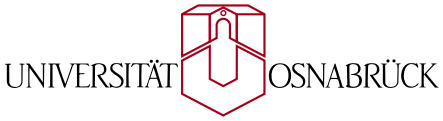

Institute of Cognitive Science
Biologically Inspired Computer Vision Group

Master's Thesis

Nodule Detection in Lung CT Scans with Deep Learning

Andrea Suckro

December 27, 2017

First supervisor: Dr. Ulf Krumnack
Second supervisor: Prof. Dr. Gunther Heidemann

Nodule Detection in Lung CT Scans with Deep Learning

This thesis makes use of a 3D Convolutional Deep Neural Network to classify whether a certain lung region contains a nodule or not. The network is trained on an open dataset containing roughly 1000 patient's full lung scan. The network is analyzed, its features visualized and compared to already existing solutions.

Contents

Chapter 1

Introduction

This master thesis centers around the automated detection of nodules in lung region CT scans. The introduction will cover the medical context necessary to understand the problem and motivate the use of Deep Neural Networks to assist in the task of nodule detection. It also explains the research question at hand and the structure of the thesis with an overview of the sections to come.

1.1 Medical Context

In 2012 34,490 men and 18,030 women in Germany were diagnosed with an illness corresponding to the ICD-10 code C33-34 [**koch2015krebs**]. This code describes malignant tumors in the breathable tract more generally summarized as lung cancer. 43,499 people died from this illness, which makes it one of the most dangerous types of cancer in Germany. The international comparison shows that other countries too suffer under its impact (see Figure 1.1).

The main risk factor has been shown to be the exposure to tobacco smoke, whether it's active consumption via cigarettes, or passive exposure especially in closed rooms. CT scanners are used to detect irregular tissue in a patients lung region. These irregularities are grouped under the term *pulmonary nodule*. A pulmonary nodule is a small, round (parenchymal) or worm (juxtapleural) shaped lesion in the lungs. Each lesion has a chance to be malignant and may grow and spread over time, becoming a risk for the patient's life in the form of lung cancer. Nodules come not only in different shapes but differ along other features as well. Ground-glass opacity (GGO) nodules are a challenging kind of nodules since they are not thoroughly solid and so harder to detect on a CT scan. The location of the nodules is crucial for the detection rate since nodules close to bigger vessels or the chest wall do not differ much in intensity to the surrounding tissue and can be easily overlooked by the radiologist.

1.2 Current Medical Approach

In the clinical setting, the supervising medical staff has additional information available apart from the CT scans. Patients are either in a high-risk group (over a certain age and heavy

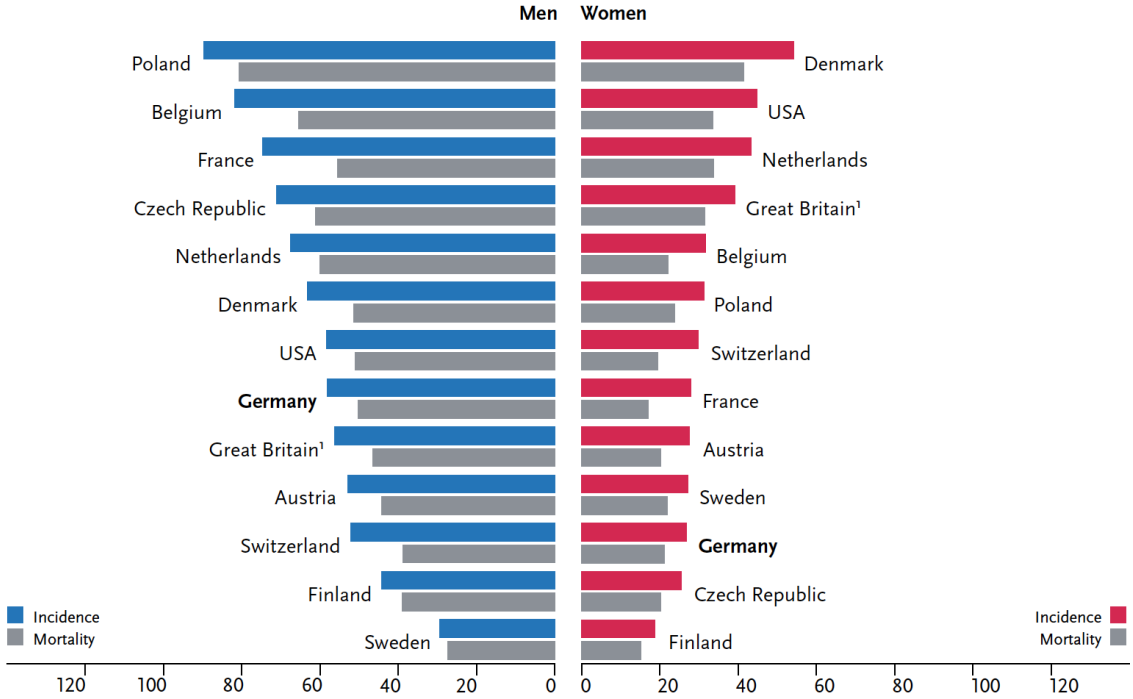

Figure 1.1: International comparison of age-standardized incidence and mortality rates for lung cancer in the year 2012

smokers) or present certain syndromes like fatigue, weight loss, cough, dyspnea, hemoptysis and chest pain. Different methods are used to find the underlying cause of these symptoms:

Sputum cytology Sputum is mucus that is produced in the lower airways which can be analyzed for bacteria and irregular cells.

Biopsy In a biopsy tissue samples are directly taken from the lung of the patient. This may be done in several ways, either bronchoscopic (obtaining the sample with an instrument through the airways) or directly by a needle or incision through the chest wall.

Imaging tests two main imaging techniques are available for analyzing the lung: X-Ray scans and MRI. The data used in this thesis stems belongs to the first group, more specifically from *low-dose computed tomography* (LDCT) scans. The radiation used in this method lies around 2.35 mSv which corresponds to roughly 80% less radiation than regular CT scans [**ono2013low**].

Lung cancer is tricky to detect since the symptoms show up very late in the process of the illness and it is often too late for the patient when those are recognized. Sputum cytology and biopsy are only used when there is already circumstantial evidence (like other symptoms) for lung cancer which may as well be too late in the development of the illness for a successful treatment. This makes imaging techniques the only source for early detection. Radiologists

are required to analyze roughly 100-600 pictures per patient depending on the slice thickness of the used technique and the body height of the patient. Medical image viewers like OsiriX (Figure 1.2) are assisting in this process and have also been used in this thesis to visualize the image data.

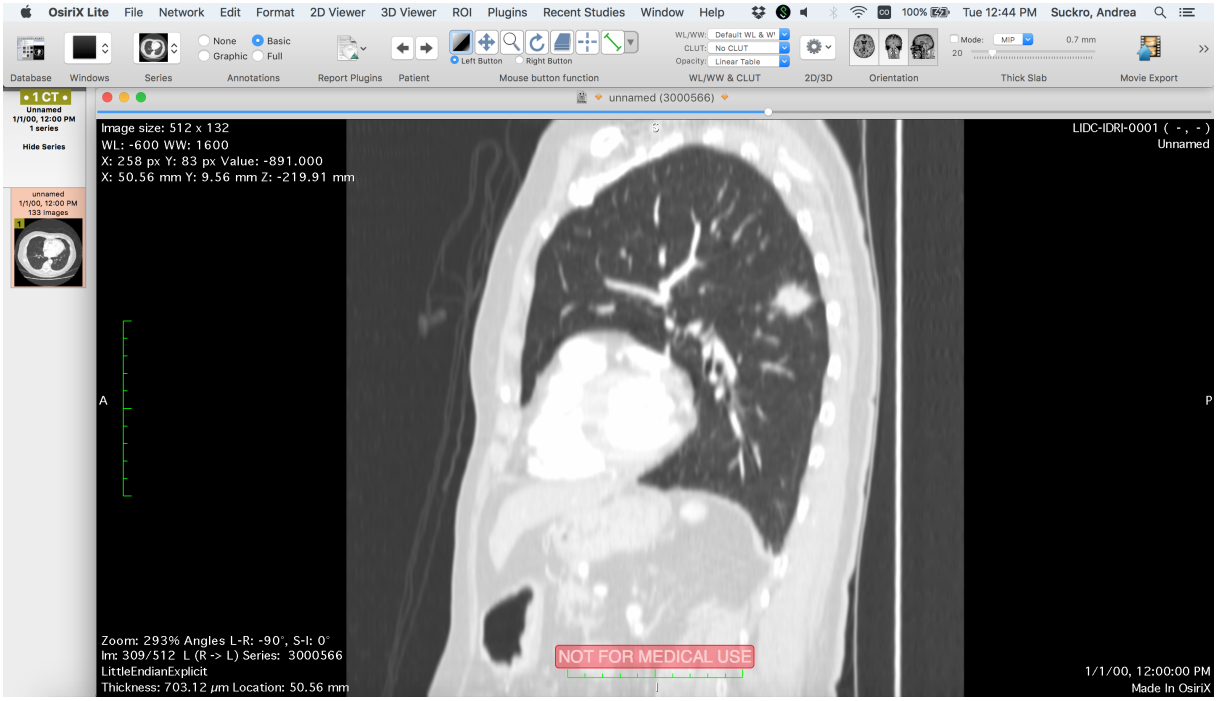

Figure 1.2: Screenshot of the analyzing software OsiriX. It is the most widely used software in the domain of medical image viewers and offers a free trial version that was used for this thesis.

