean values achieved for each metric (Dice, Jaccard, Precision, and Sensitivity), which assessed the totality of the 2653 nodules in the conventional approach, and a test set of 531 nodules in the U-Net and the SegU-Net (20% of the dataset). The scores achieved in the U-Net using the test set are associated with a loss value of 0.172 and accuracy of 0.991, while in the SegU-Net the loss is 0.177 and the accuracy is 0.992.

Furthermore, to understand the efficacy of the methodologies, Tables 7.3 and 7.4 exhibit the individual Dice coefficients obtained for the nodular characteristics studied in Chapter 4. To ensure a fair comparison, the values presented in the mentioned tables correspond to the scores achieved for the same set of nodules, meaning that the performance of the SBF was compared individually to each Deep Learning technique, minding exclusively the 531 nodules that comprised the test set in that Deep Learning approach.

Table 7.1: Evaluation metrics achieved by the conventional and Deep Learning based methods.

		Dice	**Jaccard**	**Precision**	**Sensitivity**
SBF	Max	0.940	0.887	1.000	1.000
	Min	0.032	0.024	0.024	0.029
	Mean	**0.663**	**0.500**	**0.709**	**0.732**
U-Net	Max	0.967	0.938	1.000	1.000
	Min	0.366	0.286	0.328	0.307
	Mean	**0.830**	**0.712**	**0.792**	**0.898**
SegU-Net	Max	0.926	0.962	1.000	1.000
	Min	0.379	0.250	0.250	0.391
	Mean	**0.823**	**0.702**	**0.787**	**0.858**

The timings of each methodology were also taken into consideration. The SBF needed approximately 5 hours to run, while the U-Net had a reasonable training time of 8 hours on a NVidia GeForce GTX 1080 GPU (8 GB). Using the same GPU, the SegU-Net recorded a similar training time, more specifically of 8,5 hours. Such values meet the expectations, as Deep Learning methods usually require longer periods to run, in comparison to conventional ones.

Figure 7.2 shows the training and validation loss, as well as the respective scores, achieved for the U-Net's training stage. This model exhibited fast convergence (after 14 epochs), and did not overfit to the training data, considering the validation loss is similar to the training loss (difference in the order of centesimal places). The same can be concluded for the SegU-Net model, which converged after 24 epochs as shown in Figure 7.3, and did not show any signs of overfitting. The evolution of these models' learning rates is exhibited in Figure 7.4.

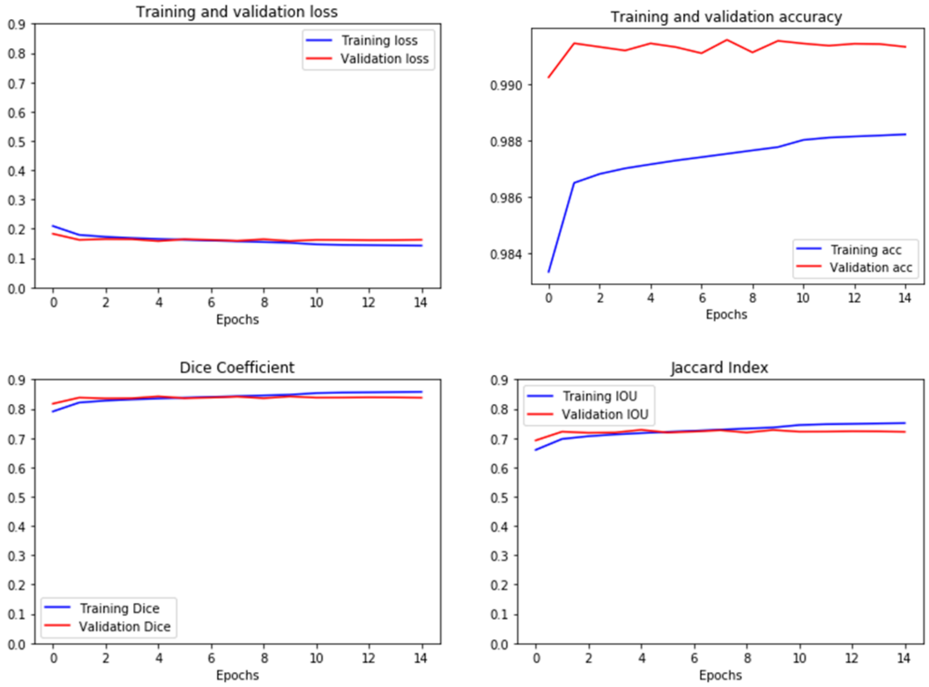


Figure 7.2: Training and validation results of the U-Net model.

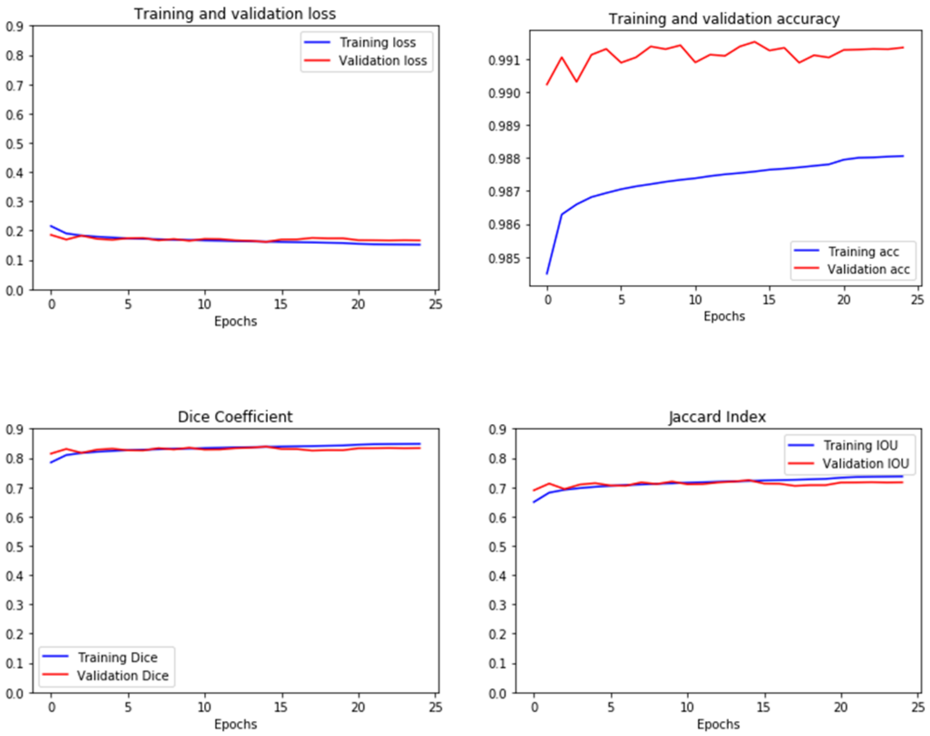
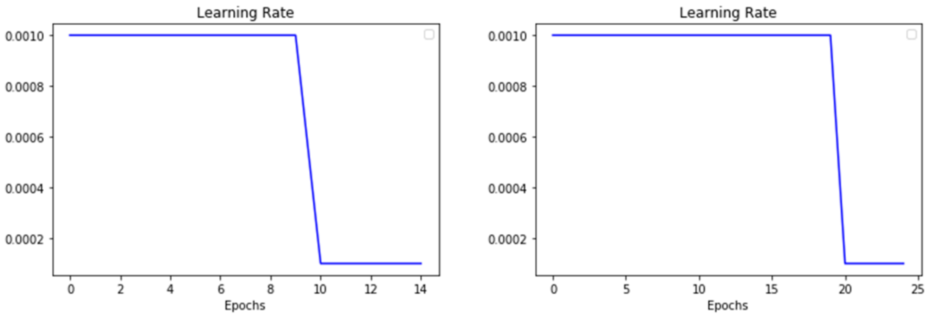


Figure 7.3: Training and validation results of the SegU-Net model.



(a) (b)

Figure 7.4: Learning rate per epoch in the U-Net and SegU-Net models, respectively.

7.2.1 Remarks on Data Representation

Several experiments were carried out during this work to define not only the most suitable hyper-parameter values, but also to establish the appropriate percentages to split the data into training and test sets, as it is intended to maximize the models' performance. In all occasions, there were no nodules which belonged to more than one set, which was an important aspect to take into consideration.

The first attempt was to train and validate the model using 20% of the data, and then test it with the remaining 80%. These percentages may seem unusual, since Deep Learning models often require a huge amount of training data to learn efficiently. However, when comparing the performance using these values to the one presented in Table 7.1, which uses 80% of the data to train and 20% to test (stated in section 6.1.2), it is possible to conclude that there is a very narrow improvement in the evaluation metrics. This reveals that in this case the 20% of the nodules which were randomly selected in the initial approach as the training set are representative of the whole dataset. In other words, the samples present in that 20% subset are an accurate unbiased reflection of the entire dataset, and consequently are enough for the Deep Learning model to learn how to proficiently segment the nodules. Table 7.2 shows the values which led to this conclusion in the case of the U-Net model, as it shows the difference of the score values is in the centesimal places. However, in spite 20% of the nodules apparently being enough to get satisfying segmentation masks, it was decided to present in the results the model which scores the highest values. Hence, a 80%-20% split intro training and test sets was chosen. The test set ensures a stratified sampling, i.e. the original proportions of the different nodular types in the LIDC (refer to Table 4.3 in Chapter 4) are kept in the test data, once again contributing to a correct representation of the data, as well as a fair evaluation of the model. For brevity reasons, only the study made for the U-Net is demonstrated in these remarks, but the same was concluded for the SegU-Net.

Table 7.2: U-Net's evaluation metrics, depending on the data percentages used to train and test the model, minding the same hyperparameters.

	Dice	**Jaccard**	**Precision**	**Sensitivity**
20% training, 80% test	0.802	0.674	0.793	0.846
80% training, 20% test	0.830	0.712	0.792	0.898

To illustrate the performances of the methods presented in this work, several examples were selected and are displayed in the following section, where the results are discussed. Figures 7.5, 7.6, and 7.7 show two nodules in which the SBF had a highly satisfactory, less satisfactory, and unsatisfactory performance, while Figures 7.8, 7.9, and 7.10 show two nodules in which the U-Net had a highly satisfactory, less satisfactory, and unsatisfactory performance, respectively. The same was done for the SegU-Net in Figures 7.11, 7.12, and 7.13. To interpret these results, a comparison plot was created, through which it is possible to understand the differences between the achieved and the ground truth segmentation masks. In this plot, the green pixels belong exclusively to the ground truth mask (False Negatives), the red pixels belong exclusively to the achieved result using the proposed method (False Positives), and the yellow pixels belong to both - meaning that the yellow pixels mark the correct predictions made by the algorithm (True Positives).

Table 7.3: Dice coefficients for each nodular characteristic, achieved by the SBF and the U-Net, evaluated in the same set of 531 nodules.

Scoring	1		#Nods	2		#Nods	3		#Nods	4		#Nods	5		#Nods	6		#Nods
	SBF	**U-Net**		**SBF**	**U-Net**		**SBF**	**U-Net**		**SBF**	**U-Net**		**SBF**	**U-Net**		**SBF**	**U-Net**	
Calcification	0	0	0	0	0	0	0.557	0.788	34	0.679	0.849	25	0.652	0.802	15	0.583	0.790	457
Int Structure	0.587	0.792	525	0.731	0.854	5	0.630	0.915	1	0	0	0						
Lobulation	0.547	0.781	297	0.646	0.813	173	0.633	0.795	41	0.584	0.798	17	0.75	0.802	3			
Malignancy	0.560	0.770	60	0.557	0.784	176	0.572	0.783	194	0.696	0.838	88	0.651	0.860	13			
Margin	0.518	0.713	14	0.529	0.759	55	0.564	0.772	69	0.622	0.807	234	0.575	0.800	159			
Sphericity	0	0	0	0.568	0.742	42	0.586	0.775	116	0.604	0.808	295	0.540	0.792	78			
Spiculation	0.564	0.786	348	0.625	0.798	132	0.677	0.835	27	0.649	0.819	18	0.593	0.829	6			
Subtlety	0.448	0.727	21	0.509	0.738	72	0.561	0.763	98	0.626	0.814	228	0.613	0.824	112			
Texture	0.457	0.748	22	0.491	0.740	29	0.523	0.755	32	0.620	0.795	85	0.602	0.803	363			

Table 7.4: Dice coefficients for each nodular characteristic, achieved by the SBF and the SegU-Net, evaluated in the same set of 531 nodules.

Scoring	1		#Nods	2		#Nods	3		#Nods	4		#Nods	5		#Nods	6		#Nods
	SBF	**SegU-Net**		**SBF**	**SegU-Net**		**SBF**	**SegU-Net**		**SBF**	**SegU-Net**		**SBF**	**SegU-Net**		**SBF**	**SegU-Net**	
Calcification	0	0	0	0	0	0	0.586	0.769	39	0.624	0.835	18	0.694	0.826	9	0.586	0.784	465
Int Structure	0.588	0.785	528	0.769	0.886	3	0	0	0	0	0	0						
Lobulation	0.538	0.777	289	0.645	0.800	162	0.662	0.788	53	0.689	0.815	24	0.404	0.584	3			
Malignancy	0.547	0.742	58	0.548	0.776	154	0.566	0.779	211	0.721	0.837	94	0.674	0.818	14			
Margin	0.483	0.707	19	0.532	0.728	60	0.594	0.772	70	0.627	0.805	230	0.566	0.795	152			
Sphericity	0	0	0	0.500	0.693	34	0.597	0.762	125	0.603	0.806	296	0.564	0.786	76			
Spiculation	0.556	0.779	340	0.645	0.804	130	0.664	0.784	36	0.692	0.812	18	0.537	0.691	7			
Subtlety	0.434	0.685	22	0.487	0.723	68	0.542	0.766	100	0.636	0.804	221	0.629	0.822	120			
Texture	0.499	0.714	36	0.478	0.765	19	0.472	0.719	24	0.622	0.786	113	0.603	0.799	339			

7.3 Discussion

The conventional approach exhibits a highly satisfactory performance when dealing with well-circumscribed solid nodules, with overall defined sharp margins. In these cases, both smaller and larger nodules tend to be segmented in accordance with the specialists. The algorithm also deals very well with nodules whose intensities vary within their border (i.e. cavitary and calcific nodules), as it is able to ignore the cavities and calcific regions during the segmentation process. Vascularized nodules have the potential to pose a challenge, considering the inherent difficulty in distinguishing the nodule from the attached vessels. However, the SBF based approach is frequently able to separate them and create a mask which does not include the vessels, as shown in Figure 7.5b. Such feat is possible thanks to the truncated mask, which guides the SBF and consequently is able to remove vascular structures from the segmentation. Contrarily, the nodules which have a pleural tail generally have the thin structure ignored by the algorithm, which does not include it in the segmentation mask. In such situations, it is possible to conclude that the result differs from the ground truth, as the specialists consider the pleural tail as part of the nodule.

The main flaws of the SBF algorithm appear when dealing with juxtapleural nodules: since these lesions are attached to the pleural wall and do not exhibit a sharp margin, the algorithm often does not know where the nodule ends and the pleura begins. In some cases, it is able to estimate to some extent where the nodule ends (Figure 7.5a), while in other cases part of the pleural wall is included in the segmentation (Figure 7.7b). The less satisfying results are also due to the unexpected irregular shape of the nodule, as shown in Figure 7.7a, or because the nodule does not have a clear margin (e.g. non-solid nodules/ground glass opacities, in Figure 7.6b).

