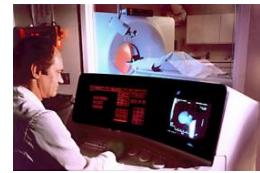


Final Report H03I7A “Design in Medical Technology”



Shape-based feature analysis for nodule detection in lung images

Ir. Kim Nuyts

Ir. Sven Van Hove

Academic year 2013-2014

Contents

Summary	2
1 Introduction	3
2 Literature review	4
2.1 Introduction	4
2.2 The biology of lung nodules	4
2.3 Overview of existing lung nodule detection systems	5
2.3.1 Commercial systems	5
2.3.2 Publications of automated lung detection systems	5
2.3.3 Performance of existing systems	8
3 Theoretical background	8
3.1 Features	8
3.1.1 Feature extraction	8
3.1.2 Feature selection	10
3.1.3 Features in nodule detection	10
3.2 Machine learning	11
3.2.1 Classifiers used in nodule detection systems	11
3.2.2 Random Forests, an ensemble classifier	11
3.2.3 Cascaded classifiers	12
3.3 Validation metrics	12
4 Methodology	13
4.1 Acquisition of the datasets	13
4.2 Preprocessing of the data	14
4.2.1 Processing of the annotations	14
4.2.2 Lung segmentation	15
4.3 Training of the classifier	16
4.3.1 Preparation of the training dataset	16
4.3.2 Feature extraction and selection	16
4.3.3 Cross validation	18
4.4 Testing and validation of the classifier	19
5 Results and discussion	20
5.1 Optimisation of the classifier parameters	20
5.2 Validation results	20
6 Conclusion	23
References	29
A Appendix	30
A.1 Gantt chart & logbook	30
A.2 Meetings	34

Summary

The goal of this project is to develop a fully automated algorithm for the detection of pulmonary nodules in CT scans. A cascaded Random Forests classifier will be used to assign nodule probabilities to individual voxels. To train the classifier, 30 datasets from the LIDC/IDRI database are used. Voxels in the center of the nodules are used as positives samples, while negative voxels are randomly sampled from any image part that is at a safe distance from all nodules. For each of these voxels, a feature vector is generated and fed into the classifier. Five-fold cross validation combined with an optimal parameter grid search shows accuracy increases to 98,2% after the last cascade level while at the same time it makes sure the classifier's hyperparameters are properly calibrated to prevent overfitting.

Once the classifier is trained, we use 8 different datasets from the LIDC/IDRI database to perform tests and validation. For performance reasons, a lung mask is calculated of each set before actual testing begins. We propose two lung segmentation algorithms, one based on thresholds and morphological operations and another based on region growing. Either of these will provide us with the initial mask to start the calculations on.

The cascaded classifier comprises of four levels. In the first level, intensity is used as a feature as it is readily available. This teaches the classifier that nodules belong to the soft tissue window, allowing it to discard any voxel outside of that range. The second level feature is a laplacian blob detector over several scales in addition to a distance map. This allows the classifier to specifically detect nodule-sized blobs in the 3D image while staying clear of the lung walls. At the third and fourth level a 3D averaging effect is used to distinguish between small nodules and other structures such as bronchioli and blood vessels by taking into account the structures extent in previous and following slices. Many other features were implemented and tested, but ultimately did not make the cut.

After the last cascade, roughly 0,5% of all voxels remain. We found on average 2,17 true positives and 4279,43 false positives per dataset, while maintaining the number of false negatives at zero. This yields a perfect sensitivity of 100% but a very low precision of 0,0634%. This number is so low because our algorithm is not strict enough yet and we have not implemented a proper false positives reduction method. On top of that, our number of false positives is expressed in number of voxelclusters (or "potential nodules") while the other two metrics are expressed in proper nodules. This discrepancy makes comparison between both metrics very difficult.

To improve these results, future work will have to continue fine-tuning the parameters of our algorithms and improve the effectiveness of the features. In particular the effectiveness of the 3D averaging feature was found lacking. Raising the cascade threshold after level two also looks promising. Other options include expanding the number of training sets and implementing new features. Furthermore, future research should look into more advanced voxel clustering strategies such that the remaining clusters are properly worthy of the term "potential nodule". We believe that this last step alone would significantly reduce our false positives and consequently increase the precision.

1 Introduction

In Belgium, one man in three and one woman in four face cancer before their 75th birthday [Kan14]. In 2010, 62.017 new cases of cancer were diagnosed according to [tK14]. Moreover, the World Health Organisation estimates the worldwide death toll from lung cancer will be 10.000.000 by 2030, which makes it one of the deadliest cancers [GKH⁺13, ZSB⁺07]. Fortunately, an early detection can increase the survival rate up to 70-80% [SJS⁺02] and will broaden the amount of treatment options [GMBW00]. Therefore, a wide range of studies have focused on this topic. The only way to detect lung cancer in an early stage is by examining a patient’s scan carefully. Currently, expert radiologists perform the investigation of the computed tomography (CT) scans. They use the shape, the texture, the location and the growth rate of the volume of the nodule as clinical parameters to determine the malignancy of the nodules and to decide on the diagnosis of lung cancer [WLB⁺10]. Nevertheless, the examination of these scans is a time-consuming task and is not free from errors. Therefore, there is an increasing demand for methods that provide assistance to the radiologist in performing this difficult task.

Fortunately, due to recent developments in CT technology it is possible nowadays to obtain near isotropic, submillimeter resolution images of the complete chest in a single breath hold. This high resolution has the advantage that it enables visualisation of small and low-contrast nodules that could hardly be screened in conventional programs. But although small nodules are in principle detectable in CT scans, a non-negligible fraction may be overlooked if they are situated in a maze of vessels of similar size [OO10]. An example of a nodule hidden in a maze of vessels is shown in Figure 1. The downside of these recent developments towards high resolution images is that enormous amounts of data are generated which increase the work load of radiologists. Especially since low-dose CT scans are more and more implemented in routine screenings. Still, this is no idle measure. Long nodules are very commonly detected on CT scans. Research shows that up to 51% of smokers

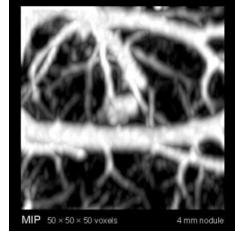


Figure 1: Nodule in maze of vessels of similar size [WRB⁺05].

aged 50 years or older have pulmonary lung nodules in CT scans [MAG⁺05]. Therefore, the United States Preventive Services Task Force stated that it “recommends annual screening for lung cancer with low-dose computed tomography in adults aged 55 to 80 years who have a 30 pack-year smoking history and currently smoke or have quit within the past 15 years” [Cin14]. Measures like this may enhance the early stage detection of lung cancers, but the workload of radiologists will be increased tremendously. No wonder the detection of pulmonary nodules in volumetric CT scans is one of the most studied computer-aided detection (CAD) applications [SSPvG06]. There is a need for a CAD system that can assist the radiologist in the detection of pulmonary nodules.

The goal of this project is to develop a fully automated algorithm for the detection of pulmonary nodules in CT scans. The algorithm will be established as a cascaded classifier which is a robust and fast machine learning algorithm. This algorithm will classify each voxel of the 3D images by assigning a nodule-probability to each of them. One of the challenges is the selection of the features that will be extracted from the image. The fact that some nodules are embedded in surrounding tissue (e.g. lung wall, blood vessel, etc.) and the fact that some nodules demonstrate very irregular shapes prevents us from using simple spherical template matching which might seem an obvious choice at first for pulmonary lung detection. In order to select the right classifier and list of features, a non-exhaustive literature review is performed. Commercial and non-commercial CAD systems for nodule detection and their performances are

discussed. Of course, in order to design an accurate and efficient CAD system, one needs to understand the biology of lung nodules so that topic is also briefly touched. Next, a short theoretical background on image processing, with a focus on feature extraction and selection, and machine learning is provided. Both topics will be related to findings in the literature. The statistical measures used for assessing the performance of classifiers are also briefly discussed as they might not be trivial in this project. Subsequently, the methodology that was established in this project is explained from the acquisition of the datasets and the preprocessing of the data to the training, testing and validating of the algorithm. The advantages and disadvantages of every step are discussed and the final results are presented. Finally, the main conclusions from the project are summarized and suggestions for future research are made. In the appendix the reader can find the reports of the weekly meetings with the supervisors, the Gantt chart and the logbook.

2 Literature review

2.1 Introduction

In order to establish a well thought pulmonary lung CAD system, the appearance of lung abnormalities is discussed and a delineation of the term 'nodule' is provided. Then an overview of commercial and research-based CAD systems is given and a comparison of performances is discussed.

2.2 The biology of lung nodules

Pulmonary nodules are lung tissue abnormalities that are roughly spherical with a diameter up to 30 mm. On chest CT scans they appear as a rounded or irregular opacity (Figure 2). Many types of lung nodules can be distinguished on CT scans. A centrilobular nodule is separated by several millimeters from the pleural surfaces, fissures and interlobular septa. Micronodules are very small nodules and have a diameter less than 3 millimeters. A ground-glass nodule

– or non-solid nodule – appears on the CT scans as a hazy attenuation in the lung. This type of nodule does not efface the bronchial and vascular margins. A solid nodule shows a homogeneous soft-tissue attenuation. Finally, a part-solid nodule exhibits both ground-glass and solid soft-tissue attenuation characteristics [HBM⁺08]. The types of nodules stated

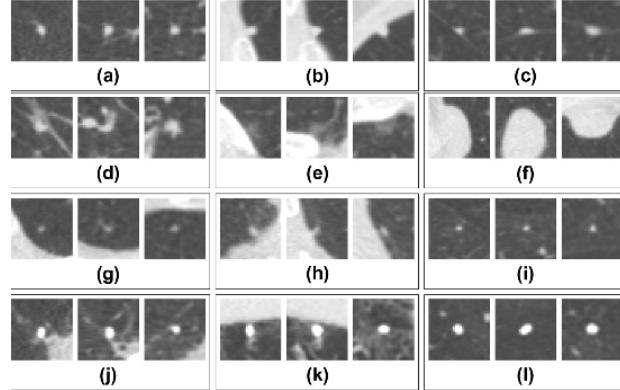


Figure 2: In every box a nodule is shown in a sagittal, coronal and axial view. The diameter is provided between brackets. (a) isolated nodule (4,4 mm); (b) pleural nodule (4,2 mm); (c) peri-fissural nodule (4,8 mm); (d) nodule with vascular attachments (5,9 mm); (e) ground-glass nodule (5,4 mm); (f) large pleural nodule (18,4 mm); (g) small nodule (3,2 mm); (h) small nodule (3,5 mm); (i) small nodule (2,3 mm); (jl) shows three examples of bright, calcified nodules (indication of benign nodules)

above can be categorised. Juxta-vascular pulmonary nodules have significant connections to their neighbouring vessels. Pleural tail nodules have only thin connections to the neighbouring pleural wall. Well-circumscribed nodules on the other hand do not have a connection to the neighbouring vessels and structures. Juxta-pleural nodules show some degree of attachment to their neighbouring pleural surface [KRYH03]. A number of nodule segmentation algorithms perform well in detecting specific types of nodules. However, these CAD systems show large limitations in detecting for example non-isolated nodules that are connected to the pulmonary wall [KATB13]. These algo-



Figure 3: In the three boxes an apical scarring, a pleural thickening and a nodular abnormality next to an emphysematous bulla are displayed in a sagittal, coronal and axial view. These may be perceived by a nodule detection algorithm as nodules, but in fact they are no nodules [vGAIdH⁺¹⁰].

rithms can be useful in particular situations, but if a detection algorithms really aims at being an asset for the radiologist, it should be able to detect all nodules while rejecting as much false positives (Figure 3) as possible.

2.3 Overview of existing lung nodule detection systems

As the demand for a reliable CAD system to detect pulmonary nodules is urgent, a lot of research has been dedicated to the matter. Several commercial systems have already been developed and many workstations that radiologists use to examine CT scans offer on-board nodule detection or enhancement capabilities [vGAIdH⁺¹⁰]. However, although a lot of efforts were made, the results shown in the various studies are rather diverse.

2.3.1 Commercial systems

In 2004 iCAD, Inc., provided lung cancer detection, analysis and tracking software for the TeraRecon's Aquarius product line. The latter licensed several software modules from iCAD. The iCAD QuickCueTM automatically detects cancerous lung nodules. The iCAD QuickMatchTM locates, compares and tracks nodules in previous or subsequent patient studies [Ter04]. However, the ImageChecker CTTM, launched by R2 Technology Inc., was the first CAD system approved by the US Food and Drug Administration (FDA)

for the detection of lung nodules during the examination of CT scans [AG08]. In 2005 R2 Technology introduced the second-generation ImageChecker CT Lung Version 2.0 CAD system which also implemented the AutoPoint temporal comparison algorithm. This CAD system “highlights abnormalities” and compares new and past images to demonstrate changes that have occurred over time [Dia05, Inc05]. In 2006 Vital Images, Inc., and R2 Technology announced the implementation of the R2 Technology's ImageChecker CT Lung CAD software into the Vitrea workstations [Ima06]. In the same year another company, Hologic, Inc., acquired R2 Technology and implemented their CAD technology [Hol06]. Then, in 2008, MeVis Medical Solutions AG, Inc., acquired the Pulmonary Computed Tomography Business from Hologic R2, Inc. [AG08].

Although the technology from R2 Technology is in high demand, some companies have developed their own software. In 2007 Medicsight plc, Inc., announced it was granted a medical device license from the Therapeutic Products Directorate of Health in Canada to introduce Medicsight LungCAD APITM [Hea07]. Another company, Median Technologies Inc., offers the LMS-Lung and LMS-Lung / CAD modules which provide quantification and detection functionalities for pulmonary (solid) nodules and micronodules [Tec14]. Siemens on the other hand has developed the syngo.CT Lung CADTM. They claim it is “a fully automated computer assisted second reader tool” that is designed to assist radiologists in the detection of solid pulmonary nodules [Sie05].

2.3.2 Publications of automated lung detection systems

Apart from these numerous commercial systems many academic research centers have tried to come up with a successful pulmonary nodule detection system. [LKH12] suggest that most CAD systems for the automated detection of lung nodules proceed according to a number of steps of which the first one is the acquisition of data. The detection of lung nodules is preferably performed on CT scans as they enable the visualisation of small volume and low-contrast nodules because of

2 LITERATURE REVIEW

the limited slice thickness. A large number of chest CT scans are available in public databases such as the LIDC/IDRI database. In the next step, the data are pre-processed to remove noise and artefacts which improves the quality of the images, but it is not necessary to do so. In the third step a segmentation of the lungs is performed. The lung lobe region is identified and the rest of the image is removed. This reduces the computational cost compared to the case where the whole image is processed and it increases the reliability and the accuracy of the algorithm. The next step is the nodule detection. “Lung nodule detection refers to the process of determining whether nodule patterns are present in the image and identifying the location of the nodules” [LKH12, p. 154]. Nodule detection methods can be categorized according to the detection method that is applied. The first group of publications uses a template matching technique to detect specific geometries. A second group relies entirely on the output of a lung nodule segmentation method. The third group applies a classification technique to classify voxels or regions of interest (ROI). In addition, a clustering method may be implemented to improve the performance of the system. The systems that include a classification component in their nodule detection algorithm have demonstrated better performances [LKH12]. In the final step of the process, the amount of false positives is reduced to achieve a maximum sensitivity.