Yet it is still a highly complicated task and despite all efforts in this field the process is of course not fail-safe and happens that nodules are not detected in an early stage of their development but later when a successful treatment is not as likely anymore. A study by Kakinuma et. al.[**kakinuma2012comparison**] shows how different features like the slice thickness and the nodule properties affect the detection rate, dropping it to 65.5% for pure ground-glass opacities in a scan with 2mm slice thickness. Another issue arises with the false positive rate. A study by Pinsky et. al.[**pinsky2013national**] reports a mean false positive rate of 28.7% across 112 radiologists at 32 screening centers in and outside of the US. This puts additional stress on the patient and requires further tests to conclude that there is no cancer present.

1.3 Opportunities for Assistance

Early detection of lung cancer is important in two ways. First, it increases the chance of a successful treatment and secondly, it avoids unnecessary scans in the scenario of undetected cancerous material or wrongly classified abnormalities in the lung. As the number of preventive screening procedures rises and the used scanners become more granular software-based assistance

becomes a helpful companion to every radiologist [**li2005computer**].

The scientific literature provides many algorithmic approaches to finding nodules in CT Scans[1] using handcrafted features and in-depth knowledge of the structure of the data and the nodules that should be detected. Some make already use of Deep Convolutional Neural Networks to solve this or related tasks (like classifying the malignancy of a nodule)[2].

Despite many efforts being devoted to the computer-aided nodule detection problem, lung CAD systems remain an ongoing research topic. One of the major difficulties is the detection of GGO nodules with low-dose thin-slice CT screening. Another two difficulties are the detection of nodules that are adjacent to vessels or the chest wall when they have very similar density and the detection of nodules that are nonspherical in shape. In such cases, classical approaches like intensity thresholding or model-based methods might fail to identify the nodules correctly.

1.4 The Problem of Neural Networks

Neural Networks are strong tools that *solve* many tasks with astonishing performance (see for example the mastery of the game Go [**silver2017alphagozero**]). Yet it seems like the solution they come up with is not intelligible to humans in the same way as feature detectors that have been directly designed to respond to geometric properties of the nodule. But in a scenario where medical decisions are based on the output of an algorithm, it is crucial that the algorithm is reliable and the way it comes up with a decision is accepted by the people responsible. As long as this is not the case the specialists working with the software will not completely accept its value and it can not be ruled out that the network produces erroneous results in cases that did not occur during the training phase.

In this thesis, a Deep Neural Network will be developed that uses 3D CT scan information to classify a given CT image patch as either containing a nodule or not. Then mechanisms will be applied to visualize the learned features and discussed whether they can be conceptualized in terms of already approved procedures. The problem in this sense is two-fold. First - can a 3D Convolutional Neural Network solve the task of nodule detection and second, how can it's solution to the problem be understood?

1.5 Structure of this thesis

This thesis is structured in the following way: in chapter 2 the used dataset is described as well as it's preprocessing for the learning algorithm. The 3rd chapter will give an overview of the field and explain some of the current computational approaches that are used for detecting nodules. The methods chapter 4 contains information about the used software packages that were necessary to implement the algorithms as well as information about Convolutional Neural Networks in general. Chapter 5 describes the properties of the used model and chapter 6 gives an overview of its performance and learned features. In chapter 7 the results are revised and items for further investigation and optimization are presented together with a more general outlook on

[1] **armato1999computerized**; **armato2001automated**; **okada2005robust**; **tao2009multi**; **ye2009shape**.

[2] **cheng2016computer**; **huang2017lung**; **shen2015multi**.

the topic of analyzing Neural Networks to gain insights into problems and not just as solutions to them.

Chapter 2

Dataset

The Lung Image Database Consortium image collection (LIDC-IDRI) is a publicly available dataset that was generated through the joined effort of seven academic centers and eight medical imaging companies. It contains data of 1018 cases which consists of diagnostic and lung cancer screening thoracic computed tomography (CT) scans with marked-up annotated lesions. More information on the origin of the dataset can be found in the appendix B.1 and the reference paper by Armato [**armato2011lung**]. The following sections provide an overview of the data structure and the implemented preprocessing.

2.1 Content and Structure

The dataset contains a folder for each patient. These folders contain a full chest CT scan and the annotations by the radiologists. The CT scan data is encoded in a list of DICOM (Digital Imaging and Communications in Medicine) files and the annotations as one XML (extended markup language) file. The structure of both file types is described in the upcoming sections. More details on how the dataset was formed and how the images are generated by the scanners as well as the parameters that play into that can be found in Appendix B.

2.1.1 Scan Data Structure

DICOM is a file format for storing medical images with for the use case relevant meta information. It is not only used for CT scans but also for radiography, ultrasonography and MRI data. It was initially introduced by the American College of Radiology (ACR) and National Electrical Manufacturers Association (NEMA) under the name ACR/NEMA 300 in 1985, but further redefined and finally in the third version released under the name DICOM in 1993 (see Pianykh [**pianykh2008**] for more information).

The cases that scans are provided for in the dataset fulfill certain criteria. The CT scans are exclusive of the lung and do not contain other body parts or organs. The reconstruction interval and collimation are kept at < 3mm. This means that there are differences in the resolution of the scan data (naturally through the different equipment used during recording), but that it is limited with an upper bound. Still one can find patients with $124 - 529$ recorded images in the data. The scan data may also include noise or other disruptive factors like metal (heart pacers

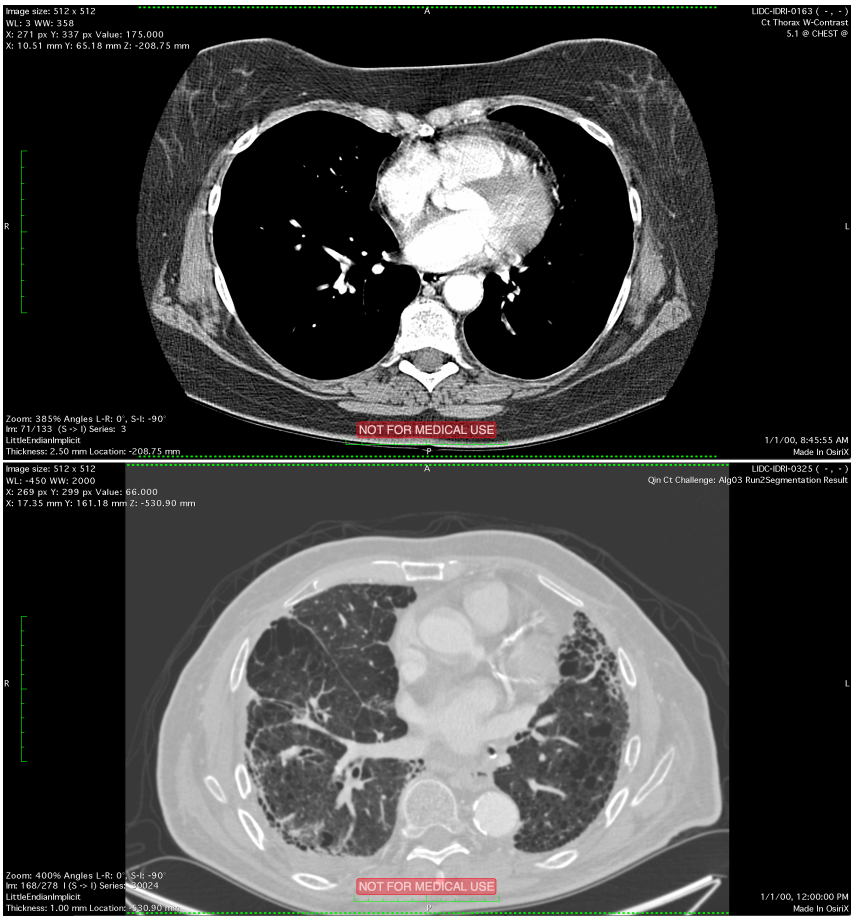

Figure 2.1: Two slices from different patients in the dataset. Images have been generated with OsiriX Lite [**rosset2004osirix**]. The information in the corner of the picture is extracted from the meta information stored in the DICOM format.

e.g.) as well as other pathological features as long as those do not interfere with the visibility of the nodules in a drastic sense.

The included cases have between 0 and 6 nodules with the longest diameter of between $3-30$ mm. The term nodule refers to a brought spectrum of tissue abnormalities and can represent not only lung cancer but also other metastatic diseases or non-cancerous processes or lesions that have a nodular morphology. Typical slices from the data can be seen in Figure 2.1.