Similarly to the SBF based approach, the U-Net is able to clearly segment well-circumscribed solid nodules, independently of their size. An example of a segmented solid nodule is illustrated in Figure 7.8a. It also functions correctly with cavitary and calcific nodules, as well as vascularized nodules. The last case is illustrated in Figure 7.9a, where the U-Net is able to exclude the vessels from the segmentation. Besides that, both methodologies disagree with the specialists and do not include pleural tails in their segmentation masks. However, unlike the SBF, the U-Net demonstrates a great skill when segmenting juxtapleural nodules, being able to tell almost perfectly where the nodule ends and the pleura begins (Figure 7.8b).

The U-Net experiences some degree of difficulty when segmenting irregular shapes and non-solid lesions. Figures 7.9 and 7.10 portray how the network seems to be less efficient when trying to establish the boundary of these types of nodules: most of the times, it tends to encompass more pixels in the segmentation than it would be expected (Figures 7.9b and 7.10a), but the opposite may also occur (Figure 7.10b). However, this approach clearly outperforms the first, even in these cases.

Table 7.3 compares the two previous methodologies using the same set of nodules and indicates that in general the U-Net outperforms the SBF. The following analysis focuses on the characteristics which are supported by a significant number of nodules (i.e. characteristics which have at least 15 nodules). It is important to highlight that the U-Net exhibits a distinctive performance for nodules which are moderately suspicious of malignancy, very obvious, solid in texture, and/or have sharper margins. Besides that, the performance is also increased when dealing with non-central and central calcification, reduced lobulation, and a degree of spiculation in between the spectrum. On the contrary and as expected, the lowest performance values (i.e. Dice lower than 75%) happen in nodules with a shape between linear and ovoid (which may be considered an irregular shape, in the sense that it is not close to round), subtle lesions, and GGOs/non-solid nodules.

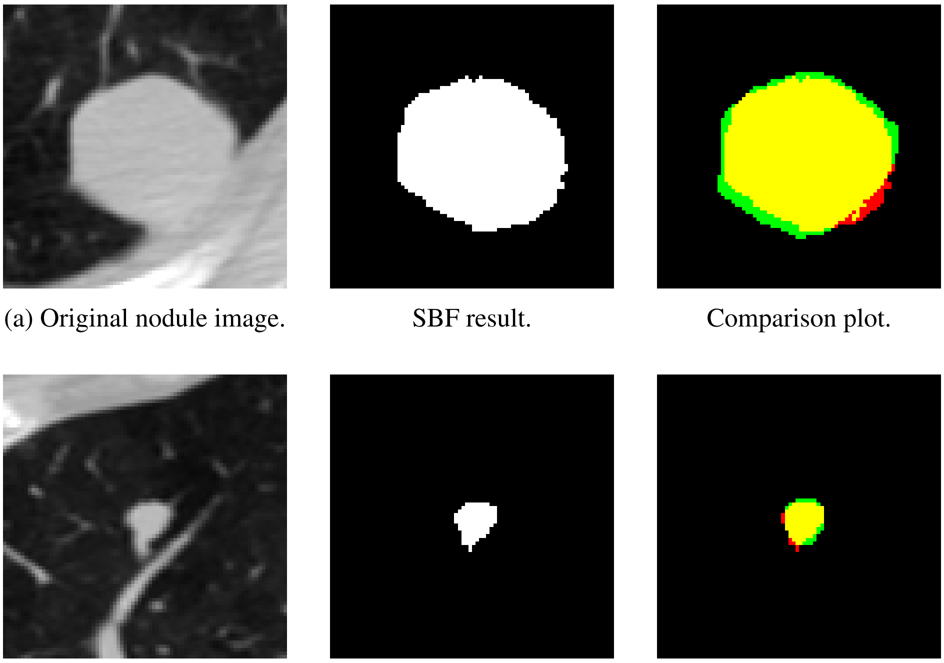
The SBF exhibits maximal performance when handling obvious solid lesions, with sharp margins, and that are moderately suspicious of malignancy. Other characteristics can be brought to one's attention, as this filter also maximized its performance in nodules with non-central and central calcification, low to moderate lobulation, and spiculation in between the spectrum, in accordance with the U-Net. On the other hand, the SBF performed with less efficiency (i.e. Dice lower than 50%) when analyzing extremely subtle lesions, and non-solid nodules/GGOs.

In general, the U-Net segmentation results are very similar to the ones given by the specialists, hence its great performance. In some cases, the U-Net even achieves a more uniform detailed segmentation in comparison to the specialists', which may be justified by its pixel-wise perspicacity.

Finally, the SegU-Net was not able to outperform the U-Net, even though their scores are identical. Figure 7.11 exemplifies how this network is capable of precisely segmenting solid nodules with defined margins, with or without vasculature and other structures near them. The same can be stated when working with cavitary (Figure 7.12a) and calcific nodules. Similarly to the approaches presented above, this one also does not consider pleural tails as part of the nodule, which contradicts the radiologists. The algorithm's slightly inferior performance in comparison to the U-Net is mostly due to an increase of difficulty when segmenting irregular shapes (Figures 7.12b and 7.13a), or non-solid lesions, as shown in Figure 7.13b.

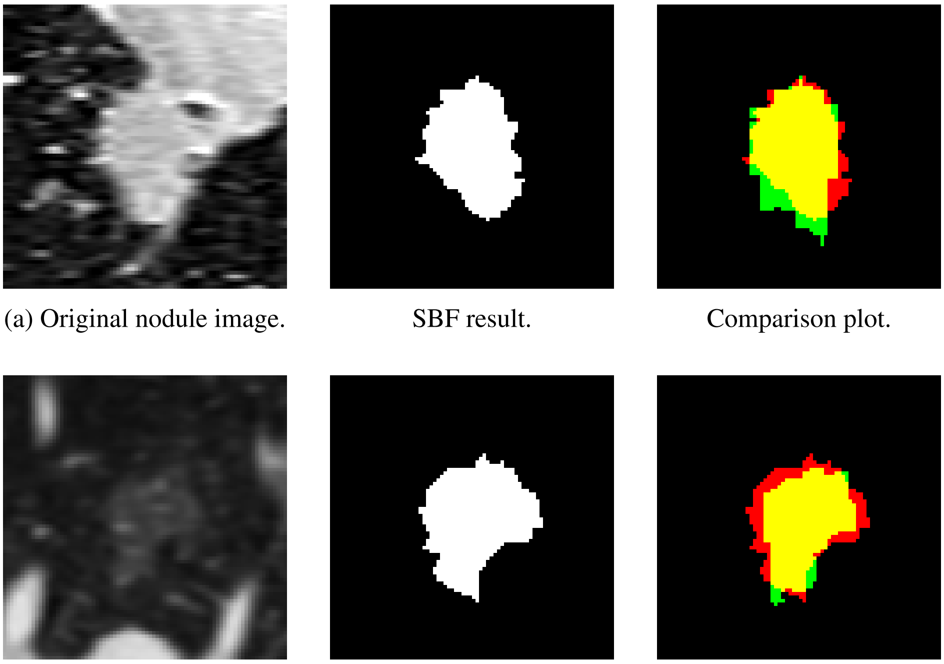
Considering that the 531 nodules in the U-Net's test set are very different from the 531 nodules in the SegU-Net's, the number of common nodules would not be representative of the results and so no direct comparison can be made. For this reason, it was decided not to create a comparison table between both Deep Learning approaches with their performance for each nodular characteristic. However, it is possible to compare the SegU-Net with the SBF (Table 7.4), and use that as an indirect mean of comparison with the U-Net. From this table, and looking at the representative examples (i.e. at least 15 nodules per characteristic), one can infer that the SegU-Net has a great ability to segment obvious nodules with sharper margins, moderate suspicion of malignancy, and/or solid texture. It also works well with all spiculation and lobulation categories. The SegU-Net seems to be less efficient (i.e. Dice lower than 75%) for subtle lesions with poorly defined margins and non-solid or mixed texture. The same happens when dealing with nodules whose shape is in between linear and ovoid (i.e. irregular), and nodules which are unlikely to be malignant.

Taking into consideration the conventional approach, Table 7.4 is in agreement with Table 7.3, proving once again that the SBF has its maximum scores when segmenting mostly obvious nodules with more solid texture, moderate suspicion of malignancy, and/or sharper margins. As mentioned before, the worst performance happens for overall subtle nodules, whose texture is either non-solid or mixed.


(a) Original nodule image. SBF result. Comparison plot.

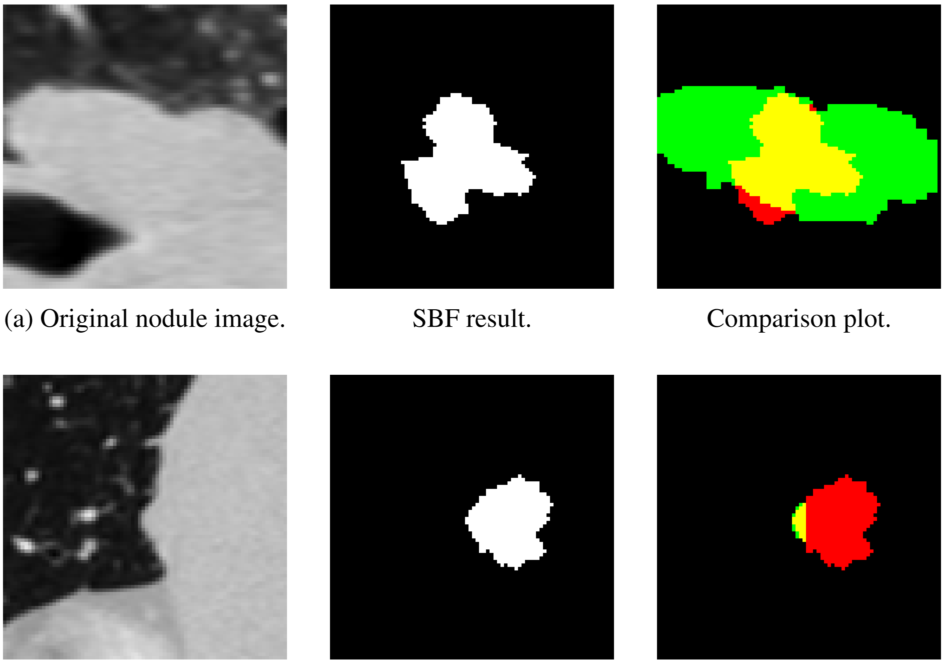
(b) Original nodule image. SBF result. Comparison plot.

Figure 7.5: Example of two nodules in which the SBF performed highly.


(a) Original nodule image. SBF result. Comparison plot.

(b) Original nodule image. SBF result. Comparison plot.

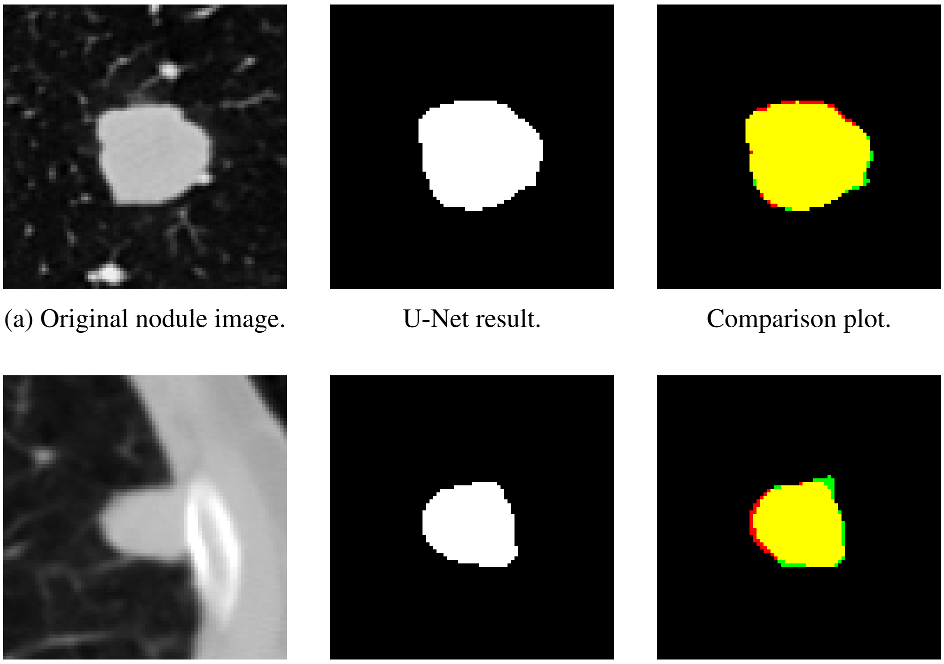
Figure 7.6: Example of two nodules in which the SBF performed less satisfactorily.



(a) Original nodule image. SBF result. Comparison plot.

(b) Original nodule image. SBF result. Comparison plot.

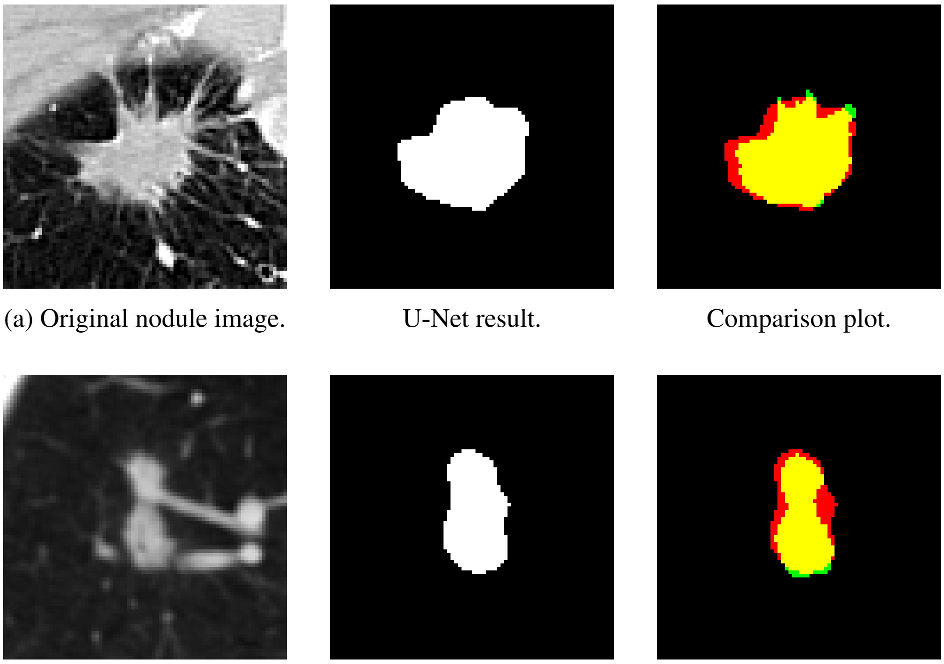
Figure 7.7: Example of two nodules in which the SBF performed poorly.



(a) Original nodule image. U-Net result. Comparison plot.

(b) Original nodule image. U-Net result. Comparison plot.