The first problem that arises when processing CT scan in search for nodules is that one has to rely on the annotations made by expert radiologists. Accurate delineation of these lung abnormalities is crucial for optimal image analysis. The current approach to obtain delineations of lung nodules in CT scans involves one or more radiologists manually drawing the boundaries of the nodules. Often this manual segmentation overestimates the nodule volume to ensure the entire lesion is enclosed [RHS⁰⁵]. Furthermore, this process shows a high inter- and intrareader variability [CMT⁰⁷]. But the success of the extraction of image features depends on the accurate delineation of the nodules. Therefore, it is of utmost importance this delineation is done in an accurate and reproducible way. [GKH¹³] have improved the “Click and Grow” algorithm that has been developed by Definiens AG

and Merck and Co., Inc., which semi-automatically isolates tumors in CT images. The idea is that a radiologist detects the nodule and clicks on the region of interest in a 2D slice. This click initiates multiple seed points in a certain area. Then the application builds out the object three-dimensionally by region growing. An ensemble segmentation is obtained from the multiple regions that were grown. An evaluation on a set of 129 CT scans using a similarity index (SI) was performed. The average SI was above 93% which shows stability of the algorithm. The average SI for two different readers was 79,5%.

Apart from improving the manual nodule delineation of the radiologists, an image processing algorithm may also increase their detection rate by assisting as a “second reader”. [RPO¹⁰] assessed the diagnostic performance of radiologists – with their years of experience ranging between 9 and 24 years – and their temporal variation using incremental CAD assistance. 20 scans containing 190 non-calcified nodules with a magnitude of 3 mm and above were examined by three radiologists. After a free search, the radiologists independently evaluated a number of CAD detections per scan. The average sensitivity of their free search was about 53% (range, 44% - 59%) at 1,15 false positives (FP) per scan. This increased up to 69% (range, 59% - 82%) and a FP rate of 1,45 per scan when using the CAD assistance. The increase in sensitivity, with only a minimal increase in FP, was significant during a time period of 100 seconds. Then the increase in sensitivity flattened from 14% to only 2%. This evolution was due to the fact that the CAD nodules were presented to the radiologists in order of CAD score and was not due to a temporal change in the readers’ performance. It was also noticed in this study that different readers may experience a variable benefit from the use of CAD as some readers tend to often reject true positive CAD candidates. This reduces the potential benefit of CAD assistance. Nevertheless, [RPO¹⁰] states that CAD has the potential to equalise performance among readers by reducing individual detection errors.

[EBAEG¹³] proposed a three step algorithm to isolate lung nodules from spiral chest low-dose CT scans. In the first step the lung tissues were iso-

lated by applying an iterative Markov-Gibbs random field (MGRF) based segmentation framework. To retain the nodules attached to the pulmonary walls, the segmentation was refined by the iterative conditional mode relaxation that maximizes a MGRF energy. Then the lung nodules, arteries, veins, bronchi and bronchioles were separated from the rest of the tissues in the slice. In the second step the lung nodules (2-12 mm) were detected by applying 3D and 2D templates which describe typical geometry and greylevel distributions within nodules of the same type. Four template shapes were used: solid sphere, hollow sphere, solid circle and solid semicircle. The radius and the greyscale intensity of the templates were made adaptable. The detection combined the normalized cross-correlation template matching and a genetic optimization algorithm. The third step eliminated the FP using three textural and geometric features that were calculated for each detected nodule. To distinguish between FP and true positives (TP) Bayesian supervised classifier statistical characteristics from a training set (20 FP, 20 lung TP, 20 lung wall TP nodules) were selected from 50 separate subjects. CT scans from 200 subjects were tested in this study and a sensitivity of 82,3% and a FP rate of 9,2% were reached. The algorithm, that was implemented in C++ on an Intel Dual Core processor with 16 GB memory, took about 5 minutes to process 182 CT slices of size 512 x 512.

Other studies rely on a nodule segmentation method to detect lung abnormalities. [KDB⁺06] applies morphological opening, erosion, thresholding, seed optimisation and boundary refinement operations to extract large nodules. [IKI⁺07] proposes a segmentation of the lung areas using SNAKES method which is an active contour model. Abnormal shadow areas over the size of 5 mm are classified by using voxel densities. The algorithm was applied on 9 CT scans and a TP fraction of 0,8 and a false negative (FN) fraction of 0,2 were obtained.

A third category of nodule detection methods are the classification based methods. The main difference in output between a classification based detection method and a segmentation method is that the latter will provide the user with a delineation of the

entire nodule, while the former will give a nodule-probability per voxel. [OO10] tested four different learning based classification methods: a Neural Network (NN) classifier, a Support Vector Machine (SVM) classifier, a Naive Bayes classifier and a logistic regression classifier. First a number of ROI were extracted by applying thresholding and an 8-directional search in which candidate lung nodule voxels had to have neighbour voxels with densities between a minimum and maximum density threshold. From these ROI a number of features were extracted: straightness, thickness, vertical and horizontal widths, regularity and vertical and horizontal black pixel ratios. These features were then fed to the four classifiers. The NN classifier showed the best results, followed by the SVM classifier. [KATB13] also applied a SVM classifier. The lung area was first segmented by active contour modelling, which was followed by a set of masking techniques to transfer nodules from non-isolated into isolated ones. Based on a set of 2D and 3D features the SVM classifier was able to detect the lung nodules. Then the contours of these nodules were extracted by active contour modelling. In a last step the lung tissues in the original image were classified into four classes: lung wall, parenchyma, bronchioles and nodules. The results from this classification were used to distinguish solitary nodules from attached ones. When this algorithm was used to detect nodules in the ANODE09 dataset an average detection rate of 37,8% was obtained while the best performing method yielded a detection rate of 63,2%. The latter algorithm, ISI-CAD, was developed by [MvGS⁺09] and used the local image features – shape index and curvature – to detect candidate nodules. Two successive k-nearest-neighbour (K-NN) classifiers were applied to reduce the number of FP. This yielded a sensitivity of 80% with an average of 4,2 FP per scan. [SvWVvG03] also used k-NN for developing an algorithm which automatically distinguishes between normal and abnormal lung tissue. Before the k-NN was applied, a principal texture analysis was performed in this study to determine local features.

In addition to applying a classification based detection method, a clustering method can be implemented as well to improve the performance of the classifier. [KNO⁺00] developed a linear discriminant classifica-

3 THEORETICAL BACKGROUND

tion boosted by k-means clustering to distinguish between malignant and benign nodules based on topological histogram features. The k-means clustering divided the datasets of nodules in homogeneous classes to improve the performances of the linear discriminant classifier. [LKH10] also applied an ensemble classification aided by clustering (CAC) method. First, nodule and non-nodule parts of a training set were merged and then clustered into M clusters to take advantage of the similarity among features of nodule and non-nodule instances. Then each cluster was again divided into two groups, namely nodules and non-nodules. A set of multi-class classifiers was then trained. The classifiers used were a Random Forest (RF) classifier, a SVM classifier and a Decision Tree (DT) classifier. There were two different clustering methods applied: k-means and expectation maximisation (EM). The best results were obtained by the RF CAC EM algorithm. It yielded a classification accuracy of 97,72%, while SVM and DT CAC EMs recorded 96,46% and 93,98% respectively. However, the execution time of the SVM CAC EM algorithm was the lowest (182 seconds). These results were compared to non-CAC methods. A classification accuracy of 95,64%, a sensitivity of 95%, a specificity of 96,28% and a FP rate of 3,72% were obtained by non-CAC RF. This method recorded the lowest execution time (10 seconds).

2.3.3 Performance of existing systems

The algorithms presented in a wide range of papers report varying degrees of success in the automated detection of nodules. However, it is very difficult to compare studies against one another in a meaningful way due to differences in the size of the datasets, the evaluation methods, the data selection criteria and the characteristics of the nodules under examination [LKH10]. Especially comparing older and contemporary studies is difficult as older ones may have used scans with thicker sections (range 2.5 - 10 mm), on which small nodules are rather difficult to detect, than the scans nowadays (2,5 mm) [LKH10, vGAIIdH⁺¹⁰, MvGS⁺⁰⁹]. Some studies focus on nodules below or above a certain size or on special types of nodules

(e.g. solid nodules). [MvGS⁺⁰⁹] performed an extended literature review and found that the number of scans used for testing varied between 5 and 500 with a median number of 29,5. Many of the studies included multiple scans from individual patients, which means that the diversity of the available nodules was reduced. Furthermore, the results of publications are often presented in diverse ways.

In order to improve the access to data, and thereby the comparability between studies, the Lung Image Data Consortium (LIDC) created a publically available database which provides researchers with a vast amount of test- and trainingsdata. Nevertheless, as one can take different subdatasets from this large database, it is still difficult to compare results in an objective and meaningful way. Therefore, [vGAIIdH⁺¹⁰] created ANODE09, a database of 55 scans and a web-based framework which allows researchers to test their algorithms and to compare results against one another.

3 Theoretical background

3.1 Features

A 'feature' represents a characteristic of a voxel and its environment. Lists of features for all the voxels in an image are used to describe the entire image.

3.1.1 Feature extraction

Feature extraction involves obtaining a list of feature from a resource – an image or another dataset – to describe that dataset in an accurate way that allows it to be analysed by proper software.

In image processing a wide range of features can be extracted. Some features are very trivial: the grey value or colour of voxels, the position of voxels, etc. Features can be shape based: template matching, blob detection, edge detection, detection of lines and circles by means of a Hough transform, etc. Once

certain objects are detected in the image, their area, curvature and size can be determined. On the other hand, the shape based features need an a priori nodule segmentation. Performing such a segmentation is not trivial and error-prone. Therefore, the features in this project will be selected in a way there is no need for performing a nodule selection in advance.

We will now highlight some of the most relevant features for our purpose.

Distance Map A binary image containing either foreground (1) or background (0) voxels can be transformed into a distance map. In this process, every foreground voxel is assigned the distance to the nearest background voxel, while the background voxels are simply set to zero. In that regard, distance maps can be seen as generalisations of edges: they contain the same information, and more. An example of a distance map can be found in Figure 4.

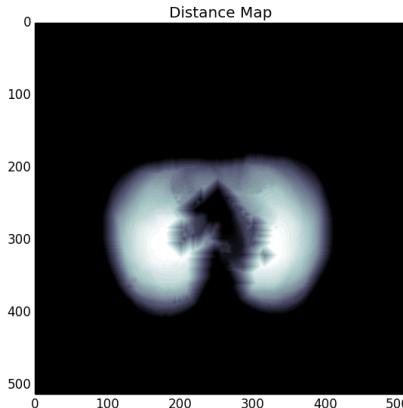


Figure 4: Example of a distance map

The distance metric must not necessarily be the Euclidean distance. Other metrics such as the Manhattan distance are also acceptable and easier to compute.

Laplacian One common feature in nodule detection is the laplacian – also called blob detector. The

laplacian operator applied to a continuous 3D function is defined as:

$$\nabla^2 f(x, y, z) = \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} \right) \quad (3.1)$$

To be applied to images, it must first be discretized into a 3D convolution mask. That mask typically has a large negative number in the center, surrounded by positive ones. As we are dealing with second derivatives, this operation is very sensitive to noise. One solution to this problem is to convolve the image with a gaussian kernel of scale t first. Because this kernel has a low-pass effect, noise will be reduced.

$$G_t(x, y, z) = \frac{1}{(\sqrt{2\pi t})^3} e^{-\frac{x^2+y^2+z^2}{2t^2}} \quad (3.2)$$

It can be represented by binomial filters – repeated convolutions of [1 1] with itself – in the discrete domain, making it rather cheap operation computation-wise. Convolution also has some interesting properties, which allow this calculation to be further optimized:

$$\nabla^2[G_t(x, y, z) * f(x, y, z)] = \nabla^2 G_t(x, y, z) * f(x, y, z) \quad (3.3)$$

The $\nabla^2 G_t(x, y, z)$ term is often referred to as the Laplacian-of-Gaussian (LoG) or Mexican hat filter. The LoG can be used to detect edges by finding its zero crossings, but we are more interested in its blob detection capabilities. It results in a strong positive response for dark blobs of uniform intensity with extent $\sqrt{2t}$ on a light background and vice versa for light blobs on dark background. The latter description fits most, if not all, nodules.

To get this strong response, we must carefully calibrate the scale parameter t to the nodule radius. However, because nodules come in different sizes, t will likewise have to range over different values. In general, larger t values yield lower filter responses. This makes finding minima and maxima in scale-space

3 THEORETICAL BACKGROUND

difficult. Using the scale-normalised laplacian operator solves this problem by multiplying the response with t . To make this computationally expensive operation more bearable, the laplacian is often approximated by the difference between two levels in a gaussian scale pyramid. The result is called the Difference-of-Gaussians (DoG).

Then there is one last technicality we must address. The above formulas assumed that the blobs were isotropic and thus that the LoG could be as well. However, because we perform most of our calculations in voxel space and voxels are not perfect cubes, this assumption does not hold. To compensate for this fact, we define a unique sigma per dimension that is a rescaled variant of the original sigma based on the corresponding voxel dimension.

Alternatively to the laplacian, one can also use the Hessian matrix of second partial derivatives as a feature. The laplacian is simply the sum of the elements on the Hessian's main diagonal. In that sense the latter is more complete, but also much more computationally expensive. For this reason, we stick with the laplacian operator.

3D Averaging This method [KATB13] is based on the fact that nodules are more or less spherical while bronchi and bronchioles are oblong. This might not be visible in a 2D image where both appear as circles depending on the orientation of the bronchioles, but it certainly is if a 3D window around a finding is taken into account. Nodules will repeat themselves in the preceeding and/or succeeding slices at the same place while the smaller bronchi and bronchioles will not. The comparison between slices was made by calculating the mean value of a window in the target slice and the average of the mean values of the same window in the succeeding and preceding slices:

$$M_{ij}^p = \frac{1}{9} \sum_{k,l=-1}^{+1} L^p(i+k, j+l) \quad (3.4)$$

$$M_{ij}^+ = \frac{1}{q} \sum_{p=p+1}^{p+q} M_{ij}^p \quad (3.5)$$

$$M_{ij}^- = \frac{1}{q} \sum_{p=p-q}^{p-1} M_{ij}^p \quad (3.6)$$

$$\text{3D averaging} = M_{ij}^- M_{ij}^+ \quad (3.7)$$

The index p indicates the z-index of the slice and the parameter q is the number of slices that are considered. The latter parameter is calculated as the ratio of the estimated nodule length to the thickness of the slices.

$$q = \frac{c}{T} \quad (3.8)$$

To differentiate between nodules and small bronchioles the parameter c was set at 5mm and 2,5mm. The window around each voxel was set at 3×3 . In case of a high q a high 3D averaging score will classify a voxel as belonging to a nodule whereas a low score will classify a voxel as a non-nodule.

3.1.2 Feature selection

A problem that may arise when analysing a dataset based on list of features, is that the amount of features is too large. Performing an analysis on large amounts of datasets requires a large amount of memory and computational power. Furthermore, the classification algorithm may overfit the training data and will not be able to generalise anymore. Therefore, a range of dimension reduction techniques can be applied to extract the uncorrelated and most important features of a list. A second option is determining the best features in an empirical way. A probability image can be used for a visual inspection.

3.1.3 Features in nodule detection

A non-exhaustive literature review revealed some commonly used features used in automatic nodule detection. [TKA13] primarily used morphological features: area, perimeter, diameter, solidity, eccentricity, aspect ratio, compactness, roundness, circularity and ellipticity. To select these features the minimum Redundancy Maximum Relevance (mRMR) method was applied. Selecting relevant features is import to improve the accuracy of the algorithm and to reduce

the processing time. [MvGS⁺09] used 3D local image features which were calculated per voxel: shape index and curvedness. [CZX⁺12] calculated the size, margins, contours and internal characteristics of the candidate nodules. [KATB13] used 2D stochastic features - grey level values and intensity values – as well as 3D anatomical features to remove the bronchioles from the list of candidate nodules. The features selected by [TF13] were area, surface area, volume, CT value, convergence, diameter and overlapping area. The algorithm of [OO10] implemented 3D features such as straightness, thickness, vertical and horizontal widths, regularity and vertical and horizontal black pixel ratios.