The additional information about the patients that is usually stored with the scans (like age, name, gender) is anonymized in this dataset.

2.1.2 Annotation Structure

Two different types of nodules are encoded in the data: nodules with a diameter of $>= 3$ mm and nodules smaller than that. The big nodules have extensive information stored with them: a rich edge map which outlines a complete contour for them in all sections (as seen in Figure 2.2)

```
<noduleID>IL057_127581</noduleID>
<characteristics>
  <subtlety>4</subtlety>
  <malignancy>3</malignancy>
  [...]
</characteristics>

<edgeMap>
  <xCoord>103</xCoord>
  <yCoord>391</yCoord>
</edgeMap>

<imageZposition>-232.535004</imageZposition>

<edgeMap>
   <xCoord>104</xCoord>
   <yCoord>393</yCoord>
</edgeMap>
```

Figure 2.2: A shortened example XML annotation for a nodule with diameter >= 3mm.

and a measure for their characteristics (like their subtlety and malignancy on a scale from 1 to 5). This extra information has not been used in the learning process for this thesis but enables research on further classification and localization tasks.

Nodules with a smaller diameter have less information stored with them. They only contain the approximate center of mass for the nodule as seen in Figure 2.3.

```
<noduleID>7</noduleID>
<roi>
<imageZposition>-227.535004</imageZposition>
<imageSOP_UID>1.3.6.1.4...</imageSOP_UID>
<inclusion>TRUE</inclusion>
<edgeMap>
   <xCoord>127</xCoord>
   <yCoord>370</yCoord>
</edgeMap>
</roi>
```

Figure 2.3: Nodules with a diameter of < 3mm have only the center of mass stored.

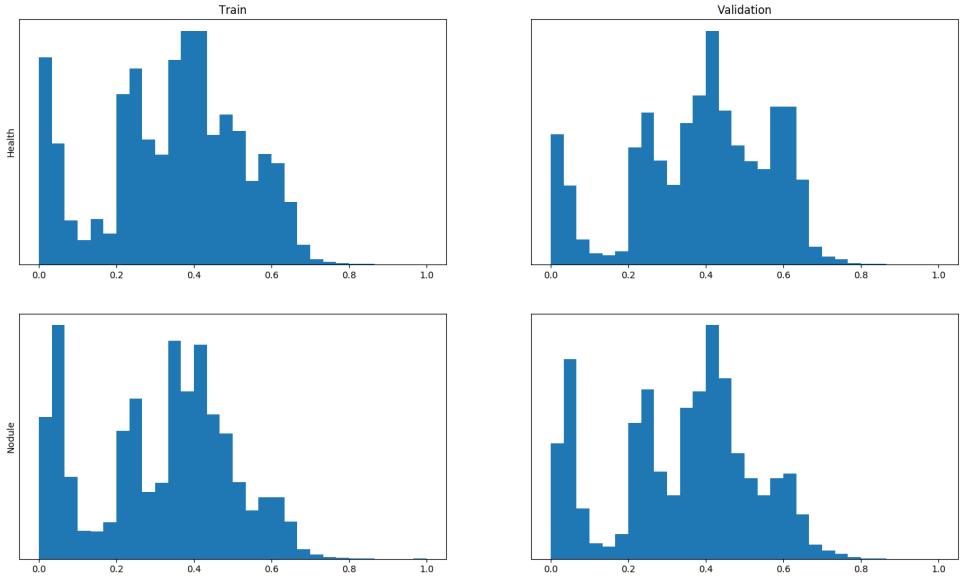

Figure 2.4: Distribution of the normalized pixel values for both classes in the training and validation set (1000 instances per set). This data is used for learning the network.

2.2 Preprocessing

The data is available in the form of sub-folders for each patient that contain the CT scan results in DICOM file format and the annotation data as XML files. The following sections describe the read in and slicing of the data.

2.2.1 Reading in the data

The whole dataset was randomly distributed to 3 folders in a 60 : 20 : 20 split ratio: train, validation and test. The training dataset is used during the learning process of the neural network. The validation dataset is used to measure the performance of the network while it is trained on the training dataset. See Figure 2.4 for a distribution of values for the two classes.

Each of the folders contains a list of patients containing one or more sub-folders with CT scans. Some of the extra folders for a patient contain only a few scans and not a complete CT. Those folders were ignored. With the use of the Python package dicom [**mason2011t**] the CT scans are converted to a 3-dimensional array. The annotation XML files are evaluated to find the location of the nodules. In the case of nodules with a diameter of $< 3mm$ the center of mass value is used, for the bigger nodules that have information of the whole edge map, the mean over all dimensions is used as an approximation for the center of mass of those nodules. 147 patient files contained no real nodules or had corrupted data (the XML file had a corrupted structure or wrong formatting that prevented the algorithm from extracting the nodule coordinates, one file

had a slice thickness of 0 e.g.). Those files have as well been excluded from the learning process.

2.2.2 Slicing the patches

The information from the annotations is used to generate the patches from the complete scan. A fixed number of patches is generated from the data per patient. The patches for the healthy data are cut randomly from the tissue that contains no nodules while the patches with the nodule information are randomly picked around the nodule's center. Determining the center of a nodule is easy in the cases of the nodules with $< 3mm$ diameter. The resulting shape of the patches is $(50, 50, 5)$. It makes sense to take less value in the z-direction since the resolution of the CT scans is lower in that direction (more information about the scanners can be found in the appendix B.3).

Chapter 3

Current Approaches

This chapter illustrates the current state of the art in the field of automated nodule detection. Roughly the field can be divided among the used methods in the *classical* approaches and the deep learning approaches. It also describes how this thesis is situated in the field and what is done differently compared to previous papers.

3.1 Classical Approaches

Nodule detection is a complex and potentially life-saving task, so it makes sense that there is a scientific community dedicated to finding algorithmic approaches to aid the radiologists. In this section some of the published papers in this domain and the techniques they use will be explained. In principal algorithms applied to the lung CT images can be subdivided in several stages (as can be for example seen in 3.1). Similar to other computer vision algorithms those can be roughly clustered in the following: Segmentation, Candidate Selection, Classification.

3.1.1 Segmentation

A lung CT scan contains more anatomical structure than just the lung area. It is necessary to exclude the trachea, heart as well as the spine from the slices in order to solely focus on the lung tissue. Armato et al. [**armato1999computerized**; **armato2001automated**] use for example twice a gray-value thresholding. First with a fixed parameter to exclude the background (the air surrounding the patient) and a second time with a varying threshold based on the distribution of gray values in the slice. Roughly two peaks mark the heart and the more solid surrounding tissue (which result in brighter values on the scan image) and a lower peak for the darker region - the threshold is then chosen between the two peaks. Gurcan et al. [**gurcan2002lung**] use k-means clustering (with $k = 2$) on the histogram to separate the two groups.

Juxtapleural nodules can pose a problem in this scenario since they lie closely connected to the membrane (pleura) that lines the lung and might be erroneously excluded. They produce cavaties on the initially segmented lung and need to be corrected. A rolling ball filter can be used to smooth the contures of the lung again and rightly add the juxtapleural nodules to the inner lung region [**armato1999computerized**]. Another approach, used by Gurcan et al. [**gurcan2002lung**] is comparing the distance between two points measured along the contour

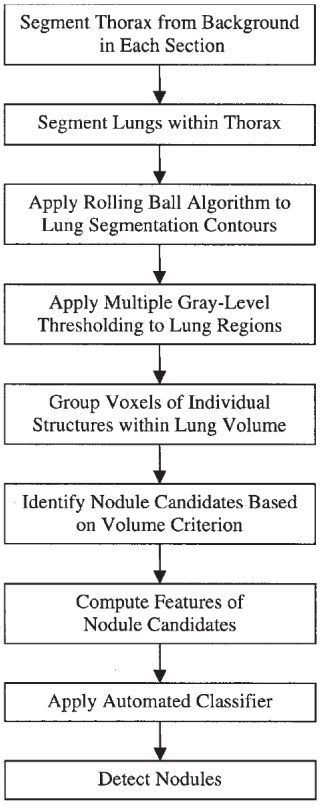

Figure 3.1: This image is taken from Armato's paper "Computerized Detection of Pulmonary Nodules on CT Scans" [**armato1999computerized**] and describes on one specific example how a traditional approach towards nodule detection is modeled.

that was formed by the initial segmentation and comparing it to the euclidean distance between the points. If the ratio is bigger than a preselected threshold the points are again connected with a line. The final segmentation in the end looks similar to Figure 3.2.

3.1.2 Candidate Selection

From the segmented lung region nodule candidates are selected. This selection can be done either by intensity- or model-based methods. The following text describes exemplary one method for each of the two types of algorithms and highlights concerns or limits of those. Armato et al. [**armato1999computerized**] use a multiple thresholding of the slices to obtain a set of 15 CT scans that only contain pixels above an increasing threshold. Now the 10 neighborhood of all on-pixels is used to group pixels together in structures, which are then classified by their volume. All structures that have a volume less than $14.1cm^3$ are nodule candidates and the others are disregarded.