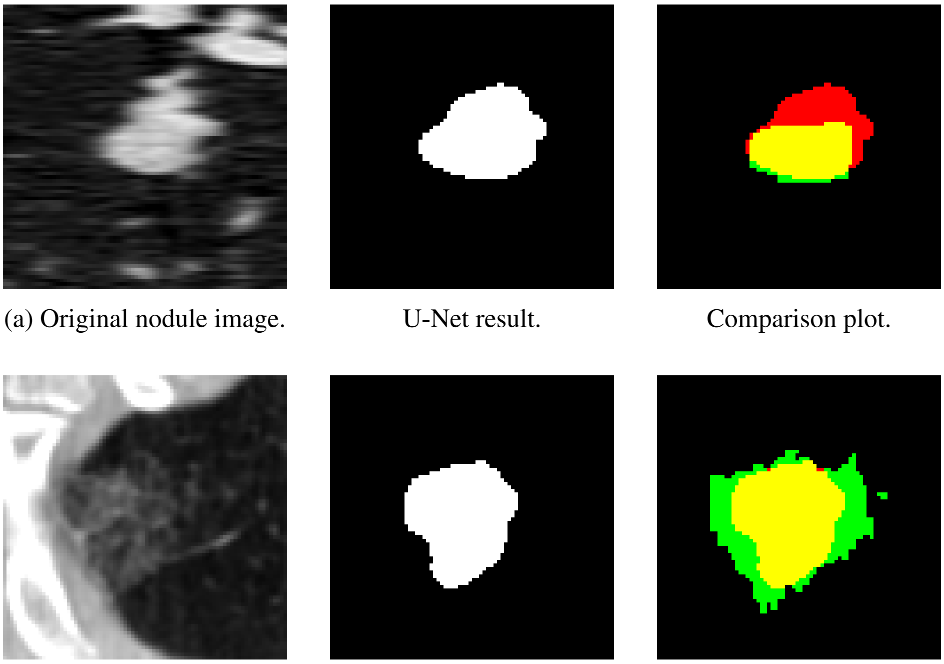
Figure 7.8: Example of two nodules in which the U-Net performed highly.



(a) Original nodule image. U-Net result. Comparison plot.

(b) Original nodule image. U-Net result. Comparison plot.

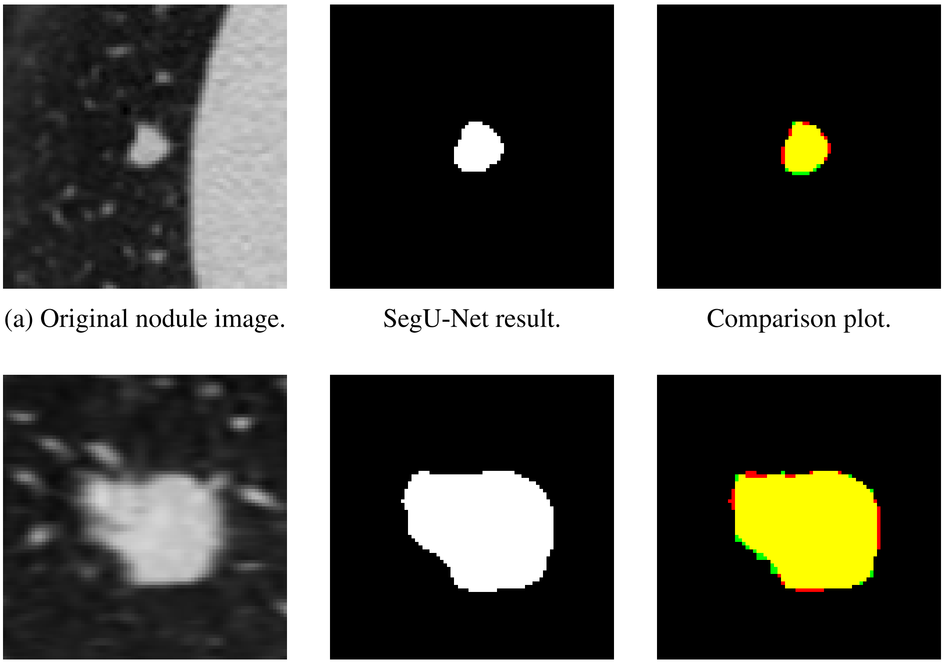
Figure 7.9: Example of two nodules in which the U-Net performed less satisfactorily.



(a) Original nodule image. U-Net result. Comparison plot.

(b) Original nodule image. U-Net result. Comparison plot.

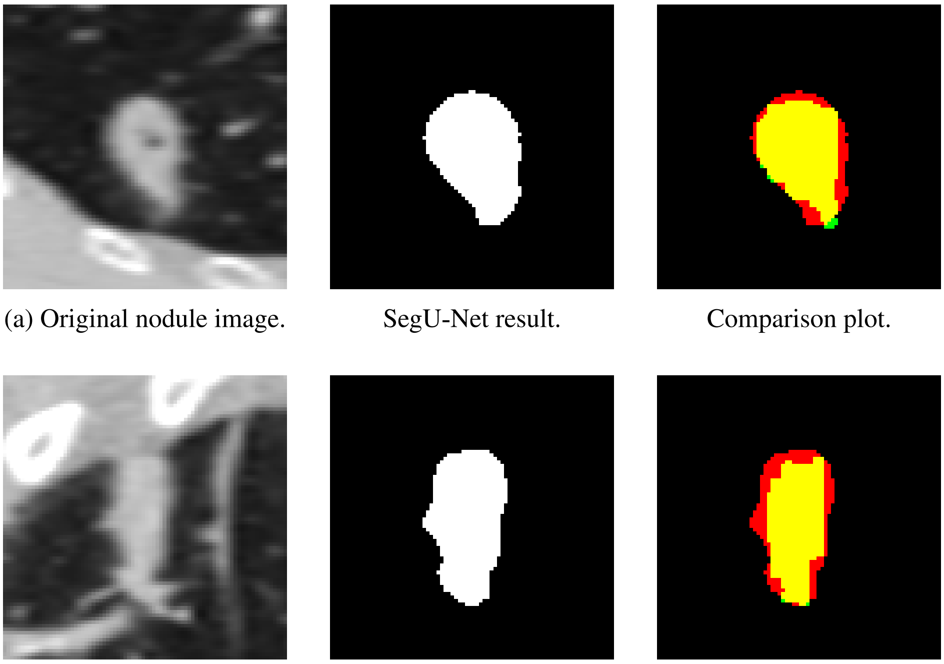
Figure 7.10: Example of two nodules in which the U-Net performed poorly.



(a) Original nodule image. SegU-Net result. Comparison plot.

(b) Original nodule image. SegU-Net result. Comparison plot.

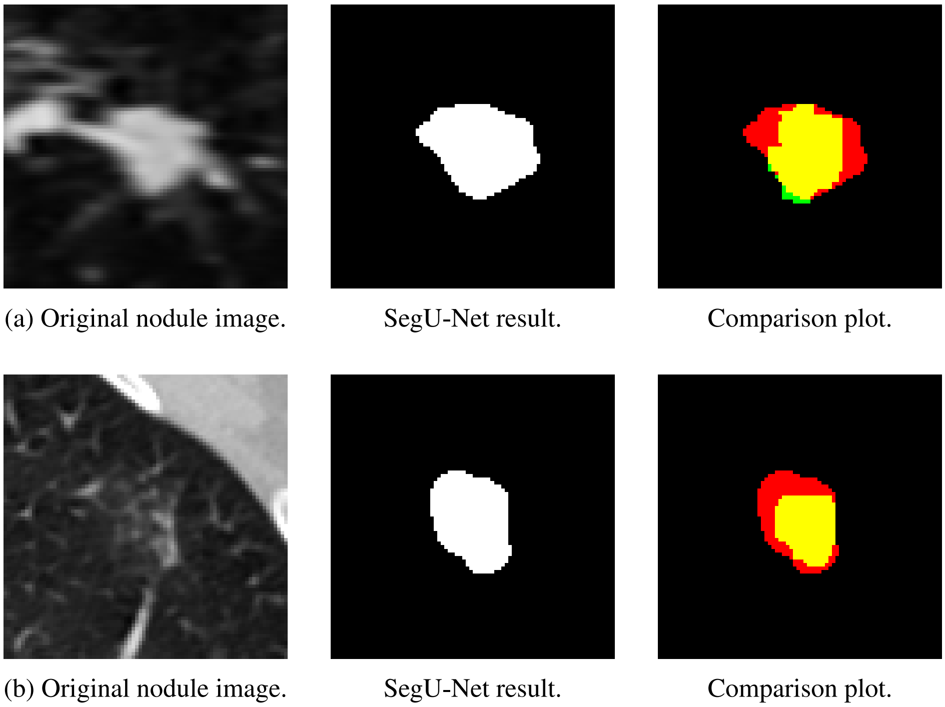
Figure 7.11: Example of two nodules in which the SegU-Net performed highly.



(a) Original nodule image. SegU-Net result. Comparison plot.

(b) Original nodule image. SegU-Net result. Comparison plot.

Figure 7.12: Example of two nodules in which the SegU-Net performed less satisfactorily.



(a) Original nodule image. SegU-Net result. Comparison plot.

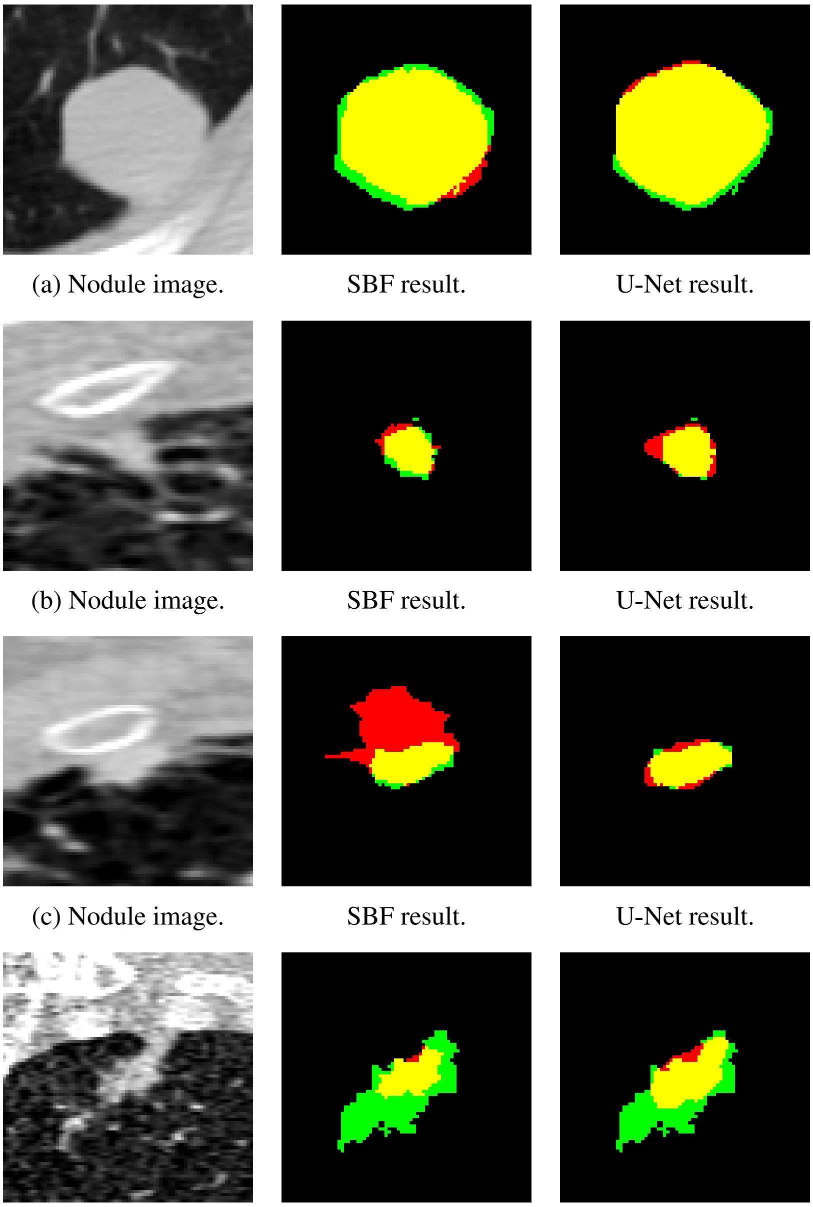
(b) Original nodule image. SegU-Net result. Comparison plot.

Figure 7.13: Example of two nodules in which the SegU-Net performed poorly.

7.3.1 Final Remarks

The conventional approach resulted in the lowest score values in comparison to the Deep Learning based approaches, with a difference of approximately 15% for the Dice Coefficient, considered an average measure of segmentation fitting (coverage) for true positive detections. The same 15% interval is documented for the Jaccard Index, Precision and Sensitivity. The Deep Learning methodologies clearly outperformed the conventional approach, and exhibited similar results among themselves. As explained in Chapter 6, the difference between the two approaches lied in the way each encoder-decoder network upsamples the feature maps. It was expected for the SegU-Net to promote a more precise and detailed segmentation, as ideally it would preserve the spatial knowledge in a more reliable way. However, while using the max pooling indices to perform max unpooling may contribute to the detailed spatial knowledge of the SegU-Net, the remaining activations are zeroed at each unpooling step. On the other hand, the upconvolution operation present in the U-Net can be interpreted as a transposed convolution and does not imply automatically setting any pixels to zero. For this reason, the U-Net still demonstrated to be slightly more efficient than the SegU-Net, and so one can conclude that the SegU-Net's upsampling method does not improve the results for the task at hand, when compared to the original U-Net's upsampling method.

All approaches performed less efficiently for the same nodular types: non-solid lesions and irregularly shaped nodules. The first can be considered a challenge due to their poorly defined margins and high subtlety, but both cases hinder the algorithms from accurately locating the nodule's border. This way, all methods experienced the same challenges, but in different degrees, the SBF being the most impaired method (increase in difficulty led to lower scores). The discrepancy of the score values between the conventional and Deep Learning approaches may also be justified by the fact that the SBF is not able to correctly segment juxtapleural nodules, while the other approaches are. Figures 7.14 and 7.15 display some examples to compare the SBF to the U-Net and the SegU-Net, respectively.



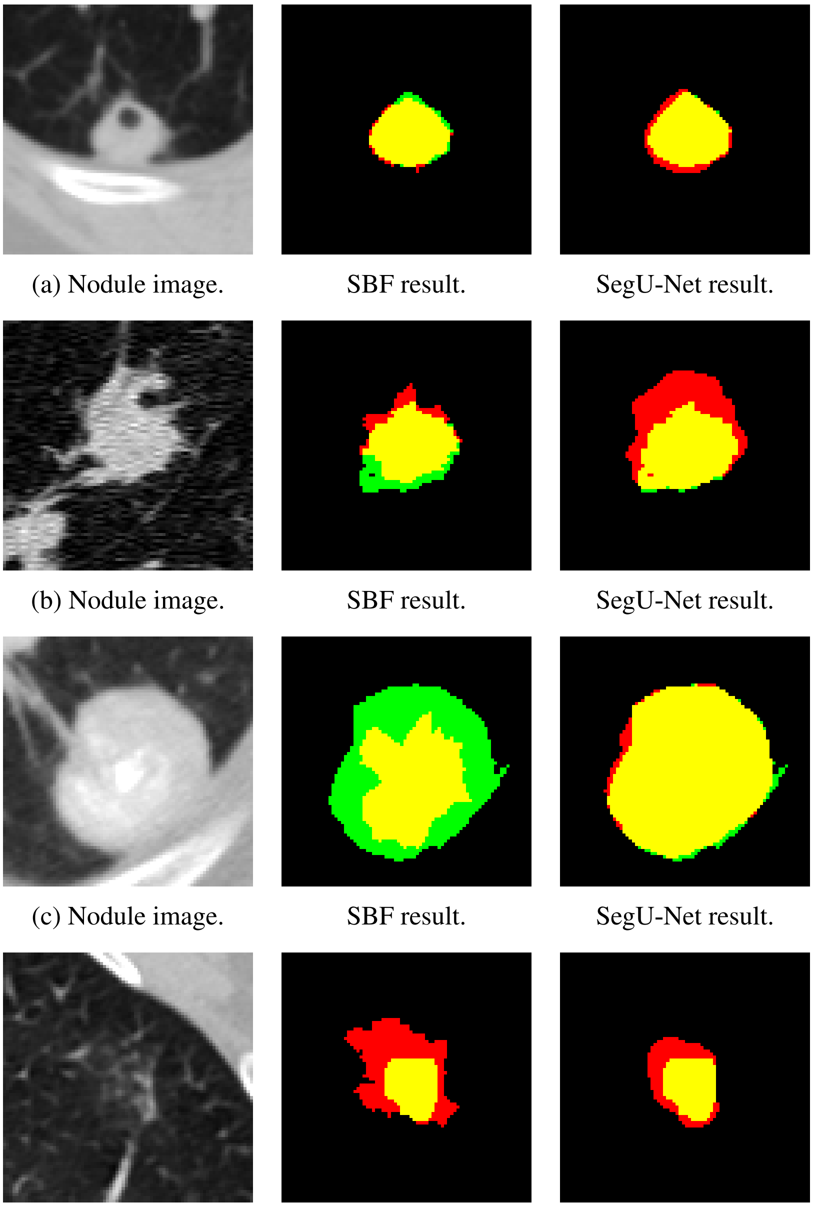
(a) Nodule image. SBF result. U-Net result.