These features are calculated based on a prior segmentation of the nodules. On the other hand, instead of doing a nodule segmentation first, a lot of features can be calculated on the image itself based on grey values, intensities, grey values in the neighbourhood, etc. Although this is not a very common approach, these features can be generated without any preprocessing of the image.

3.2 Machine learning

The research field of machine learning is dedicated to the automatic learning of software in order to make accurate predictions based on past observations [Sha08]. This concept is of course very interesting when detecting nodules as a vast amount of 'past observations' is available. Classifiers are algorithms that classify given examples into a given set of categories [Sha08]. CAD systems which implement a classifier tend to outperform the CAD systems which do not [LKH10]. Therefore, the use of classifiers is very common in this field of research and many classifiers have been tested.

3.2.1 Classifiers used in nodule detection systems

[CNM06] compared the performance of a range of classifiers on eleven binary classification problems. The supervised learning algorithms that were used are SVM, the advantage that a better predictive performance is

NN, Naive Bayes, Memory-Based Learning, RF, Decision Trees, Bagged Trees, Boosted Trees and Boosted Stumps. Prior to calibration Bagged Trees, RF and NN perform the best on average across all test problems. After calibration Boosted Trees outperformed all other methods. The performances of SVM, Boosted Stumps and Naive Bayes were also dramatically improved by calibration. The performance of RF was not increased significantly. Overall, [CNM06] suggests calibrated boosted trees is the best learning algorithm. RF are close second, followed by uncalibrated bagged trees, calibrated SVM and uncalibrated NN. However, the training of Boosted Trees is inherently sequential which makes it slower to implement than RF. Another problem may be the noisiness of the data. Therefore, the choice and tuning of the parameters of the boosted trees (depth of trees, amount of trees, etc.) algorithm should be done carefully and this will take some time.

SVM are widely used in the development of nodule detection CAD systems [KATB13, LKH10, OO10]. According to [ASS13] SVM even outperforms RF. However, SVM has some disadvantages. First of all, the features need to be determined in advance and there is no such thing as a standard feature set (see also section 3.1.3). In the ideal case these features should all have the same dimensions and/or magnitudes. With SVM problems may arise with noisy data and images are very often quite noisy. The time complexity of SVM is $O(n^2)$ [sl14].

The best performing CAD system in the ANODE09 challenge was the ISI-CAD algorithm [vGAIdH⁺10]. ISI-CAD used a k-nearest neighbour classifier to reduce the amount of FP, but this method also has the disadvantage that the features should all be of the same magnitude in order to perform an optimal classification.

3.2.2 Random Forests, an ensemble classifier

Ensemble learners combine decisions of multiple classifiers to form an integrated output [LKH10]. The use of multiple learning algorithms at the same time has the advantage that a better predictive performance is

3 THEORETICAL BACKGROUND

obtained compared to the performance demonstrated by each individual learning algorithm separately.

Random Forests (RF) is a relatively new classification method which has not been exhaustively explored yet. “Random Forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalisation error for forests converges to a limit as the number of trees in the forest becomes large.” [Bre01, p.5] For this reason RF is an ensemble learning method. So given a test sample as the input, this input vector is put down each of the trees, each tree gives a classification and the forest selects the classification that has most votes. Consequently, the output of the RF depends on the combination of results from all individual trees. In this way a variance reduction is achieved and the output is made more robust against noise [Bre01, LKH10, Jie12]. RF has the advantage that the set of features does not need to be known in advance as the algorithm itself decides on itself which features to use. Therefore, a lot of features can be generated at will and the algorithm itself will decide on using them or not. The time complexity of RF is $O(n \log n)$ [Jie12] which makes it more suitable than e.g. SVM for large datasets.

Although the RF classifier might be a good choice for analysing large datasets, the required amount of memory and computational power can still be a problem. Consequently, an optimisation of the implementation of the RF classifier is desirable. A possible optimisation is the use of a cascaded classifier.

3.2.3 Cascaded classifiers

A cascaded classifier exists of several classifiers at different levels that are concatenated. Consequently, it is a special case of an ensemble classifier. The cascaded classifier uses all the information that is obtained in a previous level to provide the classifier in the next level with additional info on the data. On a lower level in a cascaded classifier the amount of features and the complexity of those features are lower.

For example, at the first level our classifier may determine a threshold to separate nodule voxels from non-nodule voxels based on their grey values. Using this trivial feature a large part of the voxels can already be eliminated as their grey value is too high or too low to be a nodule voxel. The features on the second level are more complex and therefore require more computational power. However, this is not a problem anymore as the amount of voxels to be processed was reduced after the first level. At the second level again a number of voxels are eliminated. The number of levels, and therefore the number of features, can be increased until the end results are satisfying. The reason for using a cascaded classifier is that both memory and CPU time are often limited. By discarding a certain amount of non-nodules at each level of the classifier, it is not necessary to calculate all features for all nodules which saves memory and CPU time.

The performance of the classifier depends on the features that are implemented on each level. For data mining the general rule is the more the better. If a lot of features are provided for each voxel the classifier can set more thresholds and is therefore better able to separate different classes. We started off from this concept, but very soon we had to deal with memory errors in Python. The code was further optimised, but still we had to cut in the amount of features that were used in the final version of the classifier as the amount of memory available remained a problem during the whole project.

3.3 Validation metrics

In order to formally evaluate the performance of a binary classifier, we introduce some statistical concepts. The reader should be familiar with Type I and Type II errors. A Type I error occurs when the model predicts something to be there while in reality it is not. In this text we call these occurrences false positives (FP). In our scenario, this corresponds with a classifier indicating that a nodule is present when there is really none.

Vice versa, a Type II error occurs when the model predicts something to be absent when in reality it

present. We call them false negatives (FN). False negatives in our scenario represent nodules not detected by the classifier.

On the other hand, true positives (TP) and true negatives (TN) represent the cases where the classifier detected the presence or absence of the nodule correctly.

	Nodule	Non-Nodule
Positive	TP	FP
Negative	FN	TN

Table 1: Summary of some basic statistical measures.

Table 1 summarizes these definitions, it is sometimes called the confusion matrix.

Because the terms above are in absolute numbers, they are difficult to compare across studies. That is where sensitivity and precision come in. Sensitivity compares the amount of true positives with the total amount of actual positives. Synonyms include the true positive rate or the recall rate. Precision also focuses on the true positives, but this time in comparison to the total predicted positives.

$$\text{sensitivity} = \frac{TP}{TP + FN} \quad (3.9)$$

$$\text{precision} = \frac{TP}{TP + FP} \quad (3.10)$$

Ideally, both measures should be 100% but that is an unrealistic expectation. There is an inherent trade-off between the two. For example when the algorithm is made more strict in order to detect less false positives, precision rises. At the same time the amount of false negatives likely increase as well, causing sensitivity to drop. In our case there is a clear preference for a higher sensitivity, even though it may cost us some precision.

One other important measure is the accuracy. It is the ratio of all correctly classified occurrences over all occurrences. Note that it is much less meaningful in

unbalanced classifications like ours where one class is far more prevalent than the other. Indeed, even a trivial classifier always returning the most common class can score very high on this. Alternatives for accuracy include the F_1 score and the log loss, but these are considered out of scope for the project.

$$\text{accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (3.11)$$

Ideally TP, FP, TN and FN should all be measured in the same units: either voxels or nodules. Pure voxel classification schemes can simply calculate how many voxels were correctly identified, while classification schemes working with nodules can use the number of nodules. However, our approach is a hybrid of the two. We do not require all voxels of a nodule to label the nodule as TP, just a couple should be enough. Hence, our TP and FN are expressed in number of nodules. This leaves us with a problem for FP and TN, as these cannot be expressed in an amount of nodules. We somewhat alleviate this problem by grouping clusters of voxels together, turning them into “potential nodules”.

4 Methodology

4.1 Acquisition of the datasets

The LIDC/IDRI database consists of 1018 thoracic CT scans that are obtained from a heterogeneous range of scanner models (seven GE Medical Systems LightSpeed scanner models, four Philips Brilliance scanner models, five Siemens Definition, Emotion, and Sensation scanner models and one Toshiba Aquilion scanner model). The database includes only one scan per patient so the scans are not correlated. The nodules in the scans were delineated by at least four different expert radiologists to identify as much nodules as possible. For this purpose the identification process was also subdivided into two phases: a blinded read phase and an unblinded read phase. During the initial blinded read phase each radiologist independently reviewed all scans and indicated the nodules in the

range of 3 to 30 mm and nodules smaller than 3 mm (if not clearly benign). In the subsequent unblinded read phase the anonymized blinded read results of all radiologists were revealed to each of the radiologists who then independently reviewed their marks along with the anonymous marks of their colleagues. The delineation of the nodules was done manually or in a semi-automated way. This was allowed as a study on this topic showed that the variation in nodule delineation done by different radiologists substantially exceeded the variation derived from different software tools [AIMB⁺11].

Fifty CT scans were obtained from the LIDC database¹. Together with the original DICOM images the associated XML files were obtained. These XML files provided a set of characteristics for each nodule found: region, subtlety, spiculation, internal structure, lobulation, shape (sphericity), solidity, margin, and likelihood of malignancy [AIMB⁺11]. From the 50 scans 12 scans were removed as they only contained 1-voxel nodules. The pixel size of the scans varied between 0,586 and 0,963 mm and the slice thickness varied between 1,25 or 2,50 mm. This means the diameter of the annotated 1-voxel nodules was less than 1 mm (micronodules). These nodules are difficult to detect for a radiologist, especially if they are hidden in a maze of vessels of the same magnitude. Therefore, some databases do not require the radiologist to mark such small findings. In the Nelson Trail database for example the radiologists do not have to mark nodules of which the volume calculated by the Siemens LungCARE workstation software is less than 15 mm^3 [MvGS⁺09]. This volume corresponds to a sphere with a diameter of about 3 mm, which is obviously more the 1 mm diameter from our 1-voxel nodules. In the LIDC/IDRI database there were no limitations set on the diameters of the nodules to be marked. However, the chance the radiologists marked every single micronodule is very small. The training of the CAD algorithm can therefore not be done properly as the annotations of the datasets are most probably incomplete for nodules with a diameter less than 1 mm. Therefore, the datasets containing only 1-voxel micronodules and the annotations of 1-voxel nodules in

the remaining datasets were removed. Furthermore, focussing on the detection of these micronodules limits the level of thresholding that can be performed during the classification of the datasets. Retaining these micronodules would prevent us from eliminating a substantial part of the false positive findings. So after eliminating these, 38 scans remained for training and testing. The RF algorithm was trained and validated on 30 and 8 CT scans respectively, consisting of 5168 and 1249 slices with a average of 172 and 156 slices per scan. The training set contained 64 nodules with each nodule containing 150 voxels on average. The minimum and maximum radii of the annotated nodules was 2 mm and 18 mm respectively. The test set contained 19 nodules.

4.2 Preprocessing of the data

The initial exploration of the data and the generation of a mask to perform a lung segmentation were done in MeVisLab 2.5.1 (VC11-64) (MeVis Medical Solutions AG, Bremen, Germany). Further processing of the data and the implementation of the RF algorithm were carried out in Python 2.7.6 (Python Software Foundation, Delaware, U.S.A). The training and testing of the RF based algorithm was performed on a computer with Intel Core 2 Duo CPU (2,27 GHz) and 3 GB of RAM.

4.2.1 Processing of the annotations

The annotations contained a list of readings by different radiologists. For convenience, we always worked with the first reading in the annotations. Each reading contains a list of nodules, defined using contour points per slice. These contour points were provided in voxel coordinates (x,y), while the slice index was provided in world coordinates (Z). By reconstructing the world matrix from the DICOM header, we converted the world Z-value back to its voxel z-coordinate counterpart. From these (x,y,z) coordinates the center of gravity plus the minimum and maximum radius of each nodule was calculated per slice. The 1-voxel nodules were ignored. In the rest of the algorithm

¹Freely available at <http://cancerimagingarchive.net>.

each nodule was represented by its center of gravity and either its minimum or maximum radius per slice.

We also considered modeling the nodules as perfect spheres in 3D, but this turned out to be more complicated than expected. The nodules can only be represented by spheres in world coordinates, as they instead form ellipsoids in voxel coordinates. However, we performed the bulk of our calculations in the latter frame of reference. Even when all slices are converted to a world frame of reference, the nodules were not always perfect spheres. Figure 5 illustrates this.

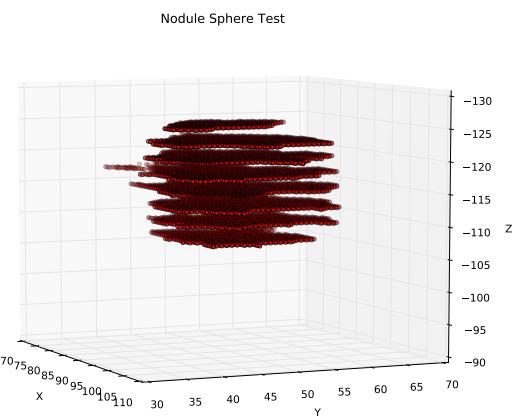


Figure 5: Flattened shape of nodule (LIDC scan 0001)

4.2.2 Lung segmentation

It was assumed that, if the whole 3D scan was fed to the cascaded classifier, only the soft tissue would remain after the first cascade. The grey value of each voxel was used as a feature on this first level. Unfortunately, this method did not prove to be efficient enough, so a second option was taken into consideration: a proper lung segmentation as a preprocessing step.

By performing a lung segmentation, the amount of voxels that have to be processed further on is significantly reduced by about 85%. Furthermore, it has the advantage that the soft tissues outside the lungs are eliminated so the accuracy of the nodule detection system is increased. Therefore, it is the first step that is performed in a lot of papers [KATB13, EBAAEG⁺13, TF13]. We started with implementing a lung segmentation algorithm based on [KATB13]. The first part of this algorithm consists of obtaining a binary lung CT image by adaptive fuzzy thresholding. In the binary image the lungs and the background are separated from the soft tissues of the body. Then two windows of different sizes are applied to close all the gaps in the mask and the initial lung mask is obtained by sweeping a rotated window over the entire binary image. This sweeping is necessary to transfer non-isolated nodules into isolated ones. Finally the mask is used to initiate an active contour model automatically for segmenting the lung area. As stated the first step of this algorithm was supposed to provide us with a variable, but accurate threshold to make the binary image. The performance of this first step was assessed by applying the algorithm on 42 slices – 28 slices with lungs and 14 without lungs – equally distributed over 7 scans. The results varied among the scans. In some cases the algorithm selected the appropriate threshold, in other cases the soft tissue around the lungs was not eliminated well. Furthermore, performing the lung segmentation as was described above would take a considerable amount of time (minutes). Instead, a fixed threshold of 1600 was empirically established to perform a body segmentation and to separate the soft body tissues from the rest of the image. This is permitted because CT scanners have a carefully calibrated output. Then gaps (lungs) in the body mask were closed by hole filling so a mask of the entire body was obtained. As this body segmentation already eliminated 55% of all voxels and no complex calculations had to be done to obtain the binary image, this result was found satisfying enough.