Another approach involves using a predefined geometrical model to find the nodule candidates. Ye et al. [**ye2009shape**] use for example a shape index as defined in Equation 3.1 (this

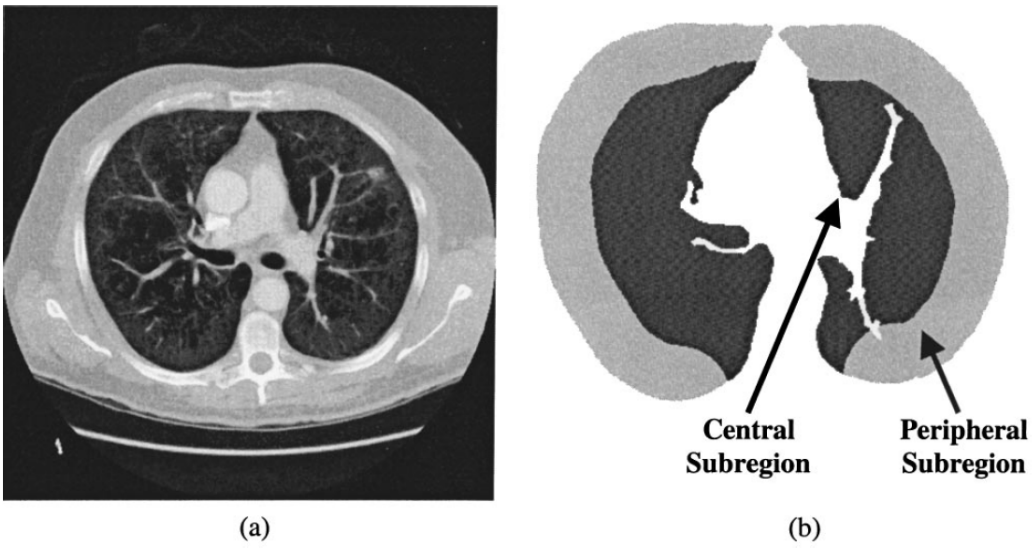

Figure 3.2: This image is taken from Gurcan's paper "Lung nodule detection on thoracic computed tomography images: Preliminary evaluation of a computer-aided diagnosis system" [**gurcan2002lung**] and shows the result of the segmentation process.

index is basically just a rescaled version of the original shape index that produces values between $-1, 1$ to $0, 1$) to classify candidates based on the shape of their surface. $k(p)$ are the values of the principal curvature in a point of interest p. Nodules obtain a higher score (closer to 1) compared to blood vessels, which have a prolonged shape.

$$SI(p) = \frac{1}{2} - \frac{1}{\pi} arctan \frac{k_1(p) + k_2(p)}{k_1(p) - k_2(p)} \tag{3.1}$$

Using sphericity as a strong nodule indicator may lead to a model that is highly sensitive to only one type of nodules while ignoring nodules that lie close to the chest wall and are not perfectly round or GGO nodules that are more diffuse in their morphology. Model-based algorithms on the other hand have the advantage of being able to incorporate a priori knowledge more easily. Different model parameters for example can be fine tuned due to the vessel density in a region of the lung - e.g. blood vessels are more common in the central region of the lung. The extracted nodule candidates can now be further processed in the classification step.

3.1.3 Classification

The candidates have to be classified to separate nodules between cancerous and non-malicious types. The classification itself can again be split into several steps, but this section will only highlight a few examples to give an overview. Armato et al. [**armato1999computerized**] and Gurcan et al. [**gurcan2002lung**] use a linear discriminant analysis classifier along several nodule features like volume, sphericity, radius of equivalent sphere and more to separate real nodules from other structures which have been found by the before described selection process. Firmino

et.al [**firmino2014computer**] provide a very rich comparison of different CAD methods and their performance, which can be seen in Figure 3.3.

Table 2 Performance comparison of lung nodule detection methods by sensitivity, FP, number of nodules, size and response time

Methods	Year	Sensitivity	FP	N° of nodules	Size	Response time	Type of nodules
Xu et al. [70]	1997	70%	1,7 per image	122	4 - 27mm	20s	NI
Armato et al. [48]	1999	70%	9,6 per case	187	3,1 - 27,8mm	NI	Solitary and juxtapleural
Lee et al. [45]	2001	72%	25,3 per case	98	< 10mm	187 min	NI
Suzuki et al. [71]	2003	80,3%	4,8 per case	121	4 - 27mm	1,4s	Juxtavascular, hilum, ground-glass opacity and juxtapleural
Murphy et al. [40]	2007	84%	8,2 per case	268	2 - 14mm	NI	Pleural and non-pleural
Ye et al. [23]	2009	90,2%	8,2 per case	220	2 - 20mm	2,5 min	Juxtavascular, isolated, ground-glass opacity and juxtapleural
Messay, Hardie and Rogers [21]	2010	82,66%	3 per case	143	3 - 30mm	2,3 min	Juxtavascular, solitary, ground-glass opacity and juxtapleural
Liu et al. [20]	2010	97%	4,3 per case	32	NI	NI	Solitary
Kumar et al. [75]	2011	86%	2,17 per case	538	NI	NI	NI
Tan et al. [76]	2011	87,5%	4 per case	574	3 - 30mm	NI	Isolated, juxtavascular, and juxtapleural
Hong, Li and Yang [22]	2012	89,47%	11,9 per case	44	NI	NI	Solitary
Cascio et al. [65]	2012	97%	6,1 per case	148	≥ 3mm	1,5 min	Internal and juxtapleural
Orozco et al. [63]	2012	96,15%	2 per case	50	NI	NI	NI
Teramoto and Fujita [24]	2013	80%	4,2 per case	103	5 - 20mm	30s	Juxtavascular, isolated, ground-glass opacity and juxtapleural

(NI = Not Informed).

Figure 3.3: This table is taken from Firmino et.al's paper "Computer-aided detection system for lung cancer in computed tomography scans: Review and future prospects" [**firmino2014computer**] and shows the performance of different algorithms. It is noticable how the number nodules used for training differ widely between the cases.

3.2 Deep Learning Approaches

A deep learning approach as defined for this thesis is any approach that utilizes in at least one of the above explained steps a neural network with more than 2 layers. All of the found papers use a mixed strategy: using more classical candidate selection strategys and using the neural network only in the final classification step. Only two other papers use 3D Convolutional Neural Networks: Anirudh et al. [**anirudh2016lung**] and Huang et al. [**huang2017lung**].

3.3 This Thesis

Where can this thesis be positioned in the field of Lung CT analysis and nodule detection? Given the sections above it is clear that it is part of the deep learning approaches, but differs in the sense that no prior candidate selection is performed on the data. The classification is based on raw ct scan patches that are fed to the network as is. The amount of data used for the training process also differs compared to the more classical papers. 2565 nodule and healthy patches have been used for the classification.

This means that no further features of the nodule can be determined like there malignancy score for example.

Chapter 4

Methods

The following sections highlight the used methods and give a rough overview about the used tools. First the used environment of software tools is described. Then the theoretical concept of a Convolutional Neural Network is explained. A more extensive explanation to the different software packages and computational resources can be found in the appendix A.

4.1 Software Packages

Writing a program for solving the task of nodule detection with neural networks makes the use of certain frameworks necessary. Using Python as a versatile programming language allowed for writing code for all aspects of the project: from data preprocessing and training the network to analyzing the results. The language narrows down the number of frameworks available for training neural networks. For this thesis Tensorflow (A.3) was chosen since it is at the moment the most active framework (in the sense of implementation Figure 4.1). Other frameworks like Cafe, Theano and Keras would have been valid choices as well. The Sun Grid Engine of the institute is used for automated execution on several machines (see A.2 for details) and was used for training the model.

4.2 Convolutional Neural Network

In this thesis a 3D Convolutional Neural Network (CNN) is used to classify the CT slices. The main motivation to use a 3D CNN in the case of nodule detection are the morphological features of the nodules that could not be fully utilized by a convolution that is only applied to 2D sections of the nodule. A CNN is similar to other artificial neural networks (ANNs) in the sense that it only uses forward connections, has an input and output layer and an arbitrary number of hidden layers in between. The hidden layers in a convolutional network are either convolutional or pooling layers which are in the end followed by one or more dense layers that perform the classification. Each of those layers is described in more detail in the following sections.