(b) Nodule image. SBF result. U-Net result.

(c) Nodule image. SBF result. U-Net result.

(d) Nodule image. SBF result. U-Net result.

Figure 7.14: Comparison between the SBF and the U-Net: in nodule a) both methodologies are successful, in nodule b) the SBF outperforms the U-Net, in nodule c) the U-Net outperforms the SBF, and finally in nodule d) none of the methodologies have a satisfying result.



(a) Nodule image. SBF result. SegU-Net result.

(b) Nodule image. SBF result. SegU-Net result.

(c) Nodule image. SBF result. SegU-Net result.

(d) Nodule image. SBF result. SegU-Net result.

Figure 7.15: Comparison between the SBF and the SegU-Net: in nodule a) both methodologies are successful, in nodule b) the SBF outperforms the SegU-Net, in nodule c) the SegU-Net outperforms the SBF, and finally in nodule d) none of the methodologies have a satisfying result.

This dissertation includes a comprehensive state of the art revision on pulmonary nodule segmentation; to elaborate on the results achieved by the proposed methodologies, the most recent state of the art algorithms mentioned in Chapter 2 were selected as references. To ensure a fair comparison, certain requirements were taken into consideration when selecting these state of the art methodologies: they must have also evaluated their results on the LIDC database, and used at least one common evaluation metric. Based on these criteria, the selected state of the art methodologies and their main conclusion are briefly described below. Please refer to Chapter 2 for more details on the implementation, and Table 7.5 for the evaluation metrics.

(a) Wang et al. (2017) — Central-focused CNN. This paper presents satisfying results on well-circumscribed lesions, cavitary/calcific, and vascularized nodules, but emphasizes the algorithm's excellent performance on juxtapleural nodules.

(b) Feng et al. (2017) — Weakly-supervised CNNs, whose segmentation is guided by residual nodule activation maps. No conclusions about the performance on specific nodular types are given.

(c) Yang et al. (2018) — Improved fuzzy C-means clustering. This paper reports successful results for well-circumscribed nodules, but less successful ones for GGOs.

(d) Wang et al. (2018) — Region-based CNN, followed by a Deep Active Self-paced learning strategy. No conclusions about the performance on specific nodular types are given.

(e) Aresta et al. (2018) — Adapted U-Net (i.e. 5 contracting steps, a 1×1 bottle neck and a higher number of feature maps on the expansive part). No conclusions about the performance on specific nodular types are given.

(f) Shakibapour et al. (2019) — Unsupervised metaheuristic search. This method's performance drops when dealing with juxtapleural and cavitary nodules.

(g) Kopelowitz and Englehard (2019) — Mask R-CNN. No conclusions about the performance on specific nodular types are given.

Table 7.5: Comparison of the proposed methodologies with state of the art algorithms. C stands for Conventional, and DL stands for Deep Learning.

Methodology	**Type**	**Dice**	**Jaccard**	**Precision**	**Sensitivity**
Wang et al. (2017)	DL	0.823		0.758	0.928
Feng et al. (2017)	DL	0.460			0.770
Yang et al. (2018)	C	0.890		0.820	0.970
Wang et al. (2018)	DL	0.640			
Aresta et al. (2018)	DL	0.790	0.630		
Shakibapour et al. (2019)	C	0.823	0.704	0.856	0.871
Kopelowitz and Englehard (2019)	DL	0.700			
SBF	C	0.663	0.500	0.709	0.732
U-Net	DL	**0.830**	**0.712**	**0.792**	**0.898**
SegU-Net	DL	0.823	0.702	0.787	0.858

This table exhibits the average scores given by the cited articles. If the average values are not available, the algorithms are tested on multiple LIDC subsets, and/or several experiments are made with the same dataset, the results with the highest score values are considered.

As displayed in Table 7.5, and minding the Dice Coefficient as the only metric common to all methodologies, the proposed U-Net model clearly outperforms all state of the art methodologies (with the exception of the conventional approach proposed in Yang et al. (2018)). In general, one can conclude that the U-Net shares common obstacles with most state of the art methodologies, e.g. facing some difficulties segmenting non-solid lesions, but similarly to these approaches, also achieves higher performance scores for obvious lesions.

In future work, the proposed algorithms may be improved by applying a more elaborated pre-processing to the images of the dataset, which can help the algorithms to more easily locate the borders of the nodules. In the conventional approach, a new and more adequate post-processing may also result in a better segmentation. This would be particularly useful to improve the segmentation of juxtapleural nodules, possibly refining the margin which separates the nodule from the pleural wall. To improve the Deep Learning based approaches, perhaps the most straightforward solution is to increase the number of images in the dataset to encompass more non-solid and irregular shaped nodules, thus enabling the models to learn better about these lesions.

7.4 Summary

Several adversities are encountered when evaluating medical segmentation, and for this reason it is very important to establish an adequate standard evaluation system. To do so, it is necessary to select which evaluation metrics to use, as they will be a mean of comparison of the algorithms. This chapter aims to present and discuss the results achieved for the methods previously proposed, in order to identify the most suitable one for the pulmonary nodule segmentation task.

The comparison is done using not only the evaluation metrics presented and defined in this chapter, but also plots which directly assess how close the segmentation is to the ground truth, marking the True Positives, False Positives, and False Negatives. The conventional algorithm was evaluated in the totality of the 2653 nodules in the dataset, while the Deep Learning based approaches used a test set of 531 nodules (20% of the dataset, chosen after studying data representation). Minding the overall evaluation scores (Dice Coefficient, Jaccard Index, Precision, and Sensitivity) and the performance for each nodular characteristic, the following was determined: the U-Net proved to be the most efficient method for pulmonary nodule segmentation - outperforming several state of the art algorithms described in this dissertation -, even though the SegU-Net had identical results. All the approaches perform as expected for well-circumscribed obvious nodules, with sharp margins and/or solid texture, and tend to fail when segmenting non-solid or irregular shaped lesions. The conventional approach exhibited the lowest performance scores, which may be justified by its difficulty segmenting juxtapleural nodules, unlike the other algorithms. As confirmed by their identical scores, the SegU-Net's upsampling method does not improve the results for the task at hand, when compared to the original U-Net's upsampling method. Several suggestions for future work are also presented in this chapter, including more suitable pre-processing and post-processing steps, and an increase of non-solid and irregular shaped nodules in the dataset.

Chapter 8

Conclusions

Lung cancer is one of the current deadliest cancers worldwide, and for that reason frequently assessing the patients' pulmonary health is essential to prevent and monitor any lesions. CAD systems may contribute to the location and characterization of pulmonary nodules through segmentation, thus helping the specialists in this highly relevant clinical task which faces particular challenges. On one hand, medical images are often noisy, blurred and/or have low contrast; on the other hand, there is an inherent complexity associated with the anatomical structures of the lung. Such complexity is often translated by the variability in the size, shape, and texture of each nodule.

This work seeks to segment pulmonary nodules, and consequently contribute to an earlier and more accurate diagnosis of lung cancer. To fulfill this task, both conventional and Deep Learning based techniques can be employed and compared. While conventional techniques are based on knowledge representation, which deduces new information using logical rules and a set of beliefs, Deep Learning approaches carry out data-driven modeling, in the sense that they observe a large amount of data to extract knowledge from it. By comparing the performance of different algorithms, it is possible to evaluate their individual efficiency and in the future perhaps incorporate the most competent one in a CAD system, to assist the specialists in the interpretation of medical images.

The focus of the conventional approach is an LCF, chosen by its ability to deal with noisy, blurred and/or low-contrast images. In specific, the SBF was applied due to its shape flexibility and capacity to ignore the gradients' behavior at the center of the nodule, as its support region is a band of fixed width whose position is adjusted in each radial direction, to maximize the convergence degree and therefore the filter's response. The algorithm applied in this work uses the pre-processed nodule images to first estimate the center of the nodules, get the corresponding support points, and finally refine the segmentation by post-processing the result.

In opposition, the two Deep Learning based methodologies presented in this document are FCNs with encoder-decoder architectures, which interpret the task at hand as a pixel-wise semantic segmentation task, where each pixel is either nodular or non-nodular (binary classification). The U-Net is a network developed for biomedical segmentation, being capable of handling the large size of medical images, and the lack of labeled training data by generating more images through data augmentation, while at the same time simulating elastic deformation in tissue. This network is mostly comprised of convolutional layers, ReLU, batch normalization, and dropout. Its encoder path downsamples the feature maps using max pooling, while the decoder path upsamples them using upconvolutions and concatenations. The perspicacity of the U-Net is mainly due to its skip connections, as they allow the output of one layer to be the input of a non-consecutive layer, and consequently enable concatenations, where the coarse contextual information of the encoder is combined with the upsampled feature maps in the decoder. For this reason, the U-Net obtains extensive detailed information by combining context with spatial knowledge.

The SegU-Net is a hybrid network, taking advantage of the U-Net's proficiency and part of the SegNet's upsampling method; in other words, the SegU-Net keeps the skip connections and concatenations of the U-Net, and replaces its upconvolutions by max unpooling. By doing so, it was expected to improve the U-Net's knowledge on location, and therefore also improve boundary delineation.

The presented techniques were applied to thoracic CT scans from the publicly available LIDC database, and evaluated minding the corresponding segmented lesions as ground truth. The analysis of the methods' performance was achieved through comparison plots to help visualize the results, and through an adequate set of evaluation metrics which take into account the several challenges inherent to medical segmentation (e.g. class unbalance). These metrics serve as a means of comparison between the three approaches. The SBF based approach was tested on a set of 2653 nodules, while the U-Net and the SegU-Net were tested on 531 nodules (20% of the data).

Based on the overall evaluation scores (Dice Coefficient, Jaccard Index, Precision, and Sensitivity) and the performance for each nodular characteristic, one can conclude that the U-Net proved to be the most efficient method for pulmonary nodule segmentation (Dice Coefficient of 0.830). All the approaches succeed when segmenting well-circumscribed obvious nodules, with sharp margins and/or solid texture, and tend to fail with non-solid or irregular shaped lesions. However, they revealed to have different levels of difficulty when dealing with these nodular types, and so each of their scores are inversely proportional to the level of difficulty - in other words, the higher the difficulty when segmenting certain nodules, the lower the score values. The conventional approach exhibited the lowest performance scores (Dice Coefficient of 0.663), which may be justified by its overall poor segmentation of juxtapleural nodules, correctly segmented by the Deep Learning algorithms. Considering that the SegU-Net achieved identical score values to the U-Net (Dice Coefficient of 0.823), it is possible to infer that the SegU-Net's upsampling method does not improve the results for the task at hand, when compared to the original U-Net's upsampling method.

Additional efforts can be done in future work to improve the algorithms' performance, namely develop a more adequate pre-processing for the input images, in order to promote the capability of the segmentation process. In the conventional approach, the border coordinates may be refined for the juxtapleural nodules by establishing a more efficient post-processing stage. In the Deep Learning approaches, the most straightforward way to enhance their performance in non-solid or irregular shaped lesions would be to add more of these examples to the training set, promoting a more advanced and perceptive learning.

Ultimately, the comparison of the conventional and the Deep Learning based approaches explored the advantages and disadvantages of each one in this segmentation task, establishing the U-Net as the most efficient method in this case. The model is particularly efficient for obvious lesions, being able to outperform several state of the art algorithms described in this dissertation, and can be successfully employed in other studies to segment objects with defined margins. Furthermore, this work contributes to a possible implementation of this model in a decision support system, thus assisting the physicians to establish a reliable and trustworthy diagnosis of lung pathologies, based on this segmentation task.

Publications

Two publications resulted from the present dissertation, enumerated below. Both papers were accepted in two distinct conferences.

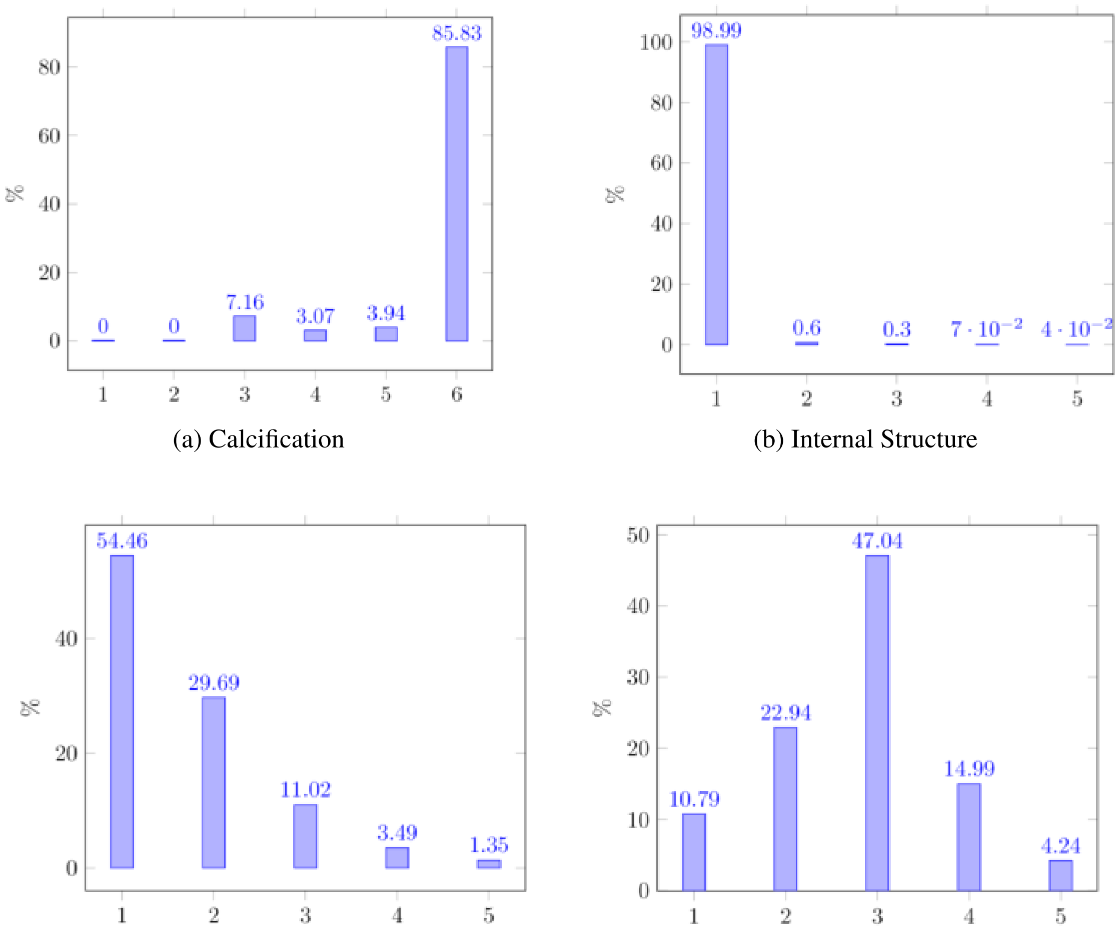
Joana Rocha, António Cunha, Ana Maria Mendonça. *Comparison of Conventional and Deep Learning based Methods for Pulmonary Nodule Segmentation in CT Images*. EPIA, Vila Real, Portugal, 3rd-6th September 2019.