Despite the reduced amount of voxels, applying the algorithm still raised memory errors depending on the dimensions of the scan. In order to reduce the amount of voxels even further in the preprocessing

phase, a full lung segmentation was performed in a semiautomatic way in MeVisLab. After the scans were loaded in MeVisLab, the user manually indicated three points inside the lung area. Based on these points region growing is performed and a binary mask for the lung area is generated. The gaps in the binary mask – which represent nodules in the lung area and nodules hidden in the lung wall – are closed by dilation. We could get rid of the extra layer of voxels on the outside by performing an extra erosion operation, but we chose to keep it to make sure nodules attached to the wall are not lost. This mask is then exported to Python for further processing of the images.

An alternative way of performing a lung segmentation in Python would be calculating the body mask at the fixed threshold of 1600 and subtracting this mask from a similar mask in which the lung gaps are closed. In this way only the lung area is retained. Then a dilation and erosion should be performed as well to include the nodules hidden in the lung wall.

4.3 Training of the classifier

4.3.1 Preparation of the training dataset

Training a classifier requires us to supply it with examples of both positive and negative features. The positive features were extracted from voxels inside two thirds of the minimum radius of the nodule. Other voxels near the edge of the nodule were ignored due to possible ambiguity. These voxels might confuse the classifier. As the aim of this project was not to delineate entire nodules but assigning nodule probabilities to the voxels in the image, this reduction in order to provide clear training data for the classifier was justified.

Next, a second list of negative voxels was constructed. The amount of these voxels was taken the same as in the list of positive voxels to obtain a balanced training dataset. The positions of the negative voxels were selected at random over the entire image. The only constraint was that they were not allowed to be situated within two times the maximum radius of any

nodule. This constraint was again imposed to avoid ambiguity in training data.

Finally, features were calculated for both the positive and the negative voxels. The features that were used are discussed in section 4.3.2. This resulted in a list of features and a class (nodule or non-nodule) per voxel. This whole process was repeated for all scans and the results were then concatenated to feed into the ensemble classifier. We summarized this procedure using pseudocode in algorithm 1.

Algorithm 1: Training Phase

```
foreach dataset ∈ trainingsets do
    load DICOM files
    load XML annotations
    foreach nodule ∈ annotations do
        foreach slice ∈ nodule do
            | select positive voxels ( $d < 0.66R_{min}$ )
        end
    end
    while negative voxels < positive voxels do
        | select random voxel in volume
        | if not near nodules ( $d > 2R_{max}$ ) then
        |     | select negative voxel
        end
    end
end
for level from 1 to max do
    foreach selected voxel do
        | generate feature vectors up to level
    end
    train & cross-validate classifier
    save classifier model
end
```

4.3.2 Feature extraction and selection

Features implemented in final algorithm The **grey value** of each voxel is the only feature of the cascaded classifier at level one. From this, the classifier learns that nodules are always part of the soft tissue window. When asked to identify nodules with this information alone, the classifier will simply highlight all

soft tissue in the image. Remember that during the preprocessing phase we applied a lung mask. Hence all soft tissue outside of the lungs – save for a thin edge caused by dilation – is already discarded. By combining the two results, we are left only with thin lung edges, nodules and bronchi/bronchioli. The task of the next classifiers in the cascade will be to further distinguish between these different structures.

In level two, our focus goes to discarding the remaining lung edges and highlighting the nodules. We do this by combining a LoG **blob detector** with a **distance map**. As explained in section 3.1.1, the LoG operator is very good in detecting light blobs on a darker background such as nodules. The remaining lung edges in contrast do not fit this description at all. To find out which sigma would work best, we performed a couple of tests. First, we knew that larger sigmas would work better on larger nodules and vice versa. To that end, we searched for the smallest and largest nodule over all our datasets. Next, we empirically established the sigmas that yielded the best results for each. Because we do not know ahead of time how big the nodules we have to detect will be, we calculate the LoG for all intermediary values. The results of our test (see figure 6) showed that nodules radii lie between 2.68 mm and 18.39 mm with a mean of 6.32 mm and that we needed sigmas between 2mm and 10mm to cover this range.

To make the difference even more clear to the classifier, a distance map was calculated based on the original lung mask. Alternatively, one could also work with an ordinary edge detector, e.g. sobel. By blurring that output with a Gaussian, we get a similar distance-to-edge feature. However, we chose the distance map over this approach because it simply contained more information. Of course we can not blindly discard any voxels too close to the edges, because nodules can be closeby as well. That is why a distance map alone is not enough.

At the third and fourth level **3D averaging** was implemented to get rid of the bronchi and bronchioles that were still present. See section 3.1.1 for more information. With the help of all the features above, the classifier should have enough information to sep-

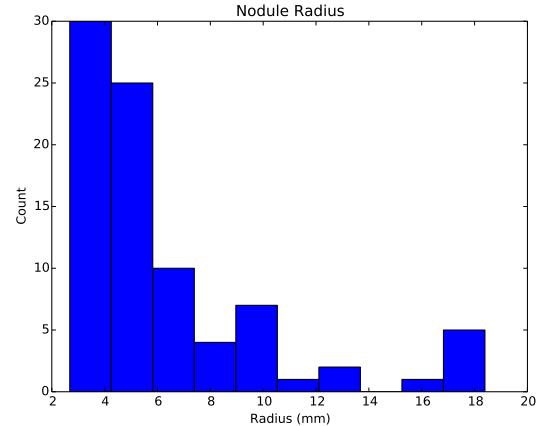


Figure 6: Histogram of nodule radii

arate the nodules from other structures in and around the lung.

Experimental features This paragraph provides an overview of other features that were tested but that were not implemented in the final classifier.

The **skewness and kurtosis** of a 3D window around each voxel was calculated as a measure for the texture of a certain part of the image. The skewness is a measure of the asymmetry of the probability distribution distribution of a real-valued random variable (grey value) about its mean. The kurtosis is a measure of the peakedness of the probability distribution of a real-valued random variable. However, the implementation of these first order statistics in the final classifier was prohibited by the time needed (hours) to calculate these features for every voxel in the image.

Another obvious feature, apart from the grey value of each voxel, is the **position** (x, y, z) of each voxel in the 3D image. An even better feature is the **relative position** $(\frac{x}{x_{max}}, \frac{y}{y_{max}}, \frac{z}{z_{max}})$ as it takes into account that the size of the datasets might change. These features could be useful when lung segmentation is not performed, because they would limit the search area. However, after testing it showed that these features would only be useful if a very large amount of

training data was used. Figure 7 shows what happens when only 10 datasets are used to train such a classifier. The training set should be large enough so that almost every possible nodule position inside the lungs was uniformly covered by the training data. This would not only be very inefficient, but interpatient differences in body and lung morphology would make it even more difficult. Hence these features were excluded from the feature set. Instead a distance map based on the lung mask was implemented as a feature.

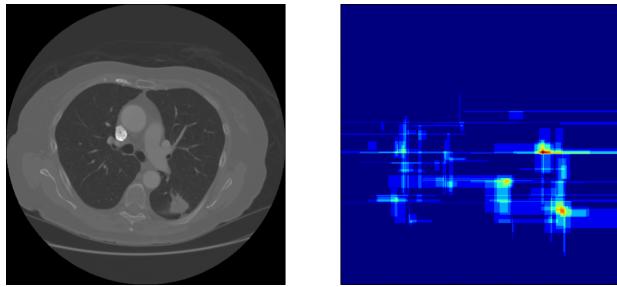


Figure 7: Probability output of a classifier trained on 10 datasets with only 1 feature: relative position. The output simply highlights the positions of nodules encountered in earlier datasets, without being relevant for the correct dataset.

A **3D Sobel filter** for edge detection was implemented as well. A Sobel filter is typically a 3×3 convolution mask that approximates the partial derivative of an image along an axis. The results of multiple sobel filters can be combined to calculate the gradient magnitude of each voxel. A high magnitude usually corresponds with an edge in the original image.

To be really useful in nodule detection, the distance from each voxel to neighbouring edges had to be calculated. However, due to a lack of time this was not feasible anymore. Instead, the distance map mentioned above was used as a substitute.

The **entropy** of an image is a measurement for the chaos of the image in a certain neighbourhood. Low entropy images are images where vast areas of voxels have the same grey values. Images with high entropy show a large contrast between neighbouring voxels. Therefore, it is a measure for the texture of an image.

The entropy of the entire 3D image was calculated and then the values for each voxel were extracted. However, the time demand for these calculations was far from negligible and as a consequence this feature was eliminated.

A feature **neighbours** calculated the sum, the subtraction, the multiplication and the quotient of the voxel in front and behind, left and right and above and below a target voxel. Each of these operations were also performed with the target voxel itself and the neighbouring voxel. Another substantial amount of features were based on calculations within **3D windows** that were swept over the lung mask of the image. The same operations as in neighbours were executed on a larger scale: the neighbouring voxels were determined by a certain adaptable window around the target voxel. Except for these basic operations the mean, the standard variance, the minimum and the maximum of the grey values of each window were calculated. Then these parameters and the grey value of the target voxel were used for performing the basic operations (sum, subtraction, multiplication, quotient) on. The same idea was applied on the rows and columns in each window. The mean of different rows and columns were determined and compared to each other by means of summing them up, subtracting them, dividing one by another and multiplying them. In these windows a frequency count was performed for each grey value in the window. These frequencies were again compared. The results that were obtained this way could be compared to one another by comparing the results for subwindows within larger windows or within the entire image. However, none of these features were implemented in the final classifier as calculating features for each individual voxel in an image took too much time (hours).

4.3.3 Cross validation

During the training phase, we already wanted to get an idea about the future performance of our classifier. Of course we could use the whole feature set in the training, and use the same features afterwards to check if our classifier is performing well. However, this is considered a form of cheating, and the test would

not tell us much about the predictive power of the classifier on new data. That is why it is important to reserve a fraction of the feature vectors for cross validation. A strategy called stratified K-fold is used to repeatedly split the feature set randomly in a train and a test fraction. The stratified adjective means that the proportion of nodules to non-nodules is similar in both fractions. Each time – also called *fold* – the classifier is trained with the train fraction and performance is checked with the test fraction. After a number of folds, these results are combined to give a proper estimate of the classifier performance in terms of accuracy (or other score metric).

This cross validation can also be used to determine the optimal hyperparameters of the classifier. By generating a grid of parameters and corresponding value ranges, all combinations can be tested. The one with the highest score is retained. In case of RF, it is important to configure it in such a way that overfitting does not occur. This can be done by increasing either the tree depth, the minimum number of samples per leaf or the minimum number of samples before a split occurs. The minimum number of samples per leaf was chosen to perform the optimisation with. A small number of samples per leaf corresponds to deeper trees and a more complex classifier, whereas a large number corresponds to shallow trees and less complexity.

4.4 Testing and validation of the classifier

From the previous section, we obtained a trained RF model that can be used to predict the class of new image features with a certain probability. We used this facility for two purposes: feature effectiveness testing and output validation. But before any of that could be done, features had to be calculated for all remaining voxels in the new dataset. A cascading approach allowed us to do this while keeping computational and memory cost acceptable. Additionally, a lung mask was applied to the whole volume, allowing us to ignore any outside voxels from the start.

To assess feature effectiveness, we ask ourselves the following question. Is a substantial amount of non-nodule voxels removed from the dataset without removing the nodule voxels as well, i.e. did the precision increase significantly without affecting the sensitivity? If this is the case the feature is kept at that level, otherwise another feature is implemented. The assessment is performed based on a probability image that is generated by combining the resulting probability output from all feature vectors. The first section of algorithm 2 summarizes this procedure.

Algorithm 2: Testing & Validation Phase

```

foreach dataset ∈ testsets do
    load DICOM files           // testing
    mask ← load lung mask
    for level from 1 to max do
        load classifier model
        foreach voxel ∈ mask do
            | generate feature vector
            | probability ← classify
        end
        combine into probability image
        mask ← (prob. image > threshold)
    end
    result = mask
    cluster remaining voxels   // validation
    load XML annotations
    foreach nodule ∈ annotations do
        if any cluster ∈ nodule ( $d < 1.5R_{max}$ )
        then
            | TP++           // nodule detected
            | delete clusters
        else
            | FN++           // nodule not detected
        end
    end
    foreach remaining cluster do
        | FP++           // spurious cluster
    end
    calculate statistics

```

When all features were decided upon, we validated the entire algorithm by calculating the aforementioned

statistics based on TP, FP and FN. This is shown in part two of algorithm 2.

5 Results and discussion

5.1 Optimisation of the classifier parameters

By performing a five fold cross validation during the training stage the most optimal parameter set for the RF algorithm were discovered. It was found that for all levels except the first, a value of 5 yields optimal results for the minimum samples per leaf parameter. For level 1, this optimal value was significantly higher at 55. This makes sense as higher values correspond to less complex classifiers.

The accuracy that was reached was respectively 80,5%, 97,9%, 98,1% and 98,2% for the four consecutive levels. This shows that the first two levels in the cascade contributed more in the process of removing negative voxels than the last two levels. The latter however were not useless as they still increased the level of accuracy. The fourth level for example still eliminated about 600 voxels per scan. Though, as mentioned before, the accuracy metric should be taken with a grain of salt in highly unbalanced classification tasks.

For each level of the classifier, a threshold was set to determine which voxels would be passed on to the next level (i.e. which voxels showed a higher nodule probability). In order to avoid discarding nodule voxels of small nodules a rather low threshold had to be set. These were empirically determined at 20%, 40%, 40% and 70% for the four levels respectively, although more rigorous testing might yield even better values.

5.2 Validation results

In this section, we will illustrate our results based on two slices from the fiftieth LIDC dataset. The volume dimensions are $512 \times 512 \times 139$ and it contains one nodule around slice 95. As figure 8 shows, slice

50 contains no nodules while slice 95 indeed contains exactly one. After applying the lung mask, 13,59% of all voxels remain.

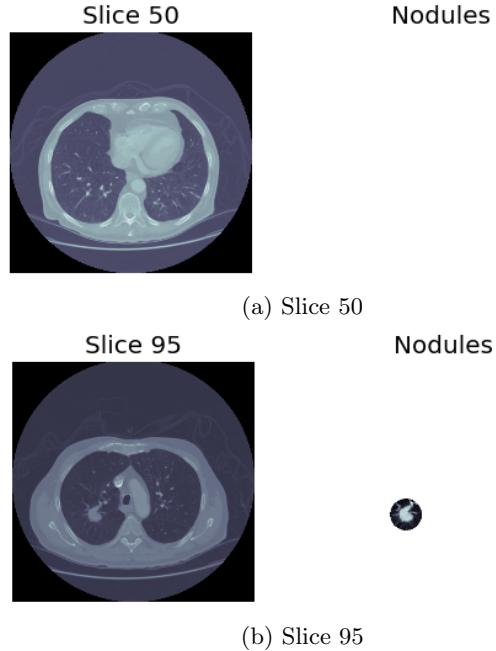
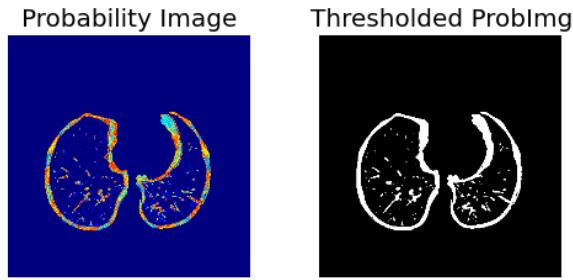


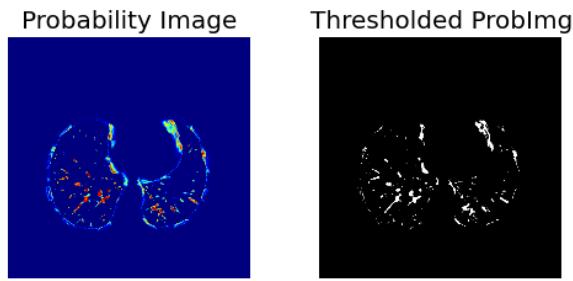
Figure 8: Example of two slices in dataset 50, one without and one with a nodule.