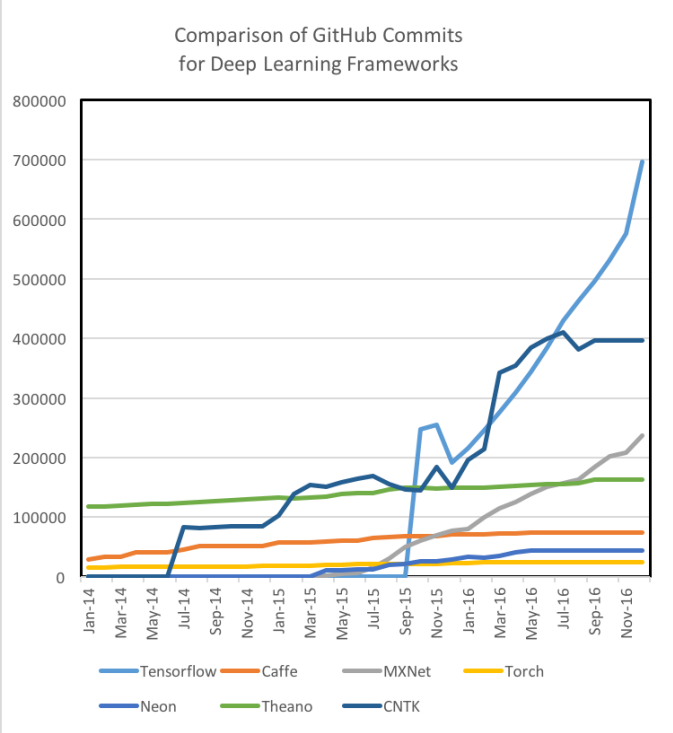

Figure 4.1: Number of commits for the different deep learning frameworks. This figure is taken from Shapiro [**shapiro2017**].

4.2.1 Convolutional Layer

A convolutional layer consists of $1..n$ kernels that are represented through shared weights which are compared to classic convolution in computer vision not *sliding* across the input but are duplicated across the image in a defined distance called stride. A stride of 2 would for example mean that the kernel is convolved with every second pixel of the image. The convolution is applied in 3D which means that the filter kernels are also 3 dimensional and the stride is as well defined in all 3 directions of the input image (x, y, z). A pixel in the output volume y can be derived from an image I with a 3D kernel h of size $(2m + 1, 2n + 1, 2p + 1)$ as described in equation (4.1).

$$y(x', y', z') = \sum_{i=-m}^{m} \sum_{j=-n}^{n} \sum_{k=-o}^{o} h(i + m, j + n, k + p) \cdot I(x + i, y + j, z + k) \tag{4.1}$$

As the layers of convolution stack the extracted features from the first layer are combined to more complex shapes.

The layers contain also a batch normalization step as described in [**ioffe2015batch**] that was implemented in the Tensorflow layers class. batch normalization is scaling the activation of the layer to become normally distributed in each dimension of the features, but has additional parameters that can be learned to shift and scale these values again. This should bring several

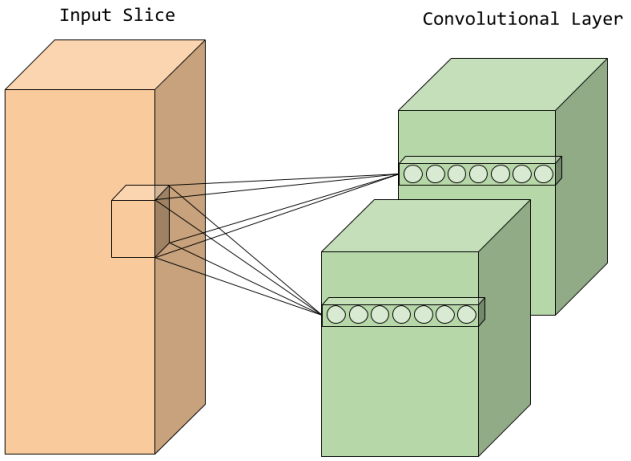

Figure 4.2: Structure of a convolutional layer. In image analysis the third dimension may encode the color informational, but in the example of Lung CT data it encodes additional spatial information. The figure shows two separate kernels that have shared weights for each kernel position in the input.

advantages like: reducing the dependence of the initialization method and allowing for higher learning rates. The batch normalization is followed by a max pooling layer which is applying a max filter to the normalized output. This adds additional small invariance to rotation and reduces the number of parameters in the network.

4.2.2 Dense Layers

The activation from the last convolutional layer is flattened into a $(1 \times n)$ vector and fed to the neurons of the fully connected layer. A neuron in this layer takes the weighted sum of the input and applies it's activation function to it. In this network neurons with a rectified linear activation function are used, whose output is defined for an input x and a number of weighted connections w as:

$$y = max(\sum_{w} w \cdot x, 0) \tag{4.2}$$

This function is one of the standard activation functions in deeper architectures that avoid the vanishing gradient problem. This problem occurs when in the back propagation step the adaption of the weights is dependent on the repeated derivative of the activation function. For networks with several layers this leads to weight stagnation in the layers close to the input.

Chapter 5

Model

With the building blocks defined in the previous chapter it is possible to construct a complete model for the task of lung nodule detection. This section describes in detail the used model and the learning process that was used to train it.

5.1 Network Architecture

The model is inspired by the network presented by Huang [**huang2017lung**]. It has 3 3d convolutional layers and 3 dense layers. The full architecture can be seen in Figure 5.1 and is in the following sections explained from the input to output.

5.1.1 Input

The input to the network are the patches that have been cut and stored from the complete lung scan as described in Chapter 2 with a shape of $(50x50x5)$. The patches are randomly augmented by flipping them in x and y plane (examples can be seen in 5.2). The augmentation is applied to make the learned classification more robust against distortions in the input and aiding in generalization. This makes sense in the specific scenario since the nodules are growing in different shapes and locations in the lung and flipping them is not producing an impossible input to the network. There is also an additional parameter that allows for scaling the input in the x, y plane.

5.1.2 Hidden Layers

The convolutional part has 3 convolutional layers with 40, 20 and 20 kernels each. The kernel size is $(3x3x3)$. This is in accordance with Huang et al.'s [**huang2017lung**] implementation. Each of them is followed by a batch normalization layer. Batch normalizaton is in TensorFlow implemented as described by Ioffe and Szegedy [**ioffe2015batch**]. The input of it is fed into a max-pooling layer with a pool size of $(2, 2, 2)$. The output of the last convolutional layer is flattened and fed to the dense layers. Two dense layers with 64 neurons each and a ReLU activation function are used in this model. Their output is finally combined in two neurons, forming a 2D output of the network.

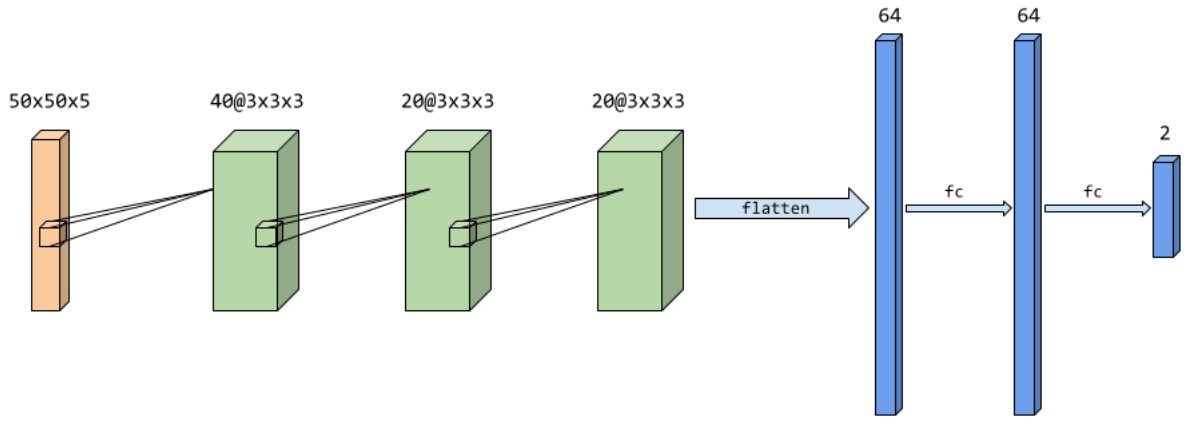

Figure 5.1: Architecture of the neural network. Each of the convolutional layers is composed of a 3D convolution layer with the respective filter size followed by a batch normalization and a pooling layer. The pool size is $(2, 2, 2)$ with a stride of $(1, 1, 1)$. The structure of the neural network resembles the one described by Huang [**huang2017lung**].

5.1.3 Output

The output of the network is the activation of the final two neurons. The class of the input is determined by the neuron with the higher activation in an one-hot design. Given the index of the maximum, is 0 encoding a healthy 1 encoding a nodule patch.

5.2 Training

During training the network is operated with batches of the input data. Regularization methods used for this network include batch normalization and dropout. Batch normalization is already described in Section 4.2.1. Dropout is another regularization method which during training drops random neurons of the network - training effectively several models at once, as discovered by Srivastava et al. [**srivastava2014dropout**], which should increase the overall performance of the network. During training no improved performance could be observed when applying dropout throughout the whole network. It was rather harmful if applied to the convolutional layers. Thus the final model uses dropout only in the fully connected layers. The loss of the network is then computed by the softmax cross entropy between the labels in their one-hot form and the output of the two neurons in the end of the network. The Adam Optimizer is used on this loss for adapting the network parameters. For each epoch the training is done on the complete training set with randomly permuted batches of size 10. The training time is set to 3 full days.