Joana Rocha, António Cunha, Ana Maria Mendonça. *Segmentation of Pulmonary Nodules in CT Images using the Sliding Band Filter*. Medicon, Coimbra, Portugal, 26th-28th September 2019.

Appendix A

Additional Information

LIDC Database



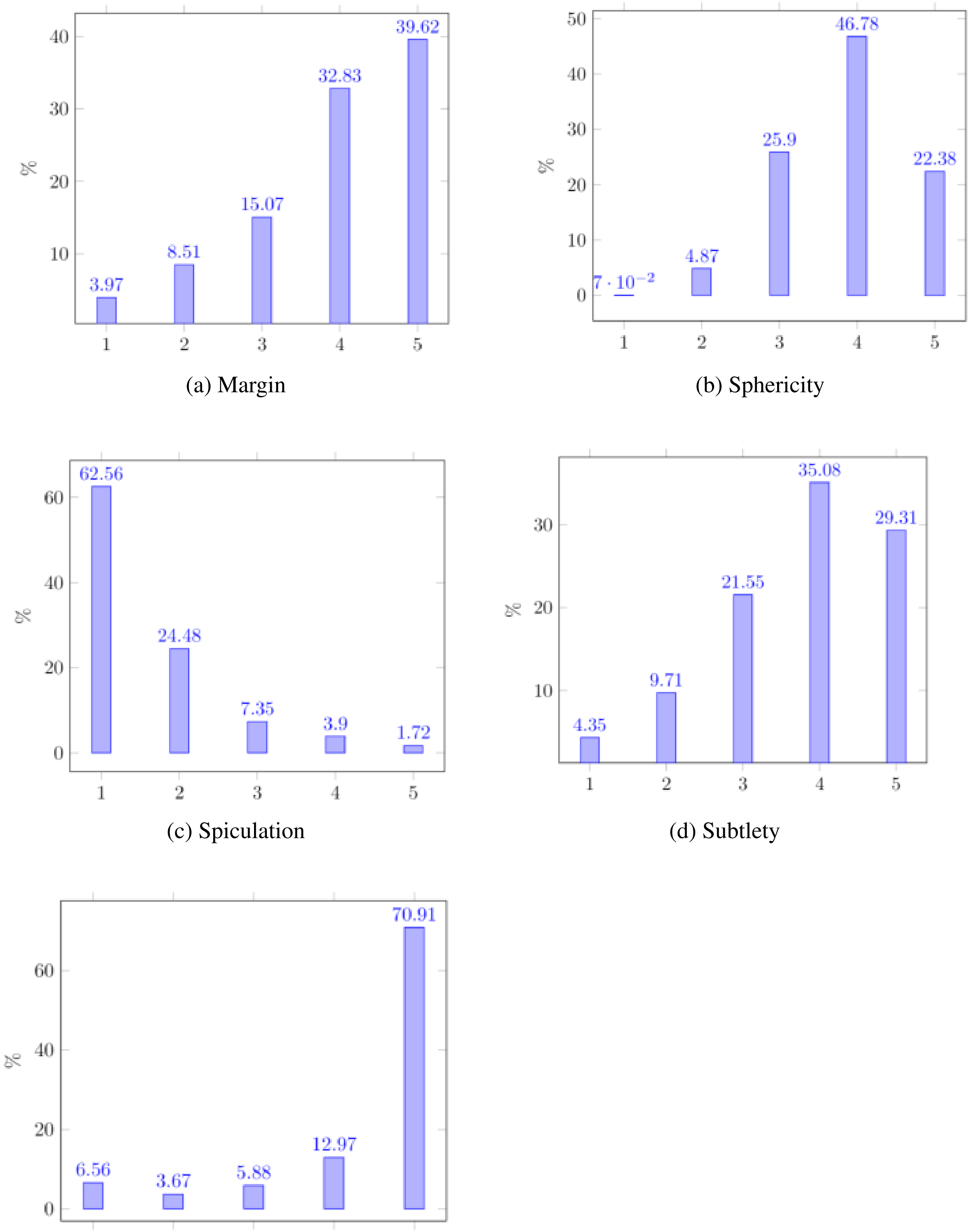
(a) Calcification

(b) Internal Structure

(c) Lobulation

(d) Malignancy

Figure A.1: Percentage of nodules that are labeled with 1, 2, 3, 4, 5 or 6, for each characteristic assessed.



(a) Margin

(b) Sphericity

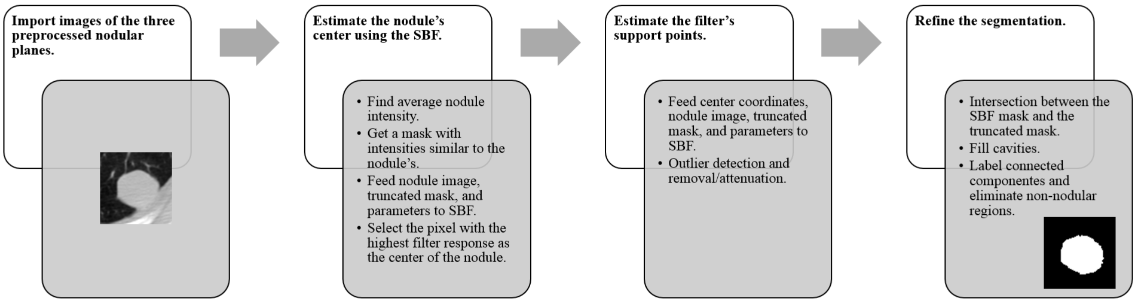
(c) Spiculation

(d) Subtlety

(e) Texture

Figure A.2: Percentage of nodules that are labeled with 1, 2, 3, 4, 5 or 6, for each characteristic assessed (continuation).

Conventional Approach



Import images of the three preprocessed nodular planes.

Estimate the nodule's center using the SBF.

- Find average nodule intensity.
- Get a mask with intensities similar to the nodule's.
- Feed nodule image, truncated mask, and parameters to SBF.
- Select the pixel with the highest filter response as the center of the nodule.

Estimate the filter's support points.

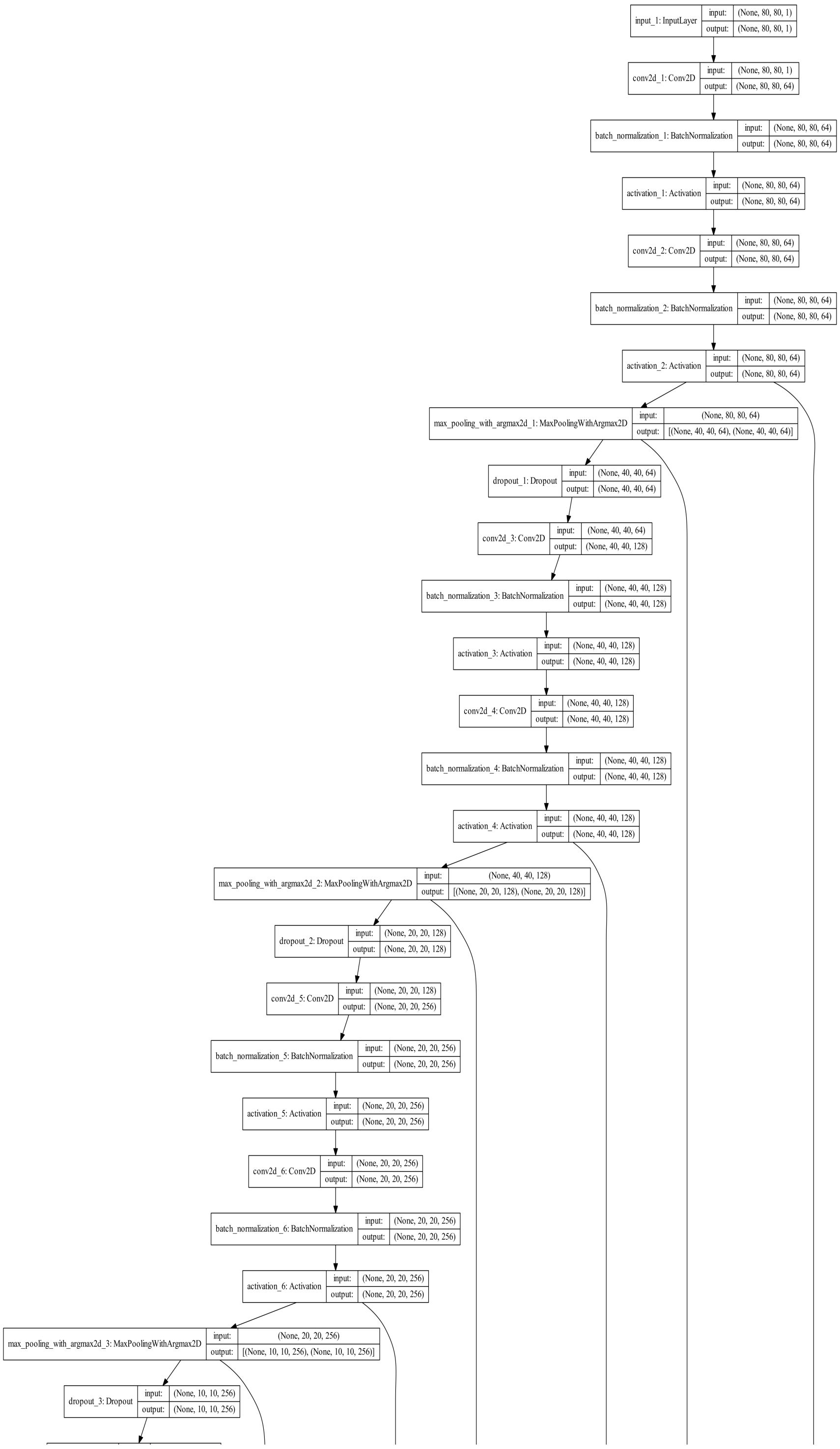
- Feed center coordinates, nodule image, truncated mask, and parameters to SBF.
- Outlier detection and removal/attenuation.

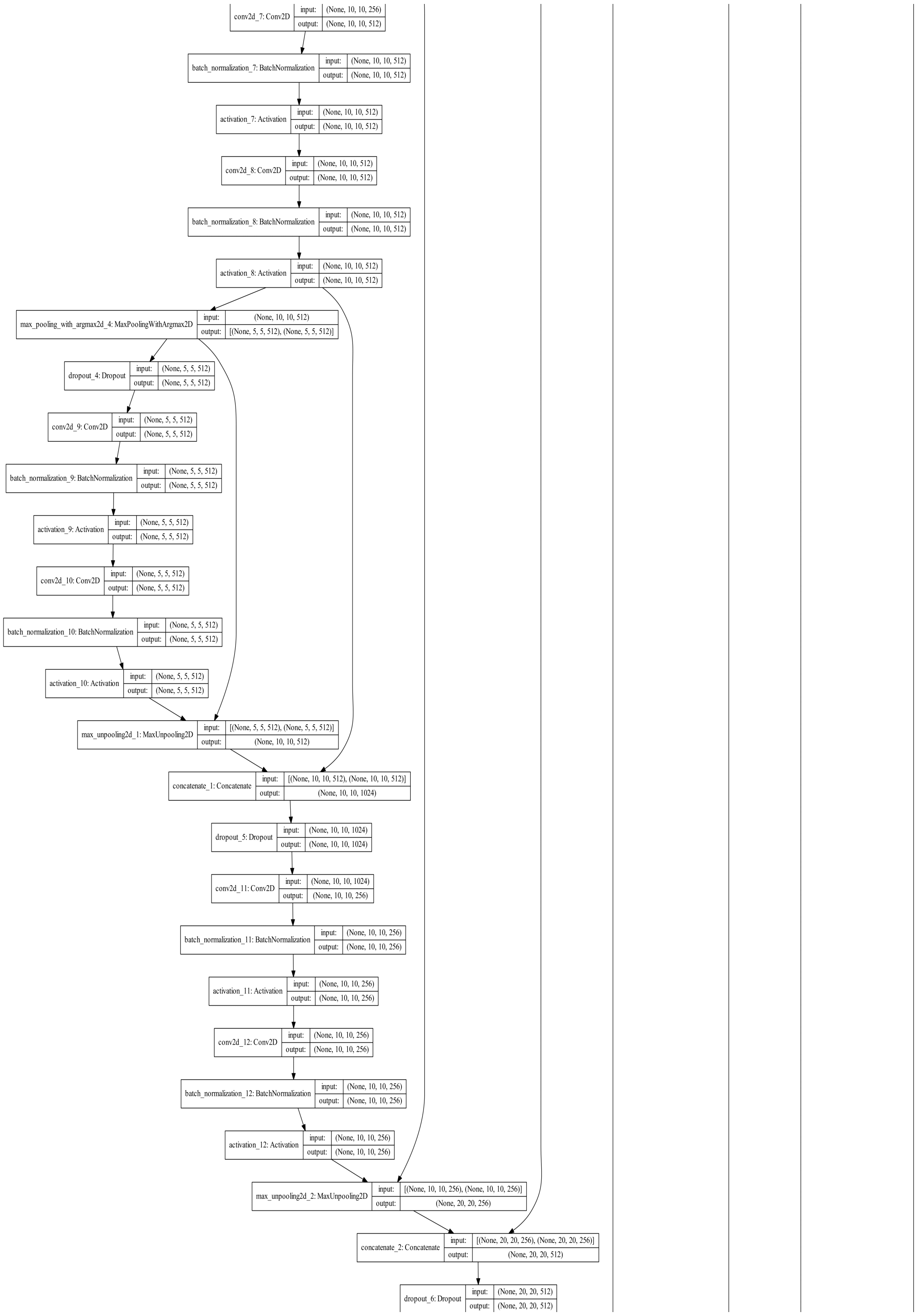
Refine the segmentation.