Level 1 As expected, we get a significant reduction in the number of voxels. More specifically only soft tissue structures – i.e. lung wall, bronchi/bronchioli and nodules – remain. The threshold performs a straightforward segmentation and cannot be improved much.

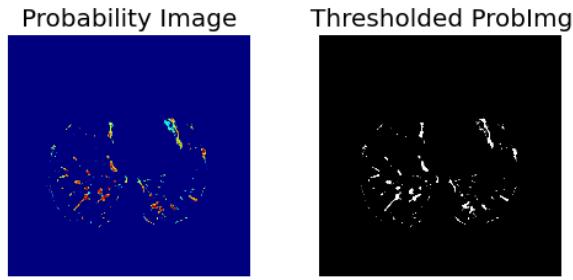
Level 2 This level was meant to highlight nodule-sized blobs while reducing the response of other structures, in particular lung walls. The laplacian filter certainly succeeded in the former, although we still get a medium-high response from certain wall segments at times. Unfortunately, the laplacian also highlights the smaller structures in the lung. This was inevitable as we needed LoG filters with small sigmas



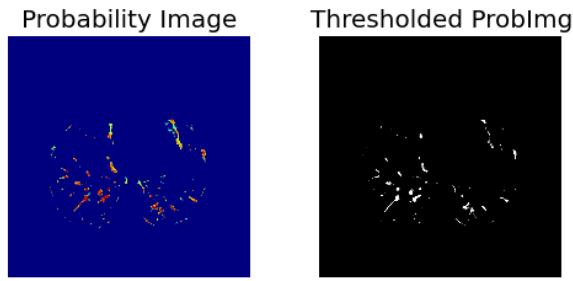
(a) Level 1 – Threshold: 20% – 5,74% remaining



(b) Level 2 – Threshold: 40% – 1,86% remaining

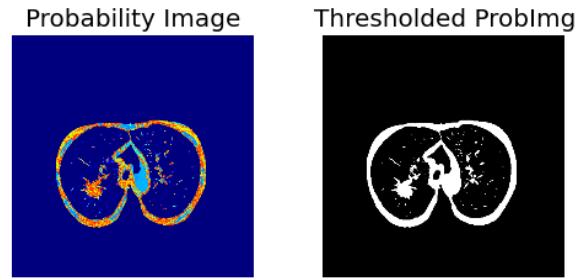


(c) Level 3 – Threshold: 40% – 1,55% remaining

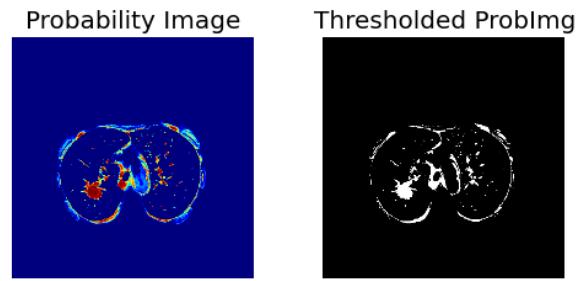


(d) Level 4 – Threshold: 70% – 0,54% remaining

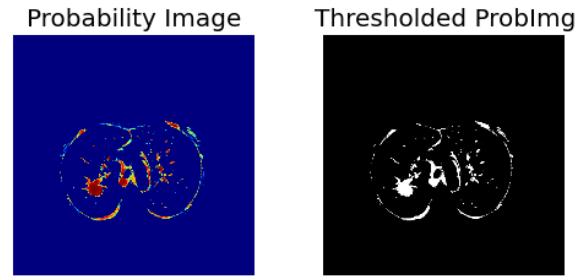
Figure 9: Processed versions of slice 50 in dataset 50. Left: probability image. Right: threshold of probability image showing the voxels that continue to the next level in the cascade. The algorithm started with 13,59% of all voxels remaining after lung segmentation.



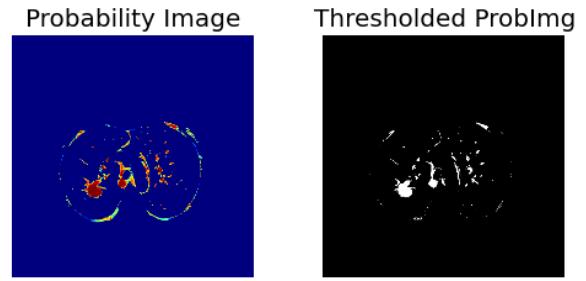
(a) Level 1



(b) Level 2



(c) Level 3



(d) Level 4

Figure 10: Processed versions of slice 95 in dataset 50. Left: probability image. Right: threshold of probability image showing the voxels that continue to the next level in the cascade. The same thresholds and remaining counts apply as in figure 9.

5 RESULTS AND DISCUSSION

to find small nodules. There is definitely room for improvement in threshold level. A higher threshold could discard more wall voxels while retaining nodules. In conclusion it is still a well performing feature that significantly reduces the voxel count, although it has some unfortunate side effects.

Level 3 and 4 The goal here was to discriminate between small bronchioli and blood vessels on the one hand, and small nodules on the other hand by means of 3D averaging. Unfortunately, this feature does not perform as well as hoped: there is only a margin decrease in non-nodule voxels. Note that the fourth level feature with the larger internal threshold works slightly better, but still not satisfactory. Future work should either fine-tune the parameters of this method or find a more effective method altogether.

The obtained sensitivity of the algorithm was 100,00% with an average of 2,17 TP and 4279,43 FP per scan. This high number of FP results in a very low precision of 0,0634%. Training the algorithm on 30 scans took 1 hour and 50 minutes. The processing of a new medium-large dataset (136 slices) takes about 10 minutes.

Keeping in mind that comparing different studies is difficult (see section 2.3.3), some relevant results from literature are presented here. [TF13] used a cylindrical nodule-enhancement filter and performed a FP reduction using a SVM classifier. This method reached a sensitivity of 80% and 4,2 FP per LIDC/IDRI scan. The detection speed ranged from 25 to 34 seconds per scan. [EBEAEG⁺13] applied template matching and found a sensitivity of 82,3% and a specificity of 9,2%. The time to process 1 scan in C++ was about 5 minutes. [LKH10] tested an ensemble classification aided by clustering (CAC) method on a set of nodules and non-nodule examples and they obtained a classification accuracy of 97,72%. An execution time of 190 seconds was registered.

These results indicate our algorithm performed well concerning detecting all nodules, but the amount of FP has to be reduced to get the precision up. At the moment, the algorithm is simply not strict enough:

it detects all nodules but at a high FP cost. This amount can be reduced in a first stage by determining the optimal value for all parameters, such as the threshold for each cascade level.

Our “potential nodule” clustering strategy also plays a significant role here. By clustering individual voxels together we hope to make them more comparable to nodules, but this is not always the case. Some clusters are simply too small to even represent a realistic nodule. Filtering out these clusters will help boost our results.

Another way of improving the algorithm is increasing the amount of training data. To determine the optimal amount of datasets a trade-off should be made between the time it takes to train the algorithm with extra datasets and the diminishing marginal improvements in the performances of that action.

A third possibility is implementing more features (e.g. Haar-like features), especially features focusing on removing edges and eliminating the bronchioles. On the other hand, implementing more features will increase the processing time. However, by optimising the thresholds in the existing algorithm these things can also be (partially) achieved. The threshold on level two can be set higher which will remove more edges. For eliminating the bronchioles the parameters of the 3D averaging features should be optimised. The results from the literature also show that we have a long processing time. However, one has to take into account that the implementation of this algorithm was done in Python – an interpreted language – which makes it inherently slower than low-level compiled languages such as C++. Nevertheless, Python was chosen for its rapid prototyping abilities. Future work may implement our algorithm in C++ or another compiled language to speed up the computational process.

[vGAIdH⁺10] compared the performances of six nodule detection CAD algorithms on the same validation dataset. The sensitivities at seven levels of false positive detection were calculated and then averaged. The best performing method in this study yielded an average sensitivity of 63,2% for the detection of all kinds of nodules. The sensitivity per nodule type was

also provided: small nodules (63,4%), large nodules (62,8%), isolated nodules (60,9%), vascular nodules (69,3%), pleural nodules (43,5%) and peri-fissural nodules (76,6%). This clearly shows that the ease of nodule detection also depends on the type of nodule. As this information is not available in the annotations of the LIDC scans and as we did not cooperate with a radiologist, it is not possible to differentiate between the different types of nodules in this project. However, as we may assume that different nodule types are represented in our testset, it is clear the algorithm is able to detect several types of nodules except for extremely small ones as we removed these from the annotations in the training and validation phase.

6 Conclusion

CT technology has come to the point where high resolution images of the whole chest can be obtained in a single breath hold. The resolution allows to detect pulmonary nodules in an early stage and therefore CT scans become more and more part of routine investigations. This causes a large increase in the workload of the radiologists, who are only human and thus also prone to errors. Time pressure and fatigue may lead to an increasing fraction of overlooked nodules. Research showed however that pulmonary lung detection systems, that serve as a second reader, can improve the performances of radiologists. Therefore, the goal of this project was to develop an accurate, fast and automated system to detect pulmonary nodules in CT scans.

A non-exhaustive literature review revealed that companies such as R2 Technology, Siemens, iCAD etc. already have invested in the development of similar software. However, a satisfying allround software package does not exist yet. The research is still ongoing to develop a system with 100% sensitivity and no false positives detection. One of the problems that arises here is that there is no golden standard to measure the performance of the CAD system against. Currently, the performances of CAD systems are compared with the findings of one or more radiologists. Although the fraction of overlooked nodules decreases

when more radiologists cooperate when analysing a CT scan, there is no guarantee that all nodules are delineated in a scan. This makes it very hard to assess the performance of a automated nodule detection system.

There are three main schools of thought in the development of pulmonary nodule CAD systems. The first group uses template matching to detect (a type of) nodules. A second group performs a nodule segmentation by means of a series of morphological operations, active contour modelling, etc. The third group applies classification methods, possibly aided by clustering. As there is evidence in literature that this method yields the best results, a more extensive literature review was performed to select a proper classification method. It was decided to use a cascaded Random Forest classifier as this type of classifiers is not yet fully explored in this area of research. This provided the opportunity to beat the state of the art in nodule detection algorithms.

Random Forests have many advantages. As an ensemble classifier it combines decisions of multiple classifiers to form an integrated output. This way of working has the advantage that a better predictive performance is obtained compared to the predictive performance demonstrated by each individual learning algorithm separately. Furthermore, RF is rather robust against noise compared with other classifier such as Support Vector Machines. Random Forests also allow to use a lot of features, even if they have different orders of magnitude, without increasing the time complexity too much. The features also do not have to be known in advance. At the same time, the method does not require a lot of parameter tuning. The only thing that has to be taken into consideration is the depth of the trees as overfitting must be avoided in order to maintain an algorithm that is able to generalise across datasets. The use of a cascaded classifier is preferred as it limits the amount of CPU time and memory storage.

The higher the level in the cascaded classifier, the more complex the features. On the first level the grey values of the voxels were used to detect soft tissue, to which class nodules belong. Although this first level is

able to eliminate the voxels outside the lungs, a lung segmentation was applied first to reduce the amount of voxels to be processed. The reason for this were recurrent memory errors when attempted otherwise. On the second level a blobdetector – Laplacian filter – and a distance map were implemented. On the third and fourth level a 3D averaging algorithm was elaborated with two different parametersettings. This 3D averaging allows to separate nodules from bronchioles and bronchi by taking into account the presence of these structures in the preceeding and/or succeeding slices. A lot of other features were implemented and tested, but most of them were removed again from the final classifier as they required a lot of processing time and did not perform accordingly. After each level in the classifier a threshold was set to determine which voxels were taken to the next level and which were to be discarded. These thresholds were empirically determined, but can still be optimised. The training and the validation of the algorithm were performed on 30 and 8 datasets respectively.

During the training of the algorithm a five-fold cross-validation was performed to determine the optimal set of parameters for the Random Forest algorithm. It was decided to take into account the optimal number of minimum samples per leaf to keep the algorithm from overfitting. An accuracy level of 98,2% was achieved in the last level of the cascaded classifier.

The validation of the optimised classifier showed 100% sensitivity, but also indicated the amount of false positives still has to be significantly reduced. The processing time of 10 minutes per scan is not extremely satisfying, but one has to take into account that Python is an interpreted language which makes it inherently slower than for instance C++. Transferring the code to a compiled language will speed up the process. The aim is to be able to process one scan within a few minutes. There is no need for faster processing as a radiologist will not need the results earlier, but it should not take much longer either.

The amount of false positives per scan (4279,43 FP per scan) has to be reduced. A first step to achieve this is the optimisation of the thresholds in the algorithm. A more accurate setting of the thresholds

will also be possible if the algorithm is trained on a larger amount of scans. On top of that, the clustering strategy might have to be revisited. Another possibility is the implementation of more features. However, there are two constraints here. The first one is trivial: the more features, the longer it will take to process the scan. A second consideration that has to be made is the fact that not all features are suitable for our approach. We only selected features that could be extracted from the image without performing any nodule segmentation in advance to make the algorithm faster an more robust. This nodule segmentation step would allow to use a wider range of features, but could also introduce errors in the algorithm. Furthermore, developing a nodule segmentation is not a trivial thing to do.

Although the use of the features and the code of the cascaded classifier were optimised considerably, one of the largest problems we encountered were the recurring out-of-memory errors. In the beginning of the project we decided to use Python 2.7.6 in the 32-bit version as it is typically more stable than the 64-bit version. This showed to be a bad choice later on as we had to spend a lot of time and efforts on the optimisation of the code concerning memory storage and computational power. This also prevented us from implementing a lot of features as we intended to do. The first concept was to calculate a range of features and let the Random Forest algorithm then decide on which and how many features to use in the final classifier. Because of the memory errors, we had to carefully select the features ourselves by visual assessment of the probability images that were generated at each level in the classifier, rather than letting RF figure out the most interesting features from a large pool.

Some other things to be done differently in the future is limiting the amount of time spent on reviewing the literature, limiting the amount of time spent on optimising the threshold values and performing the implementation of the features in a different way. In the beginning of the project we implemented a lot of features, which was also the aim before we encountered the memory errors, but a lot of them proved to be useless or required too much processing time. Instead, we should have tried some features, trained the algo-

rithm and make decisions on the type and amount of feautures based on these small experiments.

Besides the suggestions for improvement mentioned above, we have some other recommendations for future research. First of all, it would be interesting to not only detect the nodules, but also classify them as benign or malignant. In order to do this the grey value intensity gradient inside the nodule could be a helpful feature. In order to implement this feature, the nodule voxels should be clustered first. Also classifying the nodules accoring to the type of nodule (juxta-vascular; pleural tail; well-circumscribed; juxta-pleural) would be interesing as the type may already be an indication for the probability of malignancy of a nodule. As the type of nodule was not available in the annotations, it was not possible for us to implement this in the classifier. An interesting comparison that can be made is whether the subtlety that is assigned by the radiologist is comparable with the probability provided by the algorithm. And may the algorithm overlook the same nodules as one radiologist when assessing the CT scan?