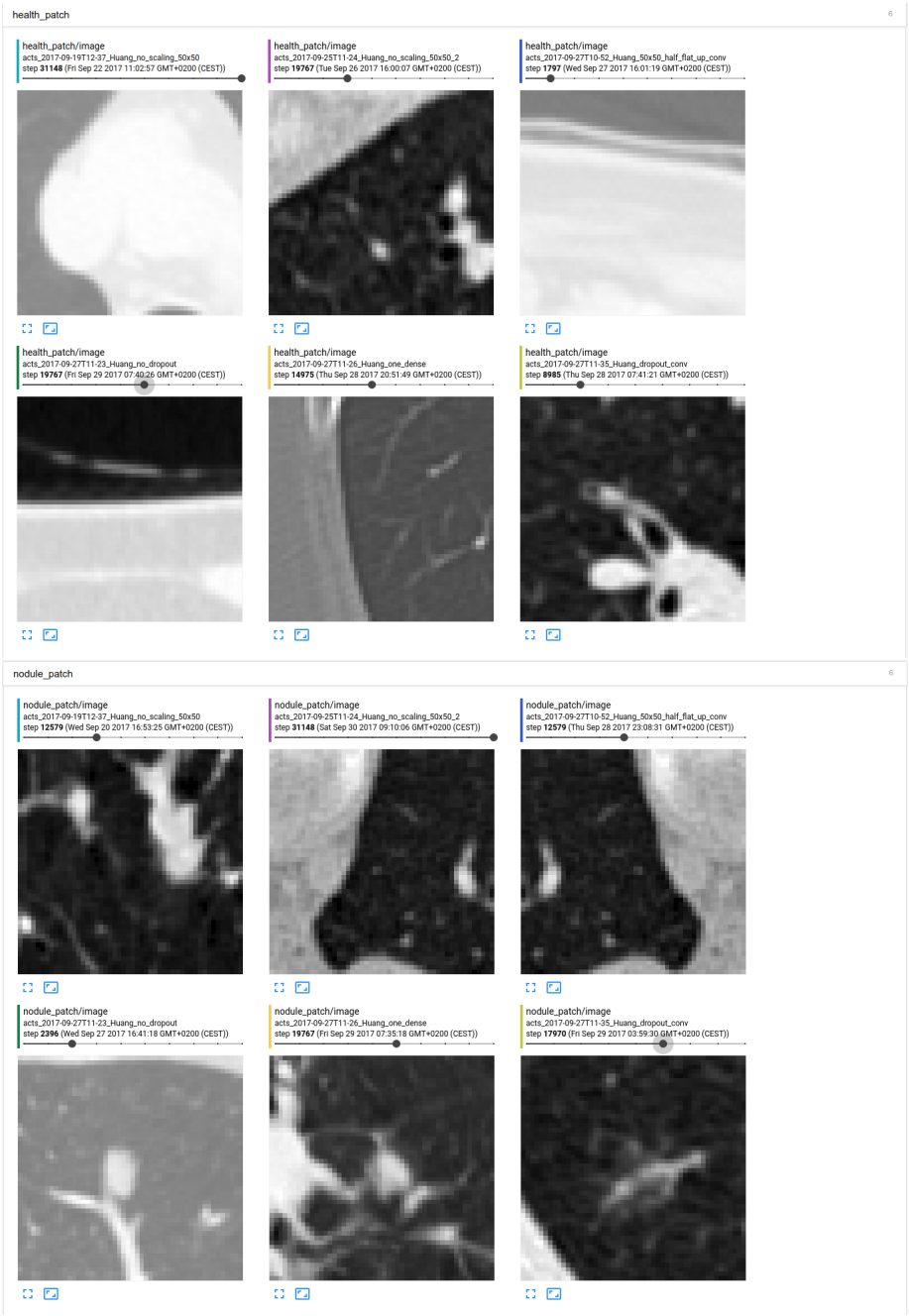

Figure 5.2: Input data for the cases of healthy and nodule patches. The image is taken from Tensorboard and shows in the case of nodules the random permutation of the input data.

Chapter 6

Results

This chapter presents the result of the training process. The final network performances are evaluated and compared to other papers. The network's extracted features are further analyzed and a way forward is sketched.

6.1 The trained Network

The model performs with a sensitivity of 81% on the validation set and a false positive rate of 0.192 per sample. This is already a better result than reported by Xu et al. [**xu1997development**], Armato et al. [**armato1999computerized**], Lee et al. [**lee2001automated**], Suzuki et al. [**suzuki2003mass**] and Teramoto et al. [**teramoto2013fast**].

This is of course not the only metric that needs to be compared in order to evaluate the approach. The way the samples have been produced in this thesis differ from the other papers

what about fp rate? what about applicability? what about time?

todo: explain why that might not be the most relevant perf number

6.2 Analyzing the Network

To understand how a network solves the task it makes sense to look at the patterns it's layers a sensitive to. The convolutional layers allow for visual inspection. Focus on the conv layers, what do they look like? Any hint on the geometry they are sensitive to? Activation patterns to synthetic data and patches from the patients.

How is that best understood? 2 Approaches: first mean activation of the filters in each layer per img.

6.2.1 Mean Class Activation

6.3 Bridge to other Approaches

How could a comparison at all be achieved? What is hindering the straightforward comparison of the kernel weights? Draw out a method to do that Show what has been done Compare

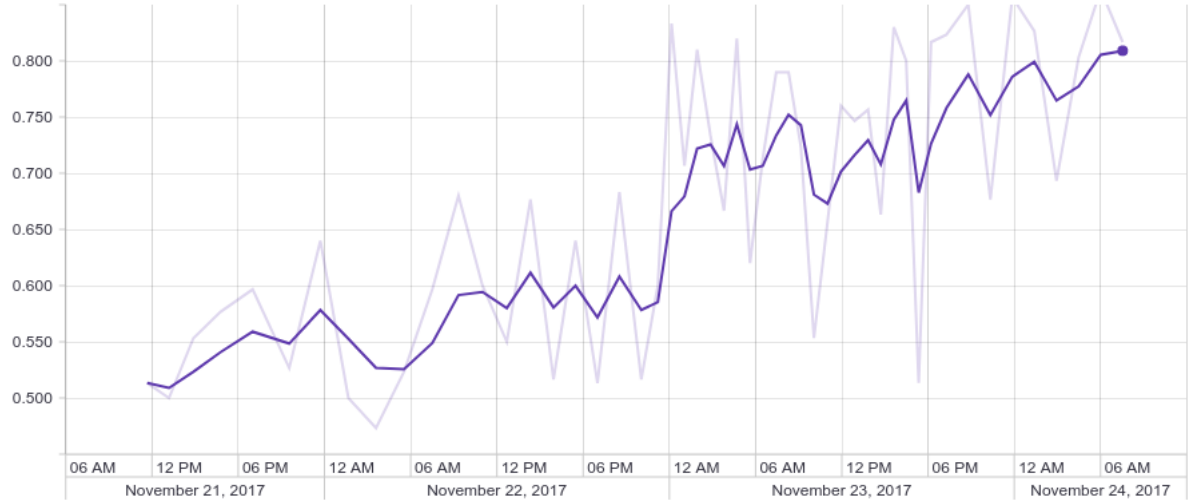

Figure 6.1: Learning process of the network - the graph shows the developement of the accuracy of the network on the validation set. The darker line represents the smoothed values of the lighter line. Since the training is performed on the CPU (GPU can not be utilized since the network was too big.), it takes several days.

performance to hand crafted approaches, take numbers out of papers

what could be similar features in the network?