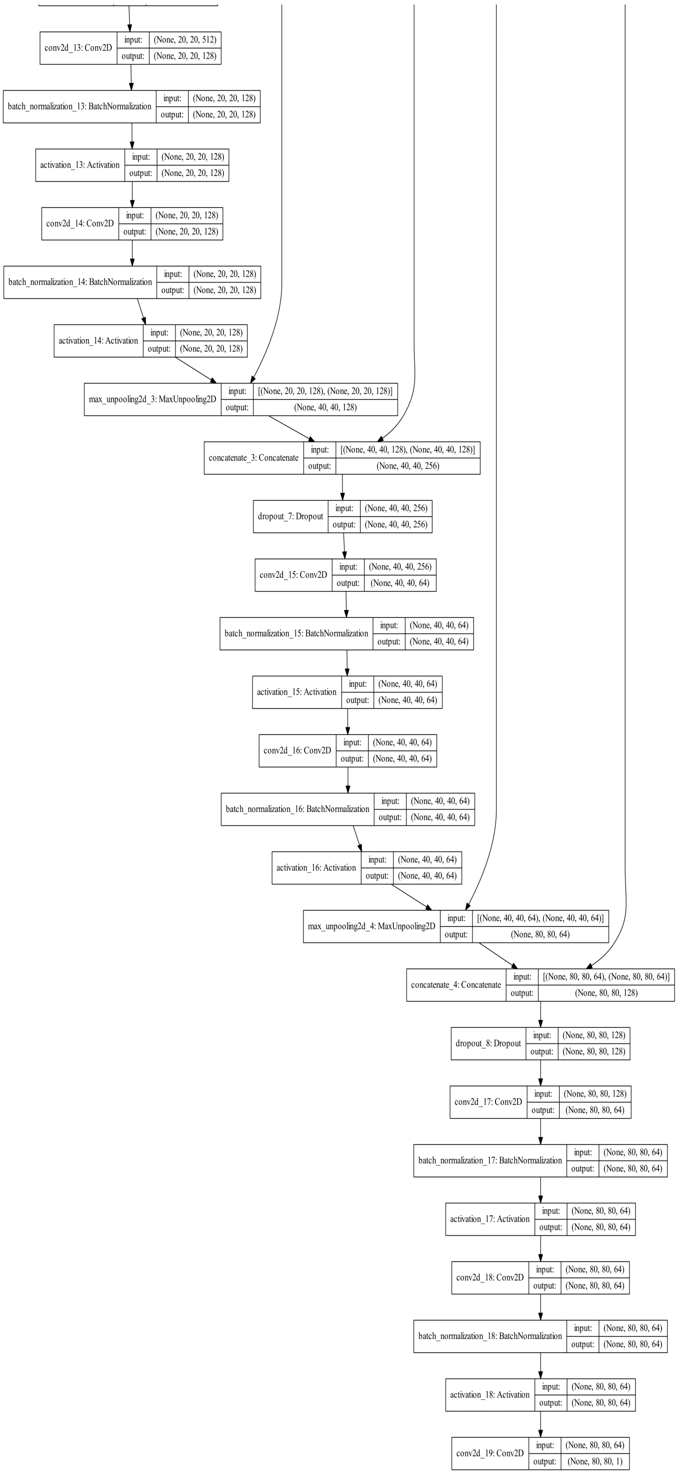
- Intersection between the SBF mask and the truncated mask.
- Fill cavities.
- Label connected componentes and eliminate non-nodular regions.

Figure A.3: Flowchart of the conventional algorithm.

SegU-Net's model


input_1: InputLayer
input: (None, 80, 80, 1)
output: (None, 80, 80, 1)
conv2d_1: Conv2D
input: (None, 80, 80, 1)
output: (None, 80, 80, 64)
batch_normalization_1: BatchNormalization
input: (None, 80, 80, 64)
output: (None, 80, 80, 64)
activation_1: Activation
input: (None, 80, 80, 64)
output: (None, 80, 80, 64)
conv2d_2: Conv2D
input: (None, 80, 80, 64)
output: (None, 80, 80, 64)
batch_normalization_2: BatchNormalization
input: (None, 80, 80, 64)
output: (None, 80, 80, 64)
activation_2: Activation
input: (None, 80, 80, 64)
output: (None, 80, 80, 64)
max_pooling_with_argmax2d_1: MaxPoolingWithArgmax2D
input: (None, 80, 80, 64)
output: [(None, 40, 40, 64), (None, 40, 40, 64)]
dropout_1: Dropout
input: (None, 40, 40, 64)
output: (None, 40, 40, 64)
conv2d_3: Conv2D
input: (None, 40, 40, 64)
output: (None, 40, 40, 128)
batch_normalization_3: BatchNormalization
input: (None, 40, 40, 128)
output: (None, 40, 40, 128)
activation_3: Activation
input: (None, 40, 40, 128)
output: (None, 40, 40, 128)
conv2d_4: Conv2D
input: (None, 40, 40, 128)
output: (None, 40, 40, 128)
batch_normalization_4: BatchNormalization
input: (None, 40, 40, 128)
output: (None, 40, 40, 128)
activation_4: Activation
input: (None, 40, 40, 128)
output: (None, 40, 40, 128)
max_pooling_with_argmax2d_2: MaxPoolingWithArgmax2D
input: (None, 40, 40, 128)
output: [(None, 20, 20, 128), (None, 20, 20, 128)]
dropout_2: Dropout
input: (None, 20, 20, 128)
output: (None, 20, 20, 128)
conv2d_5: Conv2D
input: (None, 20, 20, 128)
output: (None, 20, 20, 256)
batch_normalization_5: BatchNormalization
input: (None, 20, 20, 256)
output: (None, 20, 20, 256)
activation_5: Activation
input: (None, 20, 20, 256)
output: (None, 20, 20, 256)
conv2d_6: Conv2D
input: (None, 20, 20, 256)
output: (None, 20, 20, 256)
batch_normalization_6: BatchNormalization
input: (None, 20, 20, 256)
output: (None, 20, 20, 256)
activation_6: Activation
input: (None, 20, 20, 256)
output: (None, 20, 20, 256)
max_pooling_with_argmax2d_3: MaxPoolingWithArgmax2D
input: (None, 20, 20, 256)
output: [(None, 10, 10, 256), (None, 10, 10, 256)]
dropout_3: Dropout
input: (None, 10, 10, 256)
output: (None, 10, 10, 256)


conv2d_7: Conv2D | input: (None, 10, 10, 256) | output: (None, 10, 10, 512)
batch_normalization_7: BatchNormalization | input: (None, 10, 10, 512) | output: (None, 10, 10, 512)
activation_7: Activation | input: (None, 10, 10, 512) | output: (None, 10, 10, 512)
conv2d_8: Conv2D | input: (None, 10, 10, 512) | output: (None, 10, 10, 512)
batch_normalization_8: BatchNormalization | input: (None, 10, 10, 512) | output: (None, 10, 10, 512)
activation_8: Activation | input: (None, 10, 10, 512) | output: (None, 10, 10, 512)
max_pooling_with_argmax2d_4: MaxPoolingWithArgmax2D | input: (None, 10, 10, 512) | output: [(None, 5, 5, 512), (None, 5, 5, 512)]
dropout_4: Dropout | input: (None, 5, 5, 512) | output: (None, 5, 5, 512)
conv2d_9: Conv2D | input: (None, 5, 5, 512) | output: (None, 5, 5, 512)
batch_normalization_9: BatchNormalization | input: (None, 5, 5, 512) | output: (None, 5, 5, 512)
activation_9: Activation | input: (None, 5, 5, 512) | output: (None, 5, 5, 512)
conv2d_10: Conv2D | input: (None, 5, 5, 512) | output: (None, 5, 5, 512)
batch_normalization_10: BatchNormalization | input: (None, 5, 5, 512) | output: (None, 5, 5, 512)
activation_10: Activation | input: (None, 5, 5, 512) | output: (None, 5, 5, 512)
max_unpooling2d_1: MaxUnpooling2D | input: [(None, 5, 5, 512), (None, 5, 5, 512)] | output: (None, 10, 10, 512)
concatenate_1: Concatenate | input: [(None, 10, 10, 512), (None, 10, 10, 512)] | output: (None, 10, 10, 1024)
dropout_5: Dropout | input: (None, 10, 10, 1024) | output: (None, 10, 10, 1024)
conv2d_11: Conv2D | input: (None, 10, 10, 1024) | output: (None, 10, 10, 256)
batch_normalization_11: BatchNormalization | input: (None, 10, 10, 256) | output: (None, 10, 10, 256)
activation_11: Activation | input: (None, 10, 10, 256) | output: (None, 10, 10, 256)
conv2d_12: Conv2D | input: (None, 10, 10, 256) | output: (None, 10, 10, 256)
batch_normalization_12: BatchNormalization | input: (None, 10, 10, 256) | output: (None, 10, 10, 256)
activation_12: Activation | input: (None, 10, 10, 256) | output: (None, 10, 10, 256)
max_unpooling2d_2: MaxUnpooling2D | input: [(None, 10, 10, 256), (None, 10, 10, 256)] | output: (None, 20, 20, 256)
concatenate_2: Concatenate | input: [(None, 20, 20, 256), (None, 20, 20, 256)] | output: (None, 20, 20, 512)
dropout_6: Dropout | input: (None, 20, 20, 512) | output: (None, 20, 20, 512)


conv2d_13: Conv2D | input: (None, 20, 20, 512) | output: (None, 20, 20, 128)
batch_normalization_13: BatchNormalization | input: (None, 20, 20, 128) | output: (None, 20, 20, 128)
activation_13: Activation | input: (None, 20, 20, 128) | output: (None, 20, 20, 128)
conv2d_14: Conv2D | input: (None, 20, 20, 128) | output: (None, 20, 20, 128)
batch_normalization_14: BatchNormalization | input: (None, 20, 20, 128) | output: (None, 20, 20, 128)
activation_14: Activation | input: (None, 20, 20, 128) | output: (None, 20, 20, 128)
max_unpooling2d_3: MaxUnpooling2D | input: [(None, 20, 20, 128), (None, 20, 20, 128)] | output: (None, 40, 40, 128)
concatenate_3: Concatenate | input: [(None, 40, 40, 128), (None, 40, 40, 128)] | output: (None, 40, 40, 256)
dropout_7: Dropout | input: (None, 40, 40, 256) | output: (None, 40, 40, 256)
conv2d_15: Conv2D | input: (None, 40, 40, 256) | output: (None, 40, 40, 64)
batch_normalization_15: BatchNormalization | input: (None, 40, 40, 64) | output: (None, 40, 40, 64)
activation_15: Activation | input: (None, 40, 40, 64) | output: (None, 40, 40, 64)
conv2d_16: Conv2D | input: (None, 40, 40, 64) | output: (None, 40, 40, 64)
batch_normalization_16: BatchNormalization | input: (None, 40, 40, 64) | output: (None, 40, 40, 64)
activation_16: Activation | input: (None, 40, 40, 64) | output: (None, 40, 40, 64)
max_unpooling2d_4: MaxUnpooling2D | input: [(None, 40, 40, 64), (None, 40, 40, 64)] | output: (None, 80, 80, 64)
concatenate_4: Concatenate | input: [(None, 80, 80, 64), (None, 80, 80, 64)] | output: (None, 80, 80, 128)
dropout_8: Dropout | input: (None, 80, 80, 128) | output: (None, 80, 80, 128)
conv2d_17: Conv2D | input: (None, 80, 80, 128) | output: (None, 80, 80, 64)
batch_normalization_17: BatchNormalization | input: (None, 80, 80, 64) | output: (None, 80, 80, 64)
activation_17: Activation | input: (None, 80, 80, 64) | output: (None, 80, 80, 64)
conv2d_18: Conv2D | input: (None, 80, 80, 64) | output: (None, 80, 80, 64)
batch_normalization_18: BatchNormalization | input: (None, 80, 80, 64) | output: (None, 80, 80, 64)
activation_18: Activation | input: (None, 80, 80, 64) | output: (None, 80, 80, 64)
conv2d_19: Conv2D | input: (None, 80, 80, 64) | output: (None, 80, 80, 1)

References

S. Alqazzaz, X. Sun, X. Yang, and L. Nokes. Automated brain tumor segmentation on multi-modal MR image using SegNet. *Computational Visual Media*, Apr. 2019. ISSN 2096-0662. doi: 10.1007/s41095-019-0139-y. URL https://doi.org/10.1007/s41095-019-0139-y.

R. Anirudh, J. J. Thiagarajan, T. Bremer, and H. Kim. Lung nodule detection using 3d convolutional neural networks trained on weakly labeled data. page 978532, San Diego, California, United States, Mar. 2016. doi: 10.1117/12.2214876. URL http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2214876.

G. Aresta, T. Araújo, C. Jacobs, B. van Ginneken, A. Cunha, I. Ramos, and A. Campilho. Towards an Automatic Lung Cancer Screening System in Low Dose Computed Tomography. In D. Stoyanov, Z. Taylor, B. Kainz, G. Maicas, R. R. Beichel, A. Martel, L. Maier-Hein, K. Bhatia, T. Vercauteren, O. Oktay, G. Carneiro, A. P. Bradley, J. Nascimento, H. Min, M. S. Brown, C. Jacobs, B. Lassen-Schmidt, K. Mori, J. Petersen, R. San José Estépar, A. Schmidt-Richberg, and C. Veiga, editors, *Image Analysis for Moving Organ, Breast, and Thoracic Images*, volume 11040, pages 310–318. Springer International Publishing, Cham, 2018. ISBN 978-3-030-00945-8 978-3-030-00946-5. doi: 10.1007/978-3-030-00946-5_31. URL http://link.springer.com/10.1007/978-3-030-00946-5_31.

S. G. Armato, G. McLennan, L. Bidaut, M. F. McNitt-Gray, C. R. Meyer, A. P. Reeves, B. Zhao, D. R. Aberle, C. I. Henschke, E. A. Hoffman, E. A. Kazerooni, H. MacMahon, E. J. R. van Beek, D. Yankelevitz, A. M. Biancardi, P. H. Bland, M. S. Brown, R. M. Engelmann, G. E. Laderach, D. Max, R. C. Pais, D. P.-Y. Qing, R. Y. Roberts, A. R. Smith, A. Starkey, P. Batra, P. Caligiuri, A. Farooqi, G. W. Gladish, C. M. Jude, R. F. Munden, I. Petkovska, L. E. Quint, L. H. Schwartz, B. Sundaram, L. E. Dodd, C. Fenimore, D. Gur, N. Petrick, J. Freymann, J. Kirby, B. Hughes, A. Vande Casteele, S. Gupte, M. Sallam, M. D. Heath, M. H. Kuhn, E. Dharaiya, R. Burns, D. S. Fryd, M. Salganicoff, V. Anand, U. Shreter, S. Vastagh, B. Y. Croft, and L. P. Clarke. The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): A Completed Reference Database of Lung Nodules on CT Scans: The LIDC/IDRI thoracic CT database of lung nodules. *Medical Physics*, 38(2):915–931, Jan. 2011. ISSN 00942405. doi: 10.1118/1.3528204. URL http://doi.wiley.com/10.1118/1.3528204.

V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *arXiv:1511.00561 [cs]*, Nov. 2015. URL http://arxiv.org/abs/1511.00561. arXiv: 1511.00561.

D. R. Baldwin and M. E. Callister. The British Thoracic Society guidelines on the investigation and management of pulmonary nodules. *Thorax*, 70(8):794–798, Aug. 2015. ISSN 0040-6376, 1468-3296. doi: 10.1136/thoraxjnl-2015-207221. URL http://thorax.bmj.com/lookup/doi/10.1136/thoraxjnl-2015-207221.

B. J. Bartholmai, C. W. Koo, G. B. Johnson, D. B. White, S. M. Raghunath, S. Rajagopalan, M. R. Moynagh, R. M. Lindell, and T. E. Hartman. Pulmonary Nodule Characterization, Including Computer Analysis and Quantitative Features:. *Journal of Thoracic Imaging*, 30(2):139–156, Mar. 2015. ISSN 0883-5993. doi: 10.1097/RTI.0000000000000137. URL http://content.wkhealth.com/linkback/openurl?sid=WKPTLP:landingpage&an=00005382-201503000-00008.

M. Berger. *Geometry I.* Universitext. Springer-Verlag, Berlin ; New York, corr. 2nd print edition, 1994. ISBN 978-3-540-11658-5 978-0-387-11658-7 978-3-540-17015-0.