Finally, if all optimaisations are performed the algorithm can be implemented in MeVisLab as a stand-alone modules which can easily be used by radiologists or researchers. To assess the performance of the optimised algorithm it can be validated on the dataset of the ANODE09 challenge where it can be compared with the results of other studies.

References

- [AG08] MeVis Medical Solutions AG. Mevis medical solutions complements product portfolio through strategic investment @ONLINE, April 2008.
- [AH13] Sultan Aljahdali and Syed Naimatullah Hussain. Comparative prediction performance with support vector machine and random forest classification techniques. *MACHINE LEARNING*, 69(11), 2013.
- [AIMB⁺11] Samuel G Armato III, Geoffrey McLennan, Luc Bidaut, Michael F McNitt-Gray, Charles R Meyer, Anthony P Reeves, Binsheng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Hoffman, et al. The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. *Medical physics*, 38(2):915–931, 2011.
- [AIMGR⁺07] Samuel G Armato III, Michael F McNitt-Gray, Anthony P Reeves, Charles R Meyer, Geoffrey McLennan, Denise R Aberle, Ella A Kazerooni, Heber MacMahon, Edwin JR van Beek, David Yankelevitz, et al. The lung image database consortium (lidc): an evaluation of radiologist variability in the identification of lung nodules on ct scans. *Academic radiology*, 14(11):1409–1421, 2007.
- [ASS13] Ashfaq A.K., Aljahdali S., and Hussain S.N. Comparative prediction performance with support vector machine and random forest classification techniques. *International Journal of Computer Applications*, 69(11):12–16, 2013.
- [Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [Cin14] Borgmeyer Cindy. Evidence laching to support or oppose low-dose ct screening for lung cancer, says aafp @ONLINE, April 2014.
- [CMT⁺07] Jay S Cooper, Suresh K Mukherji, Alicia Y Toledano, Clifford Beldon, Ilona M Schmalfuss, Robert Amdur, Scott Sailer, Laurie A Loevner, Phil Kousouboris, K Kian Ang, et al. An evaluation of the variability of tumor-shape definition derived by experienced observers from ct images of supraglottic carcinomas (acrin protocol 6658). *International Journal of Radiation Oncology* Biology* Physics*, 67(4):972–975, 2007.
- [CNM06] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168. ACM, 2006.
- [CZX⁺12] Hui Chen, Jing Zhang, Yan Xu, Budong Chen, and Kuan Zhang. Performance comparison of artificial neural network and logistic regression model for differentiating lung nodules on ct scans. *Expert Systems with Applications*, 39(13):11503–11509, 2012.
- [Dia05] DiagnosticImaging. New lung product propels cad into mainstream ct @ONLINE, October 2005.
- [EBEAEG⁺13] Ayman El-Baz, Ahmed Elnakib, Mohamed Abou El-Ghar, Georgy Gimel’farb, Robert Falk, and Aly Farag. Automatic detection of 2d and 3d lung nodules in chest spiral ct scans. *International journal of biomedical imaging*, 2013, 2013.

- [GKH⁺13] Yuhua Gu, Virendra Kumar, Lawrence O Hall, Dmitry B Goldgof, Ching-Yen Li, René Korn, Claus Bendtsen, Emmanuel Rios Velazquez, Andre Dekker, Hugo Aerts, et al. Automated delineation of lung tumors from ct images using a single click ensemble segmentation approach. *Pattern recognition*, 46(3):692–702, 2013.
- [GMBW00] Robert T Greenlee, Taylor Murray, Sherry Bolden, and Phyllis A Wingo. Cancer statistics, 2000. *CA: a cancer journal for clinicians*, 50(1):7–33, 2000.
- [HBM⁺08] David M Hansell, Alexander A Bankier, Heber MacMahon, Theresa C McLoud, Nestor L Muller, and Jacques Remy. Fleischner society: glossary of terms for thoracic imaging. *Radiology*, 246(3):697, 2008.
- [Hea07] HealthImaging. Regulatory news: Medicsight, rcadia@ONLINE, February 2007.
- [HHK⁺04] Claudia I Henschke, Eric A Hoffman, Ella A Kazerooni, Heber MacMahon, Anthony P Reeves, Barbara Y Croft, and Laurence P Clarke. Lung image database consortium: developing a resource for the medical imaging research community. *Radiology*, 232:739–748, 2004.
- [Hol06] Hologic. Hologic acquires r2 technology, inc. @ONLINE, July 2006.
- [IKI⁺07] Yoshinori Itai, Hyoungseop Kim, Seiji Ishikawa, Akiyoshi Yamamoto, and Katsumi Nakamura. A segmentation method of lung areas by using snakes and automatic detection of abnormal shadow on the areas. *International Journal of Innovative Computing, Information, and Control*, 3(2):277–284, 2007.
- [Ima06] Vital Images. Vital images and r2 technology launch fully integrated lung cad solution @ONLINE, April 2006.
- [Inc05] R2 Technology Inc. R2 technology launches its second-generation imagechecker @ONLINE, May 2005.
- [Jie12] Hu Jie. Course project: Parallelized random forests learning. University Lecture, 2012.
- [Kan14] Stichting Kankerregister. Kanker in cijfers: in belgi @ONLINE, April 2014.
- [KATB13] Mohsen Keshani, Zohreh Azimifar, Farshad Tajeripour, and Reza Boostani. Lung nodule segmentation and recognition using svm classifier and active contour modeling: A complete intelligent system. *Computers in biology and medicine*, 43(4):287–300, 2013.
- [KDB⁺06] J-M Kuhnigk, Volker Dicken, Lars Bornemann, Annemarie Bakai, Dag Wormanns, Stefan Krass, and H-O Peitgen. Morphological segmentation and partial volume analysis for volumetry of solid pulmonary lesions in thoracic ct scans. *Medical Imaging, IEEE Transactions on*, 25(4):417–434, 2006.
- [KNO⁺00] Yoshiki Kawata, Noboru Niki, Hironobu Ohmatsu, Masahiko Kusumoto, Ryutaro Kakimoto, Kensaku Mori, Hiroyuki Nishiyama, Kenji Eguchi, Masahiro Kaneko, and Noriyuki Moriyama. Hybrid classification approach of malignant and benign pulmonary nodules based on topological and histogram features. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2000*, pages 297–306. Springer, 2000.

REFERENCES

- [KRYH03] William J Kostis, Anthony P Reeves, David F Yankelevitz, and Claudia I Henschke. Three-dimensional segmentation and growth-rate estimation of small pulmonary nodules in helical ct images. *Medical Imaging, IEEE Transactions on*, 22(10):1259–1274, 2003.
- [LKH10] Shu Ling Alycia Lee, Abbas Z Kouzani, and Eric J Hu. Random forest based lung nodule classification aided by clustering. *Computerized medical imaging and graphics*, 34(7):535–542, 2010.
- [LKH12] S.L.A. Lee, A.Z. Kouzani, and E.J. Hu. Automated detection of lung nodules in computed tomography images: a review. *Machine Vision and Applications*, 23, 2012.
- [MAG⁺05] Heber MacMahon, John HM Austin, Gordon Gamsu, Christian J Herold, James R Jett, David P Naidich, Edward F Patz, and Stephen J Swensen. Guidelines for management of small pulmonary nodules detected on ct scans: a statement from the fleischner society. *Radiology-Radiological Society of North America*, 237(2):395–400, 2005.
- [MvGS⁺09] Keelin Murphy, Bram van Ginneken, Arnold MR Schilham, BJ De Hoop, HA Gietema, and Mathias Prokop. A large-scale evaluation of automatic pulmonary nodule detection in chest ct using local image features and k-nearest-neighbour classification. *Medical Image Analysis*, 13(5):757–770, 2009.
- [OO10] Serhat Ozekes and Onur Osman. Computerized lung nodule detection using 3d feature extraction and learning based algorithms. *Journal of medical systems*, 34(2):185–194, 2010.
- [RHS⁺05] Jan Rexilius, Horst K Hahn, Mathias Schlüter, Holger Bourquain, and Heinz-Otto Peitgen. Evaluation of accuracy in ms lesion volumetry using realistic lesion phantoms. *Academic radiology*, 12(1):17–24, 2005.
- [RPO⁺10] Justus E Roos, David Paik, David Olsen, Emily G Liu, Lawrence C Chow, Ann N Leung, Robert Mindelzun, Kingshuk R Choudhury, David P Naidich, Sandy Napel, et al. Computer-aided detection (cad) of lung nodules in ct scans: radiologist performance and reading time with incremental cad assistance. *European radiology*, 20(3):549–557, 2010.
- [Sha08] Rob Shapire. Machine learning algorithms for classification. University Lecture, 2008.
- [Sie05] Siemens. syngo.ct lung cad @ONLINE, October 2005.
- [SJS⁺02] Stephen J Swensen, James R Jett, Jeff A Sloan, David E Midthun, Thomas E Hartman, Anne-Marie Sykes, Gregory L Aughenbaugh, Frank E Zink, Shauna L Hillman, Gayle R Noetzel, et al. Screening for lung cancer with low-dose spiral computed tomography. *American journal of respiratory and critical care medicine*, 165(4):508–513, 2002.
- [sl14] scikit learn. Support vector machines@ONLINE, April 2014.
- [SSPvG06] Ingrid Sluimer, Arnold Schilham, Mathias Prokop, and Bram van Ginneken. Computer analysis of computed tomography scans of the lung: a survey. *Medical Imaging, IEEE Transactions on*, 25(4):385–405, 2006.
- [SvWVvG03] Ingrid C Sluimer, Paul F van Waes, Max A Viergever, and Bram van Ginneken. Computer-aided diagnosis in high resolution ct of the lungs. *Medical Physics*, 30(12):3081–3090, 2003.

- [Tec14] Median Technologies. Lms outside the us @ONLINE, April 2014.
- [Ter04] TeraRecon. icad provides lung cancer detection, analysis and tracking software for terarecon aquarius 3d visualisation and imaging product @ONLINE, July 2004.
- [TF13] Atsushi Teramoto and Hiroshi Fujita. Fast lung nodule detection in chest ct images using cylindrical nodule-enhancement filter. *International journal of computer assisted radiology and surgery*, 8(2):193–205, 2013.
- [tK14] Vlaamse Liga tegen Kanker. Cijfers @ONLINE, April 2014.
- [TKA13] Ahmet Tartar, Niyazi Kilic, and Aydin Akan. Classification of pulmonary nodules by using hybrid features. *Computational and mathematical methods in medicine*, 2013, 2013.
- [vGAIIdH⁺10] Bram van Ginneken, Samuel G Armato III, Bartjan de Hoop, Saskia van Amelsvoort-van de Vorst, Thomas Duindam, Meindert Niemeijer, Keelin Murphy, Arnold Schilham, Alessandra Retico, Maria Evelina Fantacci, et al. Comparing and combining algorithms for computer-aided detection of pulmonary nodules in computed tomography scans: the anode09 study. *Medical image analysis*, 14(6):707–722, 2010.
- [WLB⁺10] Dijia Wu, Le Lu, Jinbo Bi, Yoshihisa Shinagawa, Kim Boyer, Arun Krishnan, and Marcos Salganicoff. Stratified learning of local anatomical context for lung nodules in ct images. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2791–2798. IEEE, 2010.
- [WRB⁺05] R Wiemker, P Rogalla, T Blaffert, D Sifri, O Hay, E Shah, R Truyen, and T Fleiter. Aspects of computer-aided detection (cad) and volumetry of pulmonary nodules using multislice ct. *The British journal of radiology*, 78:S46, 2005.
- [ZSB⁺07] Yuanjie Zheng, Karl Steiner, Thomas Bauer, Jingyi Yu, Dinggang Shen, and Chandra Kambhamettu. Lung nodule growth analysis from 3d ct data with a coupled segmentation and registration framework. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

A Appendix

A.1 Gantt chart & logbook

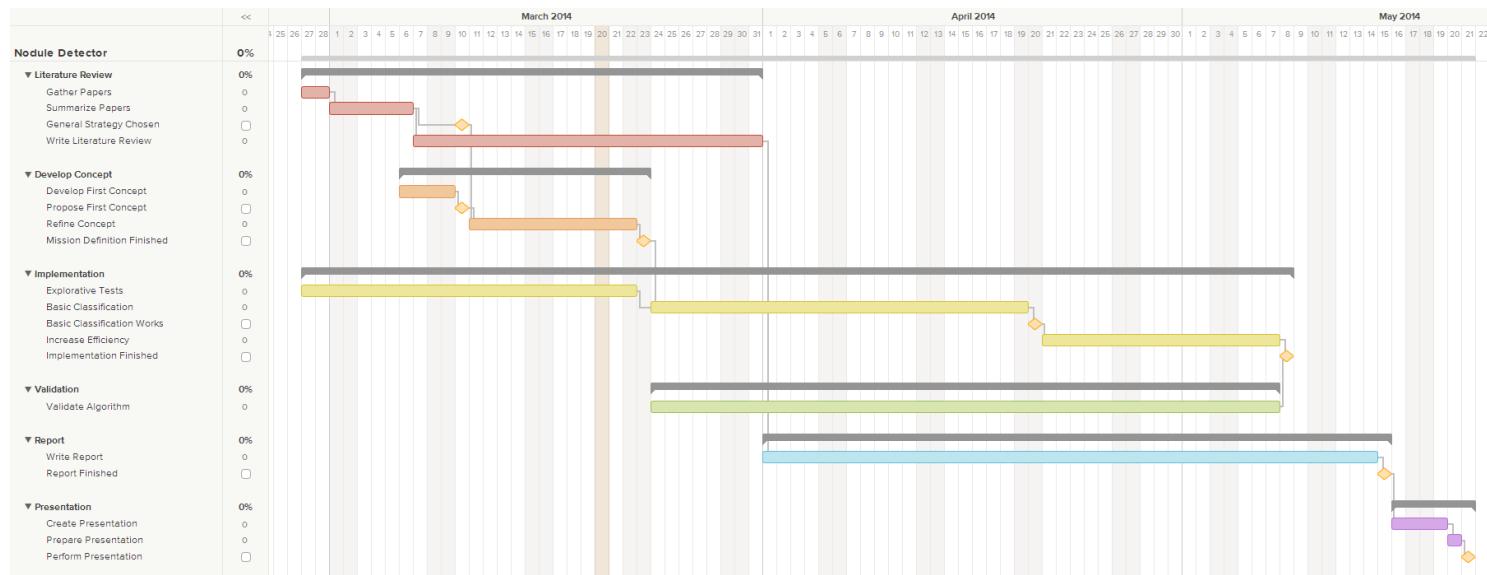


Figure 11: Gantt chart

	Datum	Wie	Uren	Type	Beschrijving
Week 3	wo 26/02/2014	Sven	2	Meeting	Voorbereiding meeting 1
	wo 26/02/2014	Kim	2	Meeting	Voorbereiding meeting 1
	do 27/02/2014	Team	2	Meeting	Voorbereiding meeting 1
	do 27/02/2014	Team	1	Meeting	Meeting 1
	do 27/02/2014	Team	1	Technisch	Configureren computer Kim (eclipse, LaTeX)
	do 27/02/2014	Sven	2	Technisch	Configureren computer Sven (github, git, python, eclipse)
	vr 28/02/2014	Sven	3	Technisch	Configureren computers + intro python
	vr 28/02/2014	Kim	3	Literatuur	Literatuurstudie
	za 01/03/2014	Sven	1	Verslag	Verslag meeting 1 uitgetypt
	zo 02/03/2014	Sven	2	Technisch	Tutorial GIT + intro MeVisLab
Week 4	di 04/03/2014	Kim	2	Literatuur	Literatuurstudie
	do 06/03/2014	Kim	3	Literatuur	Literatuurstudie
	do 06/03/2014	Kim	2,5	Literatuur	Literatuurstudie
	do 06/03/2014	Sven	4,5	Technisch	Packages python + mevislab
	vr 07/03/2014	Kim	3	Literatuur	Literatuurstudie
	vr 07/03/2014	Sven	3	Technisch	Testen MeVisLab
	zo 09/03/2014	Kim	3	Meeting	literatuurstudie +Voorbereiding meeting 2
Week 5	ma 10/03/2014	Team	1	Meeting	Meeting 2
	do 13/03/2014	Team	4	Varia	Literatuurstudie + Interpretatie CT + Testen MeVisLab
	vr 14/03/2014	Team	4	Varia	Testen MeVisLab
	vr 14/03/2014	Sven	3	Technisch	Eerste poging XML parser
	za 15/03/2014	Kim	1,5	Technisch	CT scans: organen terugvinden (hulp zus)
	za 15/03/2014	Kim	2	Technisch	Test MeVisLab: Xmarkers
	zo 16/03/2014	Team	3,5	Technisch	Exploratief testen (RF, SVM) + XML Parser afwerken
Week 6	di 18/03/2014	Kim	3,5	Verslag	Uitschrijven mission definition
	do 20/03/2014	Team	4,5	Meeting	Mission Definition + Meeting met Pieter, David en Yoni
	do 20/03/2014	Sven	1,5	Verslag	Verslagen meetings 2 en 3 bijgewerkt
	vr 21/03/2014	Sven	1	Technisch	World matrix berekenen a.h.v. DICOM header
	vr 21/03/2014	Team	3	Verslag	Mission Definition afwerken
	vr 21/03/2014	Kim	2	Verslag	Mission Definition afwerken
	vr 21/03/2014	Sven	1	Verslag	Mission Definition afwerken
	za 22/03/2014	Sven	1	Verslag	LaTeX project aangemaakt voor eindverslag