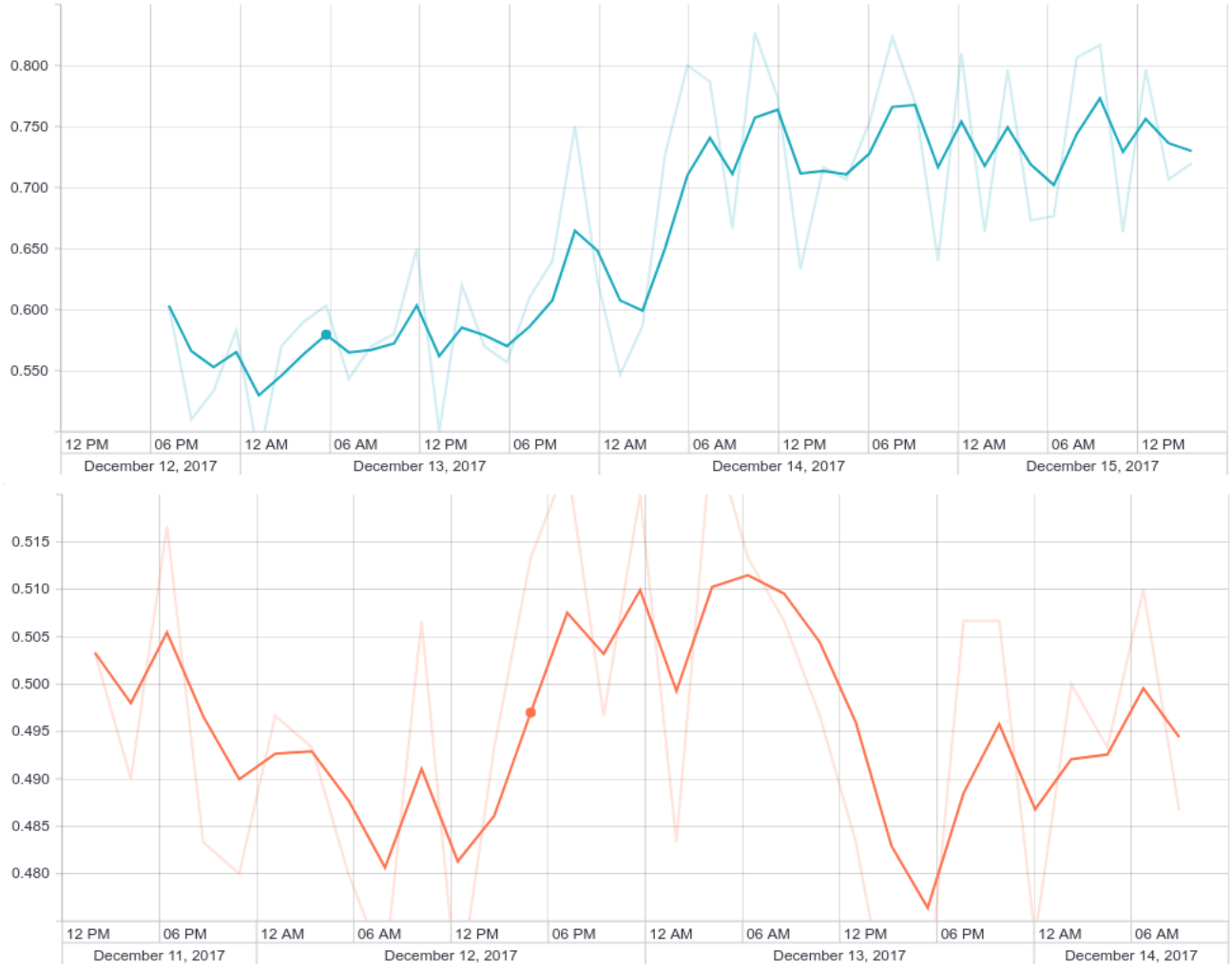

Figure 6.2: Another activation function in the dense layers (elu) and in the bottom graph with a batch size of 1.

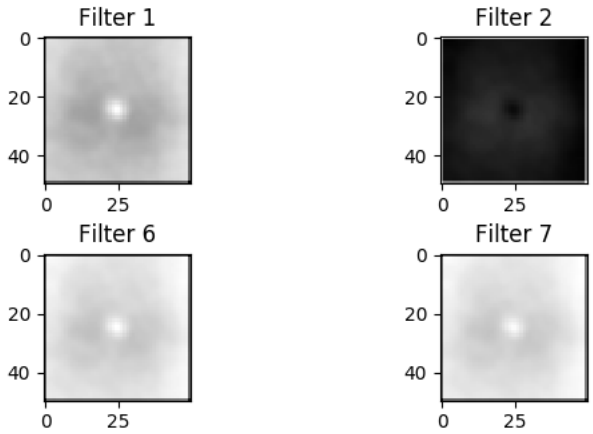

Figure 6.3: The mean activation from 500 nodule patches of the validation set. Already in the first layer the receptive field is focusing on the center area, where the nodule resides.

Chapter 7

Discussion

This thesis started out with two research questions. First- Can a 3D CNN be trained to perform the task of nodule detection and second, how can it's learned features be extracted and used for understanding the solution?

7.1 A 3DCNN Classifier

Answering the first research question: it was shown that it is possible to train a network with promising results upon which further optimization could be applied. The performance of the learned network was 81% which is surprising for it's simple structure. With more time and computational power it seems very possible to increase the performance further.

Ways forward include richer augmentation of the samples (rotating by different degrees, flipping the image in the z direction as well) or including the patches that have been labeled by the radiologists as "no nodule" as a separate class for training. Other network parameters could be systematically varied and tested for effectiveness, pushing the network performance even further. The same can be done for the training setup: prolonging the time the network has for training, varying the batch size or using a different optimizer for example. One could also think about more radical changes to the infrastructure. What would happen for example if the dense layers are replaced by further convolutional layers, making this a fully-convolutional network.

7.2 Understanding the Network

Answering the second research question proved to be more complicated. The filter kernels could be visualized, but it is hard to define a real measure of similarity to existing approaches and the extracted features did not encode any new or unknown properties of the nodules that could be easily translated into classical features.

Another interesting direction would be: "How minimal is the network allowed to be in order to perform with a certain accuracy?"

7.3 Outlook

The code for this thesis is completely openly available on "https://github.com/AndreaSuckro/acts". It has the necessary documentation available to reproduce the results of this thesis and rich documentation on the code. This allows for other interested researchers to further improve on the results and use the code or part of it in an own application. The network could be for example embedded into a complete application, that would take as an input a so far unknown complete CT scan and slide the network over the whole volume, marking in the process the regions that do potentially contain a nodule. An expert radiologist could use the results of the software to guide their own examination of the patient and check whether annotated nodules represent a real thread or are false alarms.

Appendix A

Software

A.1 Python

Python is a multi-purpose language, which means that no specific coding paradigma is imposed on the user, but one can use scripting as well as object oriented design alike. This allows for a very flexible style of programming which was used in this project for writing little tools that help with the data preprocessing as well as more complex code for the learning pipeline and the analysis of the network. Python is also widely used by the machine learning community which makes it easier to look up code examples and questions on forums like Stackoverflow. The specific version used in this thesis is 3.6 and all used packages are downloaded via pip or conda. The dependencies are listed in the file “act-env.yml" and can be installed with it as well.

A.2 Oracle Grid Engine

Our institute uses the work stations and additional hardware resources in form of a grid computing system that is managed by the Oracle grid engine (formerly known as Sun Grid Engine). The software manages the distribution of jobs to the nodes in the cluster, based on availability and resource requirements. The following bash script A.2 is used in the learning process of the network. It defines the name of the job and the necessary memory that should be available on the machines.

```
#!/bin/bash
#$ -N acts
#$ -l mem=128G
#$ -pe default 8
#$ -j y
#$ -v TESTNAME,WD,LOG_PATH

export LD_LIBRARY_PATH=$HOME/.local/cuda/lib64/:$LD_LIBRARY_PATH
export LIBRARY_PATH=$HOME/.local/cuda/lib64/:$LIBRARY_PATH
export CPATH=$HOME/.local/cuda/include:$CPATH
export PATH="/net/store/cv/projects/software/conda/bin:${PATH}"

. activate acts-cpu

python3 $WD/acts/src/learn.py \
        -d /net/store/cv/projects/datasets/image/pub/LIDC-IDRI/ \
        -l $LOG_PATH \
        -e 2000 \
        -s 1 \
        -b 5 \
        -n $LOG_PATH \
        -t $TESTNAME \
        -p 3000
```

Listing A.1: The code for calling the learning script. The parameters in the beginning are the information for the Oracle Grid Engine.

No machine with less memory is considered by the distributor as an execution host for this job. If the memory is set too low for the job, it can not complete the task and fails during execution since no more than the requested memory can be allocated dynamically during run time. It is also defined how many cores should be used on the host machine to run the job in parallel. The job can be directly executed via the command line or with a script (which makes more sense if one plans to run the grid job multiple times). The command used for this operation is *qsub*. Since the grid engine works with a concept of different queues it is possible to submit the job also just to specific queues, where one has the maximum execution time for example.

```
#!/bin/bash
export TESTNAME=Huang_no_scaling_50x50
export LOG_PATH=/net/store/cv/projects/tmp/asuckro/logs/acts_$(date +%Y-%m-%

mkdir -p $LOG_PATH

qsub -q all.q -o $LOG_PATH/grid.out runActs.sge
```

Listing A.2: The code for submitting the script to the scheduler.

All jobs are only allowed for a specified maximal amount of time depending on the users setting and the queue the job is transmitted to. All outputs to the console are logged in a file that can be specified with the '-o' variable.

A.3 Tensorflow

Tensorflow is a software library developed by Google Brain that aids the development of machine learning applications by expressing computations as a graph and taking care of the underlying optimization and execution. The version used in this thesis was 1.3.0. The approach of the framework is applicable to many computational tasks apart from neural networks as long as they can be formalized in a graph, but many of the higher level functions in the framework deal with neural networks. Tensorflow is usable via API's for Python, C++, Haskell, Java, Go, and Rust. Third party packages are available for C#, Julia, R, and Scala. Solving a problem with Tensorflow includes roughly two steps: first one needs to define a graph. A graph in Tensorflow is comprised of nodes, which are either variables (called Placeholders) or operations on those. The convolutional neural network in this thesis is defined like this: placeholders have been created for the lung patches that should be learned on, their respective labels and the phase of the learning (a boolean used for batchnormalization). Those are fed into basic transformation functions that handle the randomized flipping and rotating. The output of this function is fed into some summaries and further piped through the network. The covolutional layers have been scoped and encapsulated to own functions. The result of the dense layers is used to calculate the error that is used for the backpropagation learning in combination with the gradients. The complete structure can be seen in Figure B.1. The completed graph can now be executed in a Tensorflow session. In the session the placeholders are filled with data and a loop is used train the network for a specified number of epochs. Tensorflow only calculates the graph up to the point necessary for the queried parameter. So one can easily just ask for the prediction and skip the weight adaptation step as necessary for the analysis.