S. A. Bolliger, L. Oesterhelweg, D. Spendlove, S. Ross, and M. J. Thali. Is Differentiation of Frequently Encountered Foreign Bodies in Corpses Possible by Hounsfield Density Measurement? *Journal of Forensic Sciences*, 54(5):1119–1122, Sept. 2009. ISSN 00221198, 15564029. doi: 10.1111/j.1556-4029.2009.01100.x. URL http://doi.wiley.com/10.1111/j.1556-4029.2009.01100.x.

E. Castro, J. S. Cardoso, and J. C. Pereira. Elastic deformations for data augmentation in breast cancer mass detection. In *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pages 230–234, Las Vegas, NV, USA, Mar. 2018. IEEE. ISBN 978-1-5386-2405-0. doi: 10.1109/BHI.2018.8333411. URL http://ieeexplore.ieee.org/document/8333411/.

J.-Z. Cheng, D. Ni, Y.-H. Chou, J. Qin, C.-M. Tiu, Y.-C. Chang, C.-S. Huang, D. Shen, and C.-M. Chen. Computer-Aided Diagnosis with Deep Learning Architecture: Applications to Breast Lesions in US Images and Pulmonary Nodules in CT Scans. *Scientific Reports*, 6(1), July 2016. ISSN 2045-2322. doi: 10.1038/srep24454. URL http://www.nature.com/articles/srep24454.

E. Coche, B. Ghaye, J. Mey, and P. Duyck. *Comparative interpretation of CT and standard radiography of the chest*. Springer Science & Business Media, 2011.

A. Cook. Global Average Pooling Layers for Object Localization, 2017. URL https://alexisbcook.github.io/2017/global-average-pooling-layers-for-object-localization/.

CyberAILab. SegNet - An Image-Segmentation Neural Network, July 2018. URL https://www.cyberailab.com/home/segnet-an-image-segmentation-neural-network.

B. Dashtbozorg, A. M. Mendonça, and A. Campilho. Optic disc segmentation using the sliding band filter. *Computers in Biology and Medicine*, 56:1–12, Jan. 2015. ISSN 00104825. doi: 10.1016/j.compbiomed.2014.10.009. URL https://linkinghub.elsevier.com/retrieve/pii/S0010482514002832.

J. Dehmeshki, H. Amin, M. Valdivieso, and Xujiong Ye. Segmentation of Pulmonary Nodules in Thoracic CT Scans: A Region Growing Approach. *IEEE Transactions on Medical Imaging*, 27(4):467–480, Apr. 2008. ISSN 0278-0062. doi: 10.1109/TMI.2007.907555. URL http://ieeexplore.ieee.org/document/4359069/.

L. Deng. Deep Learning: Methods and Applications. *Foundations and Trends® in Signal Processing*, 7(3-4):197–387, 2014. ISSN 1932-8346, 1932-8354. doi: 10.1561/2000000039. URL http://nowpublishers.com/articles/foundations-and-trends-in-signal-processing/SIG-039.

Y. Deng. Why does batch normalization help? - Quora, 2017. URL https://www.quora.com/Why-does-batch-normalization-help.

S. Diciotti, G. Picozzi, M. Falchini, M. Mascalchi, N. Villari, and G. Valli. 3-D Segmentation Algorithm of Small Lung Nodules in Spiral CT Images. *IEEE Transactions on Information Technology in Biomedicine*, 12(1):7–19, Jan. 2008. ISSN 1089-7771. doi: 10.1109/TITB.2007.899504. URL http://ieeexplore.ieee.org/document/4358889/.

T. Do Nhu, S.-D. Joo, H.-J. Yang, S. Taek Jung, and S. Kim. Knee Bone Tumor Segmentation from radiographs using Seg-Unet with Dice Loss. Feb. 2019.

H. Dong, G. Yang, F. Liu, Y. Mo, and Y. Guo. Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks. In M. Valdés Hernández and V. González-Castro, editors, *Medical Image Understanding and Analysis*, Communications in Computer and Information Science, pages 506–517. Springer International Publishing, 2017. ISBN 978-3-319-60964-5.

N. Donges. Gradient Descent in a Nutshell, Mar. 2018. URL https://towardsdatascience.com/gradient-descent-in-a-nutshell-eaf8c18212f0.

D. Dubois, P. Hájek, and H. Prade. Knowledge-driven versus data-driven logics. *Journal of logic, Language and information*, pages 65–89, 2000. URL https://link.springer.com/article/10.1023/A:1008370109997.

K. Eremenko. The Ultimate Guide to Artificial Neural Networks (ANN), 2018. URL https://www.superdatascience.com/blogs/the-ultimate-guide-to-artificial-neural-networks-ann.

A. Escontrela. Convolutional Neural Networks from the ground up, June 2018. URL https://towardsdatascience.com/convolutional-neural-networks-from-the-ground-up-c67bb41454e1.

T. Esteves, P. Quelhas, A. M. Mendonça, and A. Campilho. Gradient convergence filters and a phase congruency approach for in vivo cell nuclei detection. *Machine Vision and Applications*, 23(4):623–638, July 2012. ISSN 0932-8092, 1432-1769. doi: 10.1007/s00138-012-0407-7. URL http://link.springer.com/10.1007/s00138-012-0407-7.

L. Fan, J. Qian, B. L. Odry, H. Shen, D. Naidich, G. Kohl, and E. Klotz. Automatic segmentation of pulmonary nodules by using dynamic 3d cross-correlation for interactive CAD systems. page 1362, San Diego, CA, May 2002. doi: 10.1117/12.467100. URL http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.467100.

X. Feng, J. Yang, A. F. Laine, and E. D. Angelini. Discriminative Localization in CNNs for Weakly-Supervised Segmentation of Pulmonary Nodules. In M. Descoteaux, L. Maier-Hein, A. Franz, P. Jannin, D. L. Collins, and S. Duchesne, editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2017*, volume 10435, pages 568–576. Springer International Publishing, Cham, 2017. ISBN 978-3-319-66178-0 978-3-319-66179-7. doi: 10.1007/978-3-319-66179-7_65. URL http://link.springer.com/10.1007/978-3-319-66179-7_65.

A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez. A Review on Deep Learning Techniques Applied to Semantic Segmentation. *arXiv:1704.06857 [cs]*, Apr. 2017. URL http://arxiv.org/abs/1704.06857. arXiv: 1704.06857.

D. Gilleman. Convolutional network - Deep learning essentials, 2018. URL http://www.deeplearningessentials.science/convolutionalNetwork/.

A. Goku. Topic DL01: Activation functions and its Types in Artifical Neural network, Mar. 2018. URL https://medium.com/@abhigoku10/activation-functions-and-its-types-in-artifical-neural-network-14511f3080a8.

V. Gupta. Understanding Feedforward Neural Networks | Learn OpenCV, 2017. URL https://www.learnopencv.com/understanding-feedforward-neural-networks/.

D. M. Hansell, A. A. Bankier, H. MacMahon, T. C. McLoud, N. L. Müller, and J. Remy. Fleischner Society: Glossary of Terms for Thoracic Imaging. *Radiology*, 246(3):697–722, Mar. 2008. ISSN 0033-8419, 1527-1315. doi: 10.1148/radiol.2462070712. URL http://pubs.rsna.org/doi/10.1148/radiol.2462070712.

S. Haykin. *Neural networks: a comprehensive foundation*. Pearson Education, Delhi, 1999. ISBN 978-81-7808-300-1. OCLC: 643435359.

K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, Dec. 2015. URL http://arxiv.org/abs/1512.03385. arXiv: 1512.03385.

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. *arXiv:1703.06870 [cs]*, Mar. 2017. URL http://arxiv.org/abs/1703.06870. arXiv: 1703.06870.

C. I. Henschke, D. F. Yankelevitz, R. Mirtcheva, G. McGuinness, D. McCauley, and O. S. Miettinen. CT Screening for Lung Cancer: Frequency and Significance of Part-Solid and Nonsolid Nodules. *American Journal of Roentgenology*, 178(5):1053–1057, May 2002. ISSN 0361-803X, 1546-3141. doi: 10.2214/ajr.178.5.1781053. URL http://www.ajronline.org/doi/10.2214/ajr.178.5.1781053.

X. Huang, W. Sun, T.-L. B. Tseng, C. Li, and W. Qian. Fast and fully-automated detection and segmentation of pulmonary nodules in thoracic CT scans using deep convolutional neural networks. *Computerized Medical Imaging and Graphics*, 74:25–36, June 2019. ISSN 0895-6111. doi: 10.1016/j.compmedimag.2019.02.003. URL http://www.sciencedirect.com/science/article/pii/S0895611118305366.

J. Hui. Image segmentation with Mask R-CNN, Apr. 2018. URL https://medium.com/@jonathan_hui/image-segmentation-with-mask-r-cnn-ebe6d793272.

S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]*, Feb. 2015. URL http://arxiv.org/abs/1502.03167. arXiv: 1502.03167.

L. Jacobson. Introduction to Artificial Neural Networks - Part 1, 2013. URL http://www.theprojectspot.com/tutorial-post/introduction-to-artificial-neural-networks-part-1/7.

A. Jain, Jianchang Mao, and K. Mohiuddin. Artificial neural networks: a tutorial. *Computer*, 29(3):31–44, Mar. 1996. ISSN 00189162. doi: 10.1109/2.485891. URL http://ieeexplore.ieee.org/document/485891/.

S. Jain. An Overview of Regularization Techniques in Deep Learning (with Python code), Apr. 2018. URL https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/.

F. Jiang, A. Grigorev, S. Rho, Z. Tian, Y. Fu, W. Jifara, K. Adil, and S. Liu. Medical image semantic segmentation based on deep learning. *Neural Computing and Applications*, 29(5): 1257–1265, Mar. 2018. ISSN 0941-0643, 1433-3058. doi: 10.1007/s00521-017-3158-6. URL http://link.springer.com/10.1007/s00521-017-3158-6.

Jun Wei, Y. Hagihara, and H. Kobatake. Detection of cancerous tumors on chest X-ray images -candidate detection filter and its evaluation. In *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, volume 3, pages 397–401, Kobe, Japan, 1999. IEEE. ISBN 978-0-7803-5467-8. doi: 10.1109/ICIP.1999.817143. URL http://ieeexplore.ieee.org/document/817143/.

J. Jung, H. Hong, and J. M. Goo. Ground-glass nodule segmentation in chest CT images using asymmetric multi-phase deformable model and pulmonary vessel removal. *Computers in Biology and Medicine*, 92:128–138, Jan. 2018. ISSN 00104825. doi: 10.1016/j.compbiomed.2017.11.013. URL https://linkinghub.elsevier.com/retrieve/pii/S0010482517303839.

P. Kamra, R. Vishraj, Kanica, and S. Gupta. Performance comparison of image segmentation techniques for lung nodule detection in CT images. In *2015 International Conference on Signal Processing, Computing and Control (ISPCC)*, pages 302–306, Waknaghat, Solan, India, Sept. 2015. IEEE. ISBN 978-1-4799-8436-7. doi: 10.1109/ISPCC.2015.7375045. URL http://ieeexplore.ieee.org/document/7375045/.

A. Karpathy. CS231n Convolutional Neural Networks for Visual Recognition, 2015. URL http://cs231n.github.io/convolutional-networks/.

E. A. Kazerooni and B. H. Gross. *Cardiopulmonary imaging*. The core curriculum. Lippincott Williams & Wilkins, Philadelphia, 2004. ISBN 978-0-7817-3655-8.

N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. *arXiv:1609.04836 [cs, math]*, Sept. 2016. URL http://arxiv.org/abs/1609.04836. arXiv: 1609.04836.

B. Khagi and G.-R. Kwon. Pixel-Label-Based Segmentation of Cross-Sectional Brain MRI Using Simplified SegNet Architecture-Based CNN. *Journal of Healthcare Engineering*, 2018: 1–8, Oct. 2018. ISSN 2040-2295, 2040-2309. doi: 10.1155/2018/3640705. URL https://www.hindawi.com/journals/jhe/2018/3640705/.

A. Khan, H. Al-Jahdali, C. Allen, K. Irion, S. Al Ghanem, and S. Koteyar. The calcified lung nodule: What does it mean? *Annals of Thoracic Medicine*, 5(2):67, 2010. ISSN 1817-1737. doi: 10.4103/1817-1737.62469. URL http://www.thoracicmedicine.org/text.asp?2010/5/2/67/62469.

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, Dec. 2014. URL http://arxiv.org/abs/1412.6980. arXiv: 1412.6980.

E. Kopelowitz and G. Englehard. Lung Nodules Detection and Segmentation using 3d Mask R-CNN. London, UK, 2019. URL https://openreview.net/pdf?id=Hkxqw5ilcV.

W. Kostis, A. Reeves, D. Yankelevitz, and C. Henschke. Three-dimensional segmentation and growth-rate estimation of small pulmonary nodules in helical ct images. *IEEE Transactions on Medical Imaging*, 22(10):1259–1274, Oct. 2003. ISSN 0278-0062. doi: 10.1109/TMI.2003.817785. URL http://ieeexplore.ieee.org/document/1233924/.

T. Kubota, A. K. Jerebko, M. Dewan, M. Salganicoff, and A. Krishnan. Segmentation of pulmonary nodules of various densities with morphological approaches and convexity models. *Medical Image Analysis*, 15(1):133–154, Feb. 2011. ISSN 13618415. doi: 10.1016/j.media.2010.08.005. URL https://linkinghub.elsevier.com/retrieve/pii/S136184151000109X.

P. Kumar, P. Nagar, C. Arora, and A. Gupta. U-SegNet: Fully Convolutional Neural Network based Automated Brain tissue segmentation Tool. *arXiv:1806.04429 [cs]*, June 2018. URL http://arxiv.org/abs/1806.04429. arXiv: 1806.04429.

M. Lanham. Gradient descent explained - Learn ARCore - Fundamentals of Google ARCore [Book], 2018. URL https://www.oreilly.com/library/view/learn-arcore-/9781788830409/e24a657a-a5c6-4ff2-b9ea-9418a7a5d24c.xhtml.

Y. LeCun. LeNet-5, convolutional neural networks, 2015. URL http://yann.lecun.com/exdb/lenet/.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature14539. URL http://www.nature.com/articles/nature14539.

J. H. Lee, H. S. Ahn, D. H. Choi, and K. S. Tae. Evaluation on the Usefulness of X-ray Computer-Aided Detection (CAD) System for Pulmonary Tuberculosis (PTB) using SegNet. *Journal of Biomedical Engineering Research*, 38(1):25–31, 2017. ISSN 1229-0807. doi: 10.9718/JBER.2017.38.1.25. URL http://www.koreascience.or.kr/article/JAKO201713056892701.page.

B. Li, Q. Chen, G. Peng, Y. Guo, K. Chen, L. Tian, S. Ou, and L. Wang. Segmentation of pulmonary nodules using adaptive local region energy with probability density function-based similarity distance and multi-features clustering. *BioMedical Engineering OnLine*, 15(1), Dec. 2016. ISSN 1475-925X. doi: 10.1186/s12938-016-0164-3. URL http://biomedical-engineering-online.biomedcentral.com/articles/10.1186/s12938-016-0164-3.

T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature Pyramid Networks for Object Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, Honolulu, HI, July 2017. IEEE. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.106. URL http://ieeexplore.ieee.org/document/8099589/.

G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, Dec. 2017. ISSN 13618415. doi: 10.1016/j.media.2017.07.005. URL https://linkinghub.elsevier.com/retrieve/pii/S1361841517301135.

F. Liu, Z. Zhou, H. Jang, A. Samsonov, G. Zhao, and R. Kijowski. Deep convolutional neural network and 3d deformable approach for tissue segmentation in musculoskeletal magnetic resonance imaging: Deep Learning Approach for Segmenting MR Image. *Magnetic Resonance in Medicine*, 79(4):2379–2391, Apr. 2018a. ISSN 07403194. doi: 10.1002/mrm.26841. URL http://doi.wiley.com/10.1002/mrm.26841.