Week 7	wo 26/03/2014 Team do 27/03/2014 Kim do 27/03/2014 Team vr 28/03/2014 Team za 29/03/2014 Sven zo 30/03/2014 Sven zo 30/03/2014 Sven	3 Varia 2 Technisch 4 Technisch 3,5 Technisch 2 Technisch 0,5 Meeting 0,5 Technisch	World matrix berekenen (correctie) + Technieken opzoeken in literatuur snelcursus Phyton Xmarker coordinaten berekenen in python + testen longsegmentatie Implementatie longsegmentatie (poging 1) Longsegmentatie: code optimaliseren Meeting 4 voorbereiden Python in Visual Studio testen
Week 8	ma 31/03/2014 Sven di 01/04/2014 Team wo 02/04/2014 Team wo 02/04/2014 Team wo 02/04/2014 Kim do 03/04/2014 Team vr 04/04/2014 Team vr 04/04/2014 Kim	5 Technisch 3,5 Varia 1 Meeting 2 Varia 2 Technisch 2 Technisch 1,5 Technisch 1 Technisch	Longsegmentatie: code optimaliseren Longsegmentatie: stap 1 afwerken + Meeting 4 voorbereiden Meeting 4 Verslag meeting 4 + thresholds vergelijken over slices en datasets Histogram onderzoeken Histogram onderzoeken + Start featureselectie Fixed thresholding + Feature selectie Fixed thresholding
Week 9	wo 09/04/2014 Team do 10/04/2014 Team vr 11/04/2014 Team za 12/04/2014 Team	3,5 Technisch 9,5 Technisch 12 Technisch 7,5 Technisch	Long segmentatie + features berekenen Nodule segmentatie o.b.v. annotaties + features berekenen Nodule segmentatie (polygon + sphere) + features Training classifier + features
Week 10	do 17/04/2014 Kim do 17/04/2014 Team do 17/04/2014 Team vr 18/04/2014 Sven za 19/04/2014 Kim zo 20/04/2014 Kim	3 Meeting 1 Meeting 4 Technisch 6 Technisch 1,5 Technisch 1 Technisch	Meeting 5 voorbereiden Meeting 5 3D longsegmentatie + non-nodule pixels random zoeken PixelFinder gemaakt, ClassificationTest verbeterd, Probabilistische afbeelding gegenereerd. SphereTest 3D Probability image gemaakt
Week 11	ma 21/04/2014 Team di 22/04/2014 Sven wo 23/04/2014 Team do 24/04/2014 Kim do 24/04/2014 Team do 24/04/2014 Team vr 25/04/2014 Kim	10 Technisch 2 Technisch 4 Varia 1 Verslag 5 Varia 1 Meeting 4 Technisch 2 Verslag	Pixels uit segmentatie halen, 3D Probability image gemaakt Afbeeldingen voor mail Pieter gemaakt: histograms, probimg voor intensity/location Werken aan verslag + code optimalisatie om memory errors te vermijden Verslag Voorbereiding meeting 6 + verslag + goede features zoeken + entropy + memory errors Meeting 6 Architectuur op orde gebracht + klassediagram gemaakt Info opzoeken

	vr 25/04/2014 Sven	1 Technisch	Masks Pieter getest
	za 26/04/2014 Kim	5 Verslag	Schrijven
	za 26/04/2014 Sven	2 Technisch	Classifier data in stukken inladen
	zo 27/04/2014 Kim	2 Verslag	Schrijven
Week 12	ma 28/04/2014 Team	3,5 Varia	Verslag + laplacian feature + nd arrays maken van tuple van features
	ma 28/04/2014 Kim	3 Verslag	Overzicht relevante papers + bespreking toegepaste methodes
	di 29/04/2014 Team	4 Varia	Verslag + Meeting 7 voorbereiden + ...
	wo 30/04/2014 Team	1 Meeting	Meeting 7
	wo 30/04/2014 Team	5 Varia	Verslag + laplacian sigma update + masks bijwerken
	do 01/05/2014 Team	9,5 Varia	Verslag (architecture) + piramides + laplacian sigma update + enkele fouten uit programma wegwerken
	za 03/05/2014 Kim	2 Varia	50 masks genereren in mevislab + verslag (features begin)
	zo 04/05/2014 Sven	3 Technisch	Distance map + bug fixing
Week 13	ma 05/05/2014 Team	11 Technisch	Validatie + memory errors + features(3d averaging, skewness, kurtosis)
	di 06/05/2014 Kim	3,5 Technisch	Discussie over parameters van features
	di 06/05/2014 Sven	5 Technisch	Small bugfixes
	wo 07/05/2014 Kim	9,5 Technisch	Extra level classifier + trainen en testen van classifier (memory errors op level 3)
	wo 07/05/2014 Sven	1 Technisch	Small bugfixes
	do 08/05/2014 Sven	5 Technisch	Nodule radius & laplacian sigma's testen + validatie over meerdere datasets
	vr 09/05/2014 Team	11 Varia	Verslag + Trainingfeatures in 1x berekenen + validator optimaliseren + nodule radius test
	za 10/05/2014 Sven	4 Varia	Verslag + resultaten verzamelen + code optimaliseren
	zo 11/05/2014 Sven	6 Varia	Verslag + bugfixes + resultaten verzamelen
	zo 11/05/2014 Kim	2 Verslag	Inleiding, conclusie, ...
Week 14	ma 12/05/2014 Team	5 Varia	Verslag + resultaten verzamelen
	ma 12/05/2014 Sven	6 Varia	Verslag + resultaten verzamelen
	ma 12/05/2014 Kim	3 Verslag	Conclusie, nalezen
	di 13/05/2014 Sven	5 Verslag	Verslag nalezen
	di 13/05/2014 Kim	5 Verslag	Verslag nalezen
	wo 14/05/2014 Sven	3 Verslag	Verslag nalezen + afdrukken + inbinden
DEADLINE	do 15/05/2014 Team	1 Verslag	Verslag indienen
	??? Team	10 Presentatie	Presentatie voorbereiden
	wo 21/05/2014 Team	3 Presentatie	Presentaties geven/bijwonen
	Totaal Kim:	242 / 150	
	Totaal Sven:	250 / 150	

A.2 Meetings

Verslag Meeting I

Do 27/02/2014 - 16u-17u - MIRC

Onderwerp

Shape-based feature analysis for nodule detection in lung images.

Team

- Prof. dr. ir. Paul Suetens (promotor)
- Dr. ir. Pieter Slagmolen (research manager)
- Ir. David Robben (doctoraatsstudent)
- Dr. Yoni De Witte (AGFA)

Organisatorisch

- Belcodes MIRC
 - Pieter: **49050**
 - David: 016 34 90 39
- Rol AGFA:
 - Minimaal, eerste samenwerking in jaren
 - Presentatie in Mortsel op einde?
 - Doenbaar als commercieel product?
- Rol promotor: in jury (tekst, presentatie)
- Waar werken: MIRC volzet, AGFA niet praktisch → thuis
- Yoni zal via teleconferencing toekomstige meetings bijwonen

Technisch

- Programmeertaal = python (+ open source libraries)
 - Opm.: python niet zo goed voor 3D visualisaties
 - Exploratief testen met MeVisLab (<http://www.mevislab.de>)
 - Library (machine learning): scikit-learn (<http://scikit-learn.org/stable>)
 - Library (images): enthought canopy (<https://www.enthought.com/products/canopy>)
- Suggestie algoritme: random forests (*hot topic* in medische beeldvorming)
 - Literatuurstudie: zoek enkele alternatieven, maar waarschijnlijk niet beter
 - Vooral: zoek uitbreidingen op standaard algoritme (hogere efficiëntie)
 - cascading classifiers (weak → strong)
 - eerst 2D, dan 3D
 - Selecteer “goede features” uit data
 - “Normale” 1^e master studenten: blij als basisclassificatie lukt
- Testbeelden in online catalogoog
 - Vooral CT scans, DICOM formaat
 - XML annotations (wrapper schrijven?)
 - Werken in 3D!

- Mission Definition: specificaties
 - Robustness, accuracy, sensitivity, specificity, receiver operating characteristic (ROC)...
 - Paper: 97%-100% succes = dubieuw in praktijk
 - Doe literatuurstudie om parameters en redelijke waarden te vinden
- Preprocessing
 - Hough transform niet ideaal in 3D
 - Image registration (goede overlap) mogelijk nodig
- Grootte trainingsset
 - Experimenteel vastleggen
 - Efficientie vlakt ~logaritmisch af, rekentijd neemt toe → trade-off
 - MIRC heeft rekenclusters indien echt nodig
- Opmerkingen:
 - Wat klasseren als positief? Centrale voxel of ook omgeving?
 - Voxel size kan verschillen per dataset

Medisch

- Benodigde medische kennis minimaal
 - Net iets meer dan “grijze bolletjes”
 - 10-30mm
 - Cirkelvormig, maar soms onregelmatig
 - “Verstopt” achter wall
 - Benign irrelevant? (in XML data?)
 - Verschillende soorten onbelangrijk (intro 2blz. voldoende)
 - Niet te verwarren met bloedvaten
 - MRI scans niet nodig?

Planning

- Belangrijke data
 - Zo 09/03: literatuurstudie af
 - **Ma 10/03: meeting 2 (18u @ MIRC):** “conceptueel plan” klaar
 - Ma 17/03: eerste concept
 - **Za 22/03: deadline Mission Definition**
 - **Do 03/04: bezoek GHB @ 14u**
 - **Vr 04/04: IP seminarie**
 - Vr 09/05: deadline design (informeel?)
 - **Do 15/05: deadline verslag @ 14u**
 - **Wo 21/05: presentaties** (AGFA, promotor present)
- Budget: 150u
 - Analysis of existing methods (non-exhaustive) [15%] = 22u30
 - Design of the method and its components [15%] = 22u30
 - Implementation of the method [50%] = 75u
 - Validation on open data and reporting [20%] = 30u
 - Logboek = excel in dropbox

Verslag Meeting II

Ma 10/03/2014 - 18u-19u - MIRC

Aanwezig

- Dr. ir. Pieter Slagmolen (research manager)
- Ir. David Robben (doctoraatsstudent)
- Ir. Sven Van Hove
- Ir. Kim Nuysts

Agenda

Besprekking literatuurstudie

- Probleem: literatuurstudie geeft geen directie indicatie dat RF de beste methode is
- Mail Pieter: overzicht commerciële toepassingen nodule detectie (in scanners)
- Mail David: papers
 - Caruana: vergelijking classificatie methodes
 - Van Ginneken en Murphy: ANODE challenge
- ANODE challenge:
 - Standaard dataset → interessant voor validatie
 - Objectieve methodevergelijking
 - Participeren in wedstrijd?
- Zie ook: The Medical Image Computing and Computer Assisted Intervention Society (MICCAI)

Besprekking conceptueel plan

- Pieter: benader project als commercieel product
 - Bv: real time verwerking, max 15-30min
- Nog verder uitwerken? Kwantificeren?

Besprekking tijdschema (Gantt chart)

- Deadline basic classification opschuiven (einde paasvakantie)

Besprekking technische problemen

- MeVisLab module ontwerpen i.p.v. stand-alone python script?
 - Nee, gemakkelijker om in python te programmeren dan in MeVislab
 - MeVisLab debugger is niet bij de beste
 - Eventueel wel voor classificatie te gebruiken
 - Later code omzetten in MeVisLab module [extra]
- Data inladen in Python: consistent zijn met dataformaat
 - Slice per slice
 - Als 3D volume
 - Verschillende bestandsformaten (bv. NIFTY)
- Meeste datasets die onder longen geklassificeerd staan bevatten gaan annotaties
 - Gebruik Lung Image Database Consortium (LIDC) dataset

- Annotaties bevatten Malignancy probability per nodule, relevant of nuttig?
 - o In eerste instantie niet gebruiken
 - o Vervolgens eventueel opdeling in klassen maken [extra]
- Annotaties bevatten verslagen van meerdere radiologen
 - o Neem simpelweg de eerste (consistent zijn)
 - o Gebruiken indien wij fouten maken → maken zij ze ook?

Waarom kiezen voor Random Forests?

- Support Vector Machines
 - o Vanaf 20 000 te traag
 - o Noodzakelijk om features met dezelfde grootheden te gebruiken
 - Anders lineariseren, herschalen enz. → slecht
 - o Problemen met ruis
 - o Probleem indien features nog niet op voorhand gekend
 - Hoe clustering doen (invloed op rekentijd!)
 - Eventueel filters voor features
- Random Forests
 - o Kan belangrijke features zelf bepalen
 - o Beter voor grote datasets
 - o Niet zo gevoelig aan ruis
- Boosted trees
 - o Op eerste zicht beter dan RF (Carauna)
 - o Problemen
 - Slecht bestand tegen ruizige/ambigue problemen (onze beelden)
 - Keuze van parameters wordt hierdoor gevoeliger en moeilijker (boomdiepte, aantal bomen, learning rate...)
 - Training is inherent sequentieel (trager dan RF)
- Methode E, isi-cad (Murphy)
 - o KNN classifiers: suboptimale keuze (opportunitet om state-of-the-art te kloppen)
 - o Zelfde probleem als SVM indien features verschillende grootheden hebben
- Algemene opmerkingen overzicht methodes
 - o Preprocessing en feature selectie (arbitrair???) zeer belangrijk
 - o Goede rekentijd: gebruik cascaded classifiers

Vragen

- Python versie 2.7 of 3.3?
 - o Gebruik 2.7 wegen backwards compatibility problemen in 3.x
- Gebruik chest X-rays voor eerste localisatie?
 - o Nee, zou ons te ver leiden. Mogelijk ook niet allemaal zichtbaar.
- Annotaties: nodules vs non-nodules vs other?
 - o Check bijbehorend paper (non-nodules = kanker maar geen nodulevorm?)

Volgende meeting

Donderdag 20/03/2014 @ MIRC, 17u00

Verslag Meeting III

Do 20/03/2014 - 17u-18u - MIRC

Aanwezig

- Dr. ir. Pieter Slagmolen (research manager)
- Ir. David Robben (doctoraatsstudent)
- Dr. Yoni De Witte
- Ir. Sven Van Hove
- Ir. Kim Nuyts

Agenda

Vooruitgang sinds vorige meeting

- XML annotaties inlezen met python (lxml package)
- DICOM bestanden per slice inlezen met python (pydicom package)
 - Waar zit de world matrix?
- Exploratief testen m.b.v. MeVisLab
 - Interpretatie beelden
 - View2D, View3D, OrthoView2D
 - X-Markers
 - Gebruik wereldcoordinaten
 - Klik op apply...
 - CSO's gaan ons te ver leiden
- Eerste testen SVM/RF op IRIS database

Bespreking Mission Definition

- Functional decomposition
 - “visualisation” -> only 1 module in Mevislab (not really the aim of this project)
 - No values assigned: difficult as every part is essential for functioning of whole program
- Processing time (Yoni)
 - Not necessary to improve this in Python, but the code should allow to be implemented in a compiled language (eg C++) for faster processing (so far all the processing is still done on a workstation, future work on server)
 - Processing time goal: processing is allowed to take few minutes in compiled language (real time not necessary), as long as it is done when the radiologist wants to investigate the results after scanning the patient

Planning komende week

- Mission Definition afwerken & doorsturen

- Exploratief testen afronden
- Beginnen met basisclassificatie
- Grote lijnen tekst uitschrijven

Vragen

- Visualisatie niet erg belangrijk -> hoe resultaten dan tonen?
- Featureselectie door RF?
 - o Cascaded: (1) berekening honderden features (2D of 3D) ; (2) selectie features
- Minimum Redundancy, Maximum Relevance method?
 - o Zowel mRMR als selectie op basis van RF is mogelijk
- Subtlety: niet noodzakelijk implementeren
 - o Mogelijk interessant: vinden algoritme en radioloog het even moeilijk om bepaalde nodules te detecteren?
- World matrix? (Yoni)
 - o DICOM tag bevat
 - Image position (afh van scanner) en orientation (vnl. axial)
 - Aantal rijen en kolommen + pixelspacing + slice thickness (3D volume)
 - o Test: intensiteiten nodules over hetzelfde?

Volgende meeting

Woensdag 02/04/2014 @ MIRC, 09u00

Verslag Meeting IV

Wo 02/04/2014

Aanwezig

- Dr.ir. Pieter Slagmolen
- Ir. David Robben
- Dr. Yoni De Witte
- Ir. Sven Van Hove
- Ir. Kim Nuyts

Agenda

Vooruitgang sinds vorige meeting

- Verslagen:
 - Mission Definition afgewerkt
 - Eindverslag: LaTeX template ingeladen & grote lijnen uitgezet
- Technisch:
 - World matrix correct berekend
 - Coordinaten succesvol ingeladen in MeVisLab
 - Lijst features opgezocht in literatuur
 - Sommige features hebben a priori nodule segmentatie nodig -> vermijden
 - Implementatie longsegmentatie o.b.v. Keshani et al. (adaptivefuzzythresholding)
 - Theorie: zie paper
 - Praktijk: zie demo

Planning komende weken

- Deze week niet veel tijd: bezoek GHB, IP seminarie
- Volgende week kleine 4 dagen ingepland
- Nu ongeveer 70u/150u gewerkt
- Verslag:
 - Eindverslag "organisch" laten groeien
- Technisch:
 - Longsegmentatie afwerken (5 stappen)
 1. afbakenen buitenste contour longen: afgewerkt
 2. Open ruimtes in afgebakende longen opvullen (restanten bronchiolen en nodules) mbv 2 windows (groot + klein)
 3. Niet geïsoleerde nodules (tegen wand aan) worden binnen de longcontour gezet
 4. Verbetering van de derde stap mbv window (stukken bot, grote (slag)aders en luchtpijp worden indien nodig verwijderd)

- 5. Het resultaat van de vierde stap wordt gebruikt als een mask voor active contour model
- Features selecteren
 - Per voxel, neem omgeving ook mee
 - Begin met korte featurevector, maak per cascade langer
 - RF goed voor grote datasets -> maak gebruik van (verschillende filter sizes)
 - “Data het zware werk laten doen”
 - Laplacian
 - Haar features -> snel te berekenen (cumsum)
 - Lokaal of op heel beeld?
 - 2^e afgeleide Gaussiaan (blobdetectie)
 - Afstand voxel tot longrand
 - Eerst threshold om (donkere) longen te negeren
- Classificatie
- Deadline basisclassificatie: einde paasvakantie (20/04/2014)

Vragen & Opmerkingen

- ExtraTrees O(n) is alternatief voor RF O(n log n)
 - Sneller, bijna even goed
- Vermijd for-loops in python -> gebruik matrixalgebra numpy/scipy
 - Histogram functie uit scipy gebruiken
- MR scans nooit gebruikt voor longen
 - Te traag
 - Artefacten
- Motivatie longsegmentatie heel belangrijk!
 - Rekentijd!
 - Is ruwe segmentatie goed genoeg? -> active contours niet nodig
 - Gebruiken simpele bounding boxes?
 - Segmentatie heel moeilijk voor bepaalde pathologieën
 - Is soort van eerste simpele classifier in cascading systeem
 - Doel is 100% recall van nodules, FP minder belangrijk initieel
 - Focus op eenvoud, robuustheid
 - Focus project = classificatie -> niet te veel tijd in segmentatie steken
 - Te ingewikkeld?
 - CT normaal gecalibreerd
 - Werkt constante threshold ook?
 - Vergelijk constante en berekende threshold op verschillende beelden
- Check om te zien of slice longen bevat?
 - Niet nodig, classifier zal dit wel oplossen

Volgende meeting

Donderdag 10/04 @ MIRC, 17u00

Verslag Meeting V

Do 17/04/2014 (uitgesteld van 10/04/2014)

Aanwezig

- Dr.ir. Pieter Slagmolen
- Ir. David Robben
- Dr. Yoni De Witte
- Ir. Sven Van Hove
- Ir. Kim Nuyts

Agenda

Vooruitgang sinds vorige meeting

Longsegmentatie

- Doel: ruwe aflijning om rekentijd in te korten
 - Vraag: hoeveel pixels gooij je er dan uit? REKEN UIT! Schatting: 50%
- Vorige algoritme te complex
- Test constante threshold: OK
- Methode:
 - Threshold (1600) om de chest wall van de rest te scheiden
 - Morfologische opening om kleine blobs te verwijderen
 - Morfologische reconstructie om longen op te vullen
 - Morfologische erosie om het te grote gebied wat kleiner te maken
- ➔ FixedThresholdTest.py
- TODO: zet threshold van 2D naar 3D implementatie
 - Probleem: erosie wil niet meer werken in 3D

Aflijning nodules a.h.v. vertices

- Cirkel per slice: OK
- Veelhoek per slice: OK
- Volume
 - Ellipsoïde in pixel coordinates
 - Omzetten naar wereldcoordinaten -> bol
 - Nog altijd $R_z \ll R_x, R_y$???
 - Efficiëntieproblemen
 - Toch beter formule ellipsoïde gebruiken, of cirkel per slice

Suggesties Pieter:

- gebruik niet midden ($[X_{\max} - X_{\min}] / 2$) maar bereken zwaartepunt → euclidische afstand
- geen overlap, noch spaties tussen slices

- hebben we de aflijning van de nodules als bollen nodig?
 - o Eventuele meerwaarde: bereken de gradiënt in intensiteit naar buitenkant van bol toe als feature voor malignancy
 - o Nee, betere oplossing: neem polygoon per nodule en bereken zwaartepunt (niet midden) en doe dan erosie van de polygoon om buitenste rand te elimineren
- Werken met enkel pixels in 'midden' zou geen probleem met 'holle nodules' mogen geven zolang we niet enkel intensiteit van individuele pixels als feature gebruiken

Implementatie features

- Features die geen a priori nodule segmentatie nodig hebben
- Volume Features (3D)
 - o Edges (Sobel) -> window
- Slice Features (2D)
 - o Entropy -> zwaar
 - o Blobs (max LoG)
 - TODO: 3D (voxelgrootte!) -> beter op kleiner window (om geheugen te sparen)
 - Gebruik enkel scipy library: `scipy.ndimage.convolve` (wel 3D)
- Pixel Features
 - o Intensity
 - o Location (absolute, relative)
 - o Frobenius Norm (distance to center)
 - o Neighbours (L-R, T-B, ...)
- Window Features ($n = 3:32:2$)
 - o Nodule -> window size 50 -> traag
MAAR wel gebruiken om ook grote nodules te segmenteren! Eventueel pas later in cascade? Eventueel meer stappen ertussen laten?
 - o Intensity (mean, variance, skewness, kurtosis, max, min, ...)
 - o Left/Right/Top/Bottom/Front/Back row -> mean, grad, ...
 - Implementeren met convolutiefilters?
- ➔ `featureselection.py`
- TODO: entropy, haar, ...
 - o Haar features
 - SimpleCV werkt niet goed in Python (David) -> zelf programmeren (numpy)
 - Pas laat in cascade gaan gebruiken
 - o Entropy: pas laat in cascade gaan gebruiken (wel snel berekend op hele image)
- Cascading? Gebruik `predict_proba` (geeft rijen = samples en kolommen = klassen met per klasse een probabiliteit)
 - o Niveau 1: Gebruik bv 1 simpele feature (intensiteit) en schrap daarmee deel van de trainingsdataset (alle pixels die 99% zeker geen nodule zijn eruit halen)
 - o Niveau 2: gebruik iets gesofisticeerdere features en elimineer weer deel van de dataset
 - o Niveau 3: idem
 - o ...

Classificatie

- Eerst: trainen met enkel nodule pixels binnen cirkel r/3: OK (triviaal), maar traag
- Vervolg: non-nodule pixels samplen (welke?)
 - TRAININGSDATASET (Pieter): gebruik helft pos, helft neg + sample random in 3D
- ⇒ Genereer **een probabilistisch beeld** na elke stap in de cascade en bekijk of het een goede stap is of niet (= meer 1 en minder 0 voxels? En behouden we alle nodules of moeten we extra features toevoegen om alle nodules te behouden?)
 - Vgl ook steeds met Mevislab (.jpeg, .dicom, .tif, .nifty... pydicom)
- ⇒ THRESHOLDING mogelijk niet meer nodig indien de eerste classificatiestap er al alle achtergrond pixels uitsmijt

Planning komende week

- Lichaamsegmentatie uitbreiden naar alle slices (slice per slice of 3D?)
- Basisclassificatie uittesten met zowel nodule als non-nodule pixels
- Cascaded classifier implementeren
- Meer features implementeren (haar, entropy, ...) Suggesties?
 - Beter eerst huidige features analyseren

Vragen & Opmerkingen

- Welke pixels uitkiezen voor
 - nodule training
 - non-nodule training -> willekeurig! Maar niet te dicht in buurt van nodules
- We hebben nu meer dan 500 features (met windowvariaties inbegrepen) die van elke voxel in de scan moeten bepaald worden. Tot nu toe hebben we het proces versneld door de data te downsamplen, maar zelfs dan duurt het nog redelijk lang: een paar minuten voor 1 nodule. Daarom hadden we onze maximum windowgroottes ook al beperkt (van 50 tot 30). Het algoritme op alle pixels met alle windowgroottes te laten draaien zal veel tijd vragen.
 - Ligt deze rekentijd in de lijn van verwachtingen? Of doen we iets fout waardoor onze rekentijd onredelijk lang wordt?
 - Voor het trainen van de classifier zijn er volgens ons twee mogelijkheden. Ofwel samplen we de scan en nemen we om de zoveel pixel er een uit die op 0 (geen nodule) of op 1 (nodule) kan gezet worden. Ofwel gaan we explicet op zoek naar de positieve pixels en berekenen we de negatieve apart. (hier ontstond ook wat verwarring: voorlopig werken we met een cirkel per slice)
 - → windowgroottes minder laten varieren in het begin
- Bij het resizen van onze dataset (omwille van bovenstaande efficiëntieredenen) zien we ook dat onze nodulemasks helemaal vervormd worden.
 - ➔ Door subsamplen dataset: mogelijk voor developing, in def versie zeker niet!! En let op, want zo verlies je mogelijk info over de kleine nodules
 - ➔ Neem liever ROI rond nodule (beetje long, beetje bot, beetje spier ...) om sneller development te kunnen doen
- De Haar Features: We vinden een library van SimpleCV die met deze haar features werkt. Python doet echter moeilijk als we deze willen inladen. Is er een beter pakket hiervoor?

- Entropy feature: We hebben een feature die de entropy van de afbeelding bepaald. Hiervoor moeten we echter een .view('uint8') commando gebruiken. Hier ondervinden we ook problemen mee. (from skimage.filter.rank import entropy)
 - ➔ Gebruik NIET .view, gebruik WEL .astype('uint8')
- Cascaded feature selection: Hoe pakken we dit concreet aan?
- We hebben twee methodes gevonden om output te genereren met ExtraTrees. Daarvan hebben we deze gekozen omdat we dachten dat die probabiliteiten als uitkomst zou geven, maar er is geen verschil in resultaat merkbaar:
 - result = clf.predict_proba(X[0,:])
- Voorlopig sampelen we door de slices met nodules in om een trainingsset te genereren: pixels worden op 0 (geen nodule) of 1 (nodule) gezet. Sampelen we best ook door alle andere slices om ook daar de robuustheid tegen FP te verhogen? Dit gaat dan wel ten koste van de rekentijd.
 - We dachten de rekentijd te beperken door enkel door die slices te sampelen waar nodules inzitten. Dit zijn de belangrijkste voor de radioloog om het aantal FP's in te reduceren (in onze ogen). Een radioloog die een FP ter hoogte van de lever krijgt, ziet dat zelf ook meteen en kan die dus negeren. Maar een FP op een plaats waar je nodules redelijkerwijs kan verwachten, willen we wel zeker vermijden.

Volgende meeting

Donderdag 24/04/2014 @ MIRC, 17u00

Verslag Meeting VI

Do 24/04/2014

Aanwezig

- Dr.ir. Pieter Slagmolen
- Ir. David Robben
- Ir. Sven Van Hove
- Ir. Kim Nuyts

Agenda

Memory errors!!

- training subsampelen, niet voor testen
- segmentatie: Pieter gaat ons scriptje geven voor longsegmentatie (MeVisLab)
- chunks van data aan classifier geven ipv in geheel

Cascaded classifier

- 1) Relatieve positie (tov nulpunt beeld): enkel waarde indien heel veel trainingsdata of indien tov rand longen ofzo
- 2) Entropie: hoelang rekenen per voxel? Ipv met bol met een rechthoekig window werken
- 3) Laplacian: verschil tss opeenvolgende Laplacianen (multilevel) -> standaardfunctie?
- 4) Window-features: for loops vermijden door ev beeld steeds te transponeren + sommeren, maar moeilijk

Validatie

- Niet met afstand tot middelpunt van nodule, gewoon ligt 'hit' erin of niet? + dilatatie van nodule annotaties om randen erbij te hebben

Vooruitgang sinds vorige meeting

- Nodules zijn zowel in MeVisLab als in onze testen platgedrukte bollen