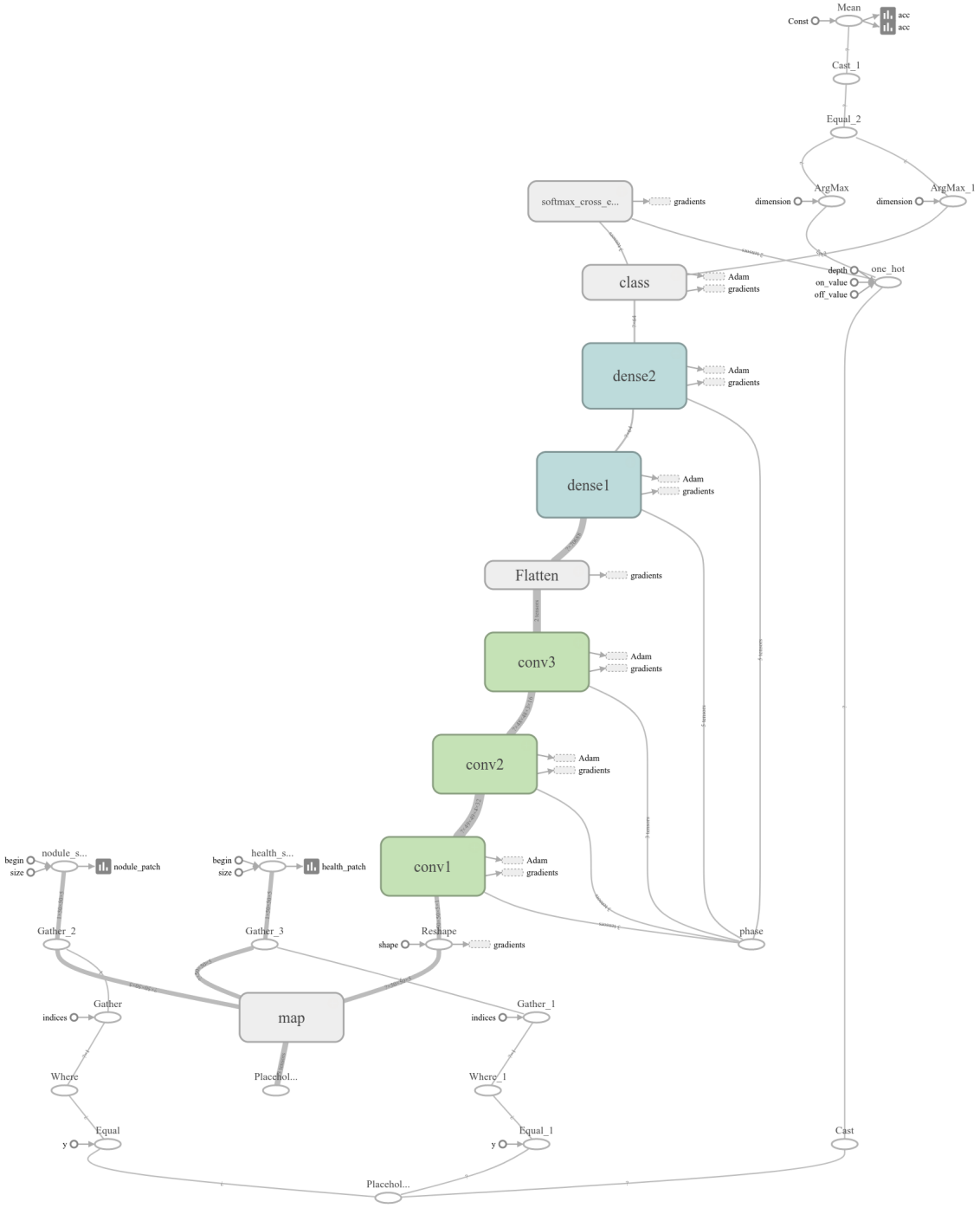

Figure A.1: The graph of the model that was generated for this thesis. Scoping of the different layers help a lot to encapsulate similar behavior. The Placeholders do not only flow through the network but are used to generate summaries and reports.

Appendix B

Data

B.1 LIDC-IDRI Dataset

The database as it is available now is the result of a long process that began in April 2000. The National Cancer Institute (NCI) - the U.S. federal government's principal agency for cancer research and training submitted RFAs to create guidelines of how such a combined reference database could look like. The Lung Image Database Consortium (LIDC) was formed by the Weill Cornell Medical College, University of California, Los Angeles, University of Chicago, University of Iowa and University of Michigan in 2001. There task was to develop a web-accessible resource for CT scans with attached meta information (like the slice thickness, tube current and other technical specifications as well as patient information) and nodule information based on expert knowledge. The initiative was further advanced in 2004 by the Foundation for the National Institutes of Health (FNIH) which founded the Image Database Resource Initiative (IDRI). They brought two additional medical centers (MD Anderson Cancer Center and Memorial Sloan-Kettering Cancer Center) and eight imaging companies (AGFA Healthcare, Carestream Health, Inc., Fuji Photo Film Co., GE Healthcare, iCAD, Inc., Philips Healthcare, Riverain Medical, and Siemens Medical Solutions) to the initiative. The new members contributed significantly to the whole database and since the process of data aquisition and annotation was streamlined to the previous recorded data the whole set is referred to as the LIDC-IDRI Database. It's aim is to further develop, improve and evaluate automated methods for lung cancer detection and diagnosis and it is comparable to other public datasets in the medical data community like the Digital Database for Screening Mammography (DDSM) which contains roughly 3000 mammograms - a pioneer in the field of public medical imaging datasets.

B.2 CT Scanner Technology

CT scanners cover a wide range of computed tomography devices, like positron emission tomography (PET) and single-photon emission computed tomography (SPECT), but most commonly refer to tube X-ray scanners. Those scanners work roughly like this: the object of interest (the patient) is placed in the center of the tube. A X-ray emitter rotates around the object and gives off radiation that pierces through the object and is reflected depending on the density of the

materials the object is composed of. Those differences in X-ray absorption make it possible to separate bones, nodules and blood vessels later on in the analysis. The recorded 2D images are combined by a software to a 3D representation, which makes it often necessary that the patients hold very still, even holding their breath during a scanning session.

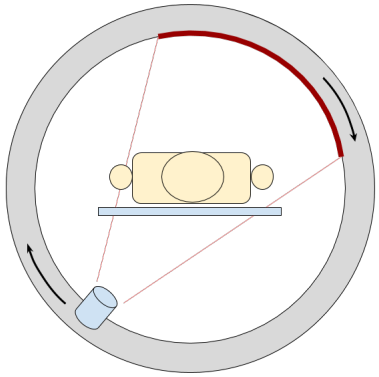

Figure B.1: This sketch illustrates the fundamental operation of a CT scanner. An emitter is rotating together with the sensor around the patient's body. The so recorded 2D images are used to reconstruct a 3D representation of the inner body.

B.3 CT Specifications

A range of scanner manufacturers and models was used to generate the data. The models and the number of samples they provided in the database are listed in the following table:

GE Medical Systems LightSpeed	Philips Brilliance	Siemens Definition, Emotion, Sensation	Toshiba Aquilion
670	74	205	69

Table B.1: The distribution of the 1018 samples among each CT scanner model. Vaules are taken from [**armato2011lung**]

.

The tube peak potential energies used for scan acquisition were as follows: 120 kVp , 130 kVp , 135 kVp and 140 kVp. Tube current ranged from 40 to 627 mA (mean: 222.1 mA).

The number of images per patient depend on the body size but also on the slice thickness which was 0.6 mm, 0.75 mm, 0.9 mm, 1.0 mm, 1.25 mm, 1.5 mm, 2.0 mm, 2.5 mm, 3.0 mm, 4.0 mm and 5.0 mm and the reconstruction interval that ranged from 0.45 to 5.0 mm (mean: 1.74 mm) [**armato2011lung**].

The number of pixels for each scan slice depends on the in plane resolution which ranged from 0.461 to 0.977 mm per pixel(mean: 0.688 mm). While the convolution kernels used for

image reconstruction differ among manufacturers, these convolution kernels may be classified broadly as “soft” (67) math formula, “standard/nonenhancing” (n=560), “slightly enhancing” (n=264), and “overenhancing” (n=127) (in order of increasing spatial frequencies accentuated by each class).

List of Figures

List of Algorithms

Declaration of Authorship

I hereby certify that the work presented here is, to the best of my knowledge and belief, original and the result of my own investigations, except as acknowledged, and has not been submitted, either in part or whole, for a degree at this or any other university.

Osnabrück, December 27, 2017