M. Liu, J. Dong, X. Dong, H. Yu, and L. Qi. Segmentation of Lung Nodule in CT Images Based on Mask R-CNN. In *2018 9th International Conference on Awareness Science and Technology (iCAST)*, pages 1–6, Sept. 2018b. doi: 10.1109/ICAwST.2018.8517248.

J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. *arXiv:1411.4038 [cs]*, Nov. 2014. URL http://arxiv.org/abs/1411.4038. arXiv: 1411.4038.

V. Maini and S. Sabri. *Machine Learning for Humans*. e-book, 2017. URL https://www.dropbox.com/s/e38nil1dnl7481q/machine_learning.pdf?dl=0.

J. Malo, C. Luraschi-Monjagatta, D. M. Wolk, R. Thompson, C. A. Hage, and K. S. Knox. Update on the Diagnosis of Pulmonary Coccidioidomycosis. *Annals of the American Thoracic Society*, 11(2):243–253, Feb. 2014. ISSN 2325-6621. doi: 10.1513/AnnalsATS.201308-286FR. URL http://www.atsjournals.org/doi/abs/10.1513/AnnalsATS.201308-286FR.

M. F. McNitt-Gray, S. G. Armato, C. R. Meyer, A. P. Reeves, G. McLennan, R. C. Pais, J. Freymann, M. S. Brown, R. M. Engelmann, P. H. Bland, G. E. Laderach, C. Piker, J. Guo, Z. Towfic, D. P.-Y. Qing, D. F. Yankelevitz, D. R. Aberle, E. J. van Beek, H. MacMahon, E. A. Kazerooni, B. Y. Croft, and L. P. Clarke. The Lung Image Database Consortium (LIDC) Data Collection Process for Nodule Detection and Annotation. *Academic Radiology*, 14(12):1464–1474, Dec. 2007. ISSN 10766332. doi: 10.1016/j.acra.2007.07.021. URL http://linkinghub.elsevier.com/retrieve/pii/S1076633207004497.

A. Mikolajczyk and M. Grochowski. Data augmentation for improving deep learning in image classification problem. In *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, pages 117–122, Swinoujście, May 2018. IEEE. ISBN 978-1-5386-6143-7. doi: 10.1109/IIPHDW.2018.8388338. URL https://ieeexplore.ieee.org/document/8388338/.

F. Milletari, N. Navab, and S.-A. Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. *arXiv:1606.04797 [cs]*, June 2016. URL http://arxiv.org/abs/1606.04797. arXiv: 1606.04797.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2018. ISBN 978-0-262-01825-8.

J. H. Moltz, L. Bornemann, J. Kuhnigk, V. Dicken, E. Peitgen, S. Meier, H. Bolte, M. Fabel, H. Bauknecht, M. Hittinger, A. Kießling, M. Pusken, and H. Peitgen. Advanced Segmentation Techniques for Lung Nodules, Liver Metastases, and Enlarged Lymph Nodes in CT Scans. *IEEE Journal of Selected Topics in Signal Processing*, 3(1):122–134, Feb. 2009. ISSN 1932-4553. doi: 10.1109/JSTSP.2008.2011107.

F. C. Monteiro and A. C. Campilho. Distance measures for image segmentation evaluation. pages 794–797, Kos, Greece, 2012. doi: 10.1063/1.4756257. URL http://aip.scitation.org/doi/abs/10.1063/1.4756257.

G. Motley, N. Dalrymple, C. Keesling, J. Fischer, and W. Harmon. Hounsfield unit density in the determination of urinary stone composition. *Urology*, 58(2):170–173, Aug. 2001. ISSN 00904295. doi: 10.1016/S0090-4295(01)01115-3. URL http://linkinghub.elsevier.com/retrieve/pii/S0090429501011153.

R. Naseem, K. S. Alimgeer, and T. Bashir. Recent trends in Computer Aided diagnosis of lung nodules in thorax CT scans. In *2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*, pages 1–12, Karachi, Pakistan, Apr. 2017. IEEE. ISBN 978-1-5090-3310-2. doi: 10.1109/ICIEECT.2017.7916548. URL http://ieeexplore.ieee.org/document/7916548/.

H. Noh, S. Hong, and B. Han. Learning Deconvolution Network for Semantic Segmentation. *arXiv:1505.04366 [cs]*, May 2015. URL http://arxiv.org/abs/1505.04366. arXiv: 1505.04366.

D. Ost, A. M. Fein, and S. H. Feinsilver. The Solitary Pulmonary Nodule. *New England Journal of Medicine*, 348(25):2535–2542, June 2003. ISSN 0028-4793, 1533-4406. doi: 10.1056/NEJMcp012290. URL http://www.nejm.org/doi/abs/10.1056/NEJMcp012290.

S. Patel. Introduction to Deep Learning: What Is Deep Learning?, 2017. URL https://uk.mathworks.com/videos/introduction-to-deep-learning-what-is-deep-learning--1489502328819.html.

C. S. Pereira, H. Fernandes, A. M. Mendonça, and A. Campilho. Detection of Lung Nodule Candidates in Chest Radiographs. In J. Martí, J. M. Benedí, A. M. Mendonça, and J. Serrat, editors, *Pattern Recognition and Image Analysis*, volume 4478, pages 170–177. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007a. ISBN 978-3-540-72848-1 978-3-540-72849-8. doi: 10.1007/978-3-540-72849-8_22. URL http://link.springer.com/10.1007/978-3-540-72849-8_22.

C. S. Pereira, A. M. Mendonça, and A. Campilho. Evaluation of Contrast Enhancement Filters for Lung Nodule Detection. In M. Kamel and A. Campilho, editors, *Image Analysis and Recognition*, volume 4633, pages 878–888. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007b. ISBN 978-3-540-74258-6 978-3-540-74260-9. doi: 10.1007/978-3-540-74260-9_78. URL http://link.springer.com/10.1007/978-3-540-74260-9_78.

L. Perez and J. Wang. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *arXiv:1712.04621 [cs]*, Dec. 2017. URL http://arxiv.org/abs/1712.04621. arXiv: 1712.04621.

P. Quelhas, M. Marcuzzo, A. M. Mendonca, and A. Campilho. Cell Nuclei and Cytoplasm Joint Segmentation Using the Sliding Band Filter. *IEEE Transactions on Medical Imaging*, 29(8): 1463–1473, Aug. 2010. ISSN 0278-0062, 1558-254X. doi: 10.1109/TMI.2010.2048253. URL http://ieeexplore.ieee.org/document/5477157/.

B. Romera-Paredes and P. H. S. Torr. Recurrent Instance Segmentation. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, volume 9910, pages 312–329. Springer International Publishing, Cham, 2016. ISBN 978-3-319-46465-7 978-3-319-46466-4. doi: 10.1007/978-3-319-46466-4_19. URL http://link.springer.com/10.1007/978-3-319-46466-4_19.

O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, volume 9351, pages 234–241. Springer International Publishing, Cham, 2015. ISBN 978-3-319-24573-7 978-3-319-24574-4. doi: 10.1007/978-3-319-24574-4_28. URL http://link.springer.com/10.1007/978-3-319-24574-4_28.

H. R. Roth, C. Shen, H. Oda, M. Oda, Y. Hayashi, K. Misawa, and K. Mori. Deep learning and its application to medical image segmentation. *arXiv:1803.08691 [cs]*, Mar. 2018. doi: 10.11409/mit.36.63. URL http://arxiv.org/abs/1803.08691. arXiv: 1803.08691.

J. Schmidhuber. Deep Learning in Neural Networks: An Overview. *Neural Networks*, 61:85–117, Jan. 2015. ISSN 08936080. doi: 10.1016/j.neunet.2014.09.003. URL http://arxiv.org/abs/1404.7828. arXiv: 1404.7828.

A. A. A. Setio, F. Ciompi, G. Litjens, P. Gerke, C. Jacobs, S. J. van Riel, M. M. W. Wille, M. Naqibullah, C. I. Sanchez, and B. van Ginneken. Pulmonary Nodule Detection in CT Images: False Positive Reduction Using Multi-View Convolutional Networks. *IEEE Transactions on Medical Imaging*, 35(5):1160–1169, May 2016. ISSN 0278-0062, 1558-254X. doi: 10.1109/TMI.2016.2536809. URL http://ieeexplore.ieee.org/document/7422783/.

A. Sevastopolsky. Optic disc and cup segmentation methods for glaucoma detection with modification of U-Net convolutional neural network. *Pattern Recognition and Image Analysis*, 27(3):618–624, July 2017. ISSN 1555-6212. doi: 10.1134/S1054661817030269. URL https://doi.org/10.1134/S1054661817030269.

E. Shakibapour, A. Cunha, G. Aresta, A. M. Mendonça, and A. Campilho. An unsupervised metaheuristic search approach for segmentation and volume measurement of pulmonary nodules in lung CT scans. *Expert Systems with Applications*, 119:415–428, Apr. 2019. ISSN 09574174. doi: 10.1016/j.eswa.2018.11.010. URL https://linkinghub.elsevier.com/retrieve/pii/S0957417418307292.

S. Sharma. Activation Functions in Neural Networks, Sept. 2017. URL https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*, Sept. 2014. URL http://arxiv.org/abs/1409.1556. arXiv: 1409.1556.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. URL http://jmlr.org/papers/volume15/srivastava14a.old/srivastava14a.pdf.

R. Tachibana and S. Kido. Automatic segmentation of pulmonary nodules on CT images by use of NCI lung image database consortium. page 61440M, San Diego, CA, Mar. 2006. doi: 10.1117/12.653366. URL http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.653366.

A. A. Taha and A. Hanbury. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC Medical Imaging*, 15(1), Dec. 2015. ISSN 1471-2342. doi: 10.1186/

s12880-015-0068-x. URL http://bmcmedimaging.biomedcentral.com/articles/10.1186/s12880-015-0068-x.

F. B. Tek, A. G. Dempster, and I. Kale. Blood Cell Segmentation Using Minimum Area Watershed and Circle Radon Transformations. In C. Ronse, L. Najman, and E. Decencière, editors, *Mathematical Morphology: 40 Years On*, volume 30, pages 441–454. Springer-Verlag, Berlin/Heidelberg, 2005. ISBN 978-1-4020-3442-8. doi: 10.1007/1-4020-3443-1_40. URL http://link.springer.com/10.1007/1-4020-3443-1_40.

G. Tong, Y. Li, H. Chen, Q. Zhang, and H. Jiang. Improved U-NET network for pulmonary nodules segmentation. *Optik*, 174:460–469, Dec. 2018. ISSN 00304026. doi: 10.1016/j.ijleo.2018.08.086. URL https://linkinghub.elsevier.com/retrieve/pii/S003040261831235X.

L. A. Torre, R. L. Siegel, and A. Jemal. Lung Cancer Statistics. In A. Ahmad and S. Gadgeel, editors, *Lung Cancer and Personalized Medicine*, volume 893, pages 1–19. Springer International Publishing, Cham, 2016. ISBN 978-3-319-24221-7 978-3-319-24223-1. doi: 10.1007/978-3-319-24223-1_1. URL http://link.springer.com/10.1007/978-3-319-24223-1_1.

R. Varma. Implementing a Neural Network in Python, 2017. URL https://rohanvarma.me/Neural-Net/.

J. Wang, R. Engelmann, and Q. Li. Segmentation of pulmonary nodules in three-dimensional CT images by use of a spiral-scanning technique: Nodule segmentation in 3d CT images. *Medical Physics*, 34(12):4678–4689, Nov. 2007. ISSN 00942405. doi: 10.1118/1.2799885. URL http://doi.wiley.com/10.1118/1.2799885.

S. Wang, M. Zhou, Z. Liu, Z. Liu, D. Gu, Y. Zang, D. Dong, O. Gevaert, and J. Tian. Central focused convolutional neural networks: Developing a data-driven model for lung nodule segmentation. *Medical Image Analysis*, 40:172–183, Aug. 2017. ISSN 13618415. doi: 10.1016/j.media.2017.06.014. URL https://linkinghub.elsevier.com/retrieve/pii/S1361841517301019.

W. Wang, Y. Lu, B. Wu, T. Chen, D. Z. Chen, and J. Wu. Deep Active Self-paced Learning for Accurate Pulmonary Nodule Segmentation. In A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, volume 11071, pages 723–731. Springer International Publishing, Cham, 2018. ISBN 978-3-030-00933-5 978-3-030-00934-2. doi: 10.1007/978-3-030-00934-2_80. URL http://link.springer.com/10.1007/978-3-030-00934-2_80.

T. W. Way, L. M. Hadjiiski, B. Sahiner, H.-P. Chan, P. N. Cascade, E. A. Kazerooni, N. Bogot, and C. Zhou. Computer-aided diagnosis of pulmonary nodules on CT scans: Segmentation and classification using 3d active contours: Computer-aided diagnosis of lung nodules on CT. *Medical Physics*, 33(7Part1):2323–2337, June 2006. ISSN 00942405. doi: 10.1118/1.2207129. URL http://doi.wiley.com/10.1118/1.2207129.

E. W. Weisstein. Affine Transformation, 2018. URL http://mathworld.wolfram.com/AffineTransformation.html.

H. T. Winer-Muram. The Solitary Pulmonary Nodule. *Radiology*, 239(1):34–49, Apr. 2006. ISSN 0033-8419, 1527-1315. doi: 10.1148/radiol.2391050343. URL http://pubs.rsna.org/doi/10.1148/radiol.2391050343.

T. Yang, J. Cheng, and C. Zhu. A segmentation of pulmonary nodules based on improved fuzzy C-means clustering algorithm. *MATEC Web of Conferences*, 232:03011, 2018. ISSN 2261-236X. doi: 10.1051/matecconf/201823203011. URL https://www.matec-conferences.org/10.1051/matecconf/201823203011.

Ying-Lun Fok, J. Chan, and R. Chin. Automated analysis of nerve-cell images using active contour models. *IEEE Transactions on Medical Imaging*, 15(3):353–368, June 1996. ISSN 02780062. doi: 10.1109/42.500144. URL http://ieeexplore.ieee.org/document/500144/.

S. S. F. Yip, C. Parmar, D. Blezek, R. S. J. Estepar, S. Pieper, J. Kim, and H. J. W. L. Aerts. Application of the 3d slicer chest imaging platform segmentation algorithm for large lung nodule delineation. *PLOS ONE*, 12(6):e0178944, June 2017. ISSN 1932-6203. doi: 10.1371/journal.pone.0178944. URL https://dx.plos.org/10.1371/journal.pone.0178944.

W. Zhang, X. Zhang, J. Zhao, Y. Qiang, Q. Tian, and X. Tang. A segmentation method for lung nodule image sequences based on superpixels and density-based spatial clustering of applications with noise. *PLOS ONE*, 12(9):e0184290, Sept. 2017. ISSN 1932-6203. doi: 10.1371/journal.pone.0184290. URL https://dx.plos.org/10.1371/journal.pone.0184290.

Z.-H. Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1): 44–53, Jan. 2018. ISSN 2095-5138, 2053-714X. doi: 10.1093/nsr/nwx106. URL https://academic.oup.com/nsr/article/5/1/44/4093912.

D. Ziganto. Model Tuning (Part 2 - Validation & Cross-Validation), Jan. 2018. URL https://dziganto.github.io/cross-validation/data%20science/machine%20learning/model%20tuning/python/Model-Tuning-with-Validation-and-Cross-Validation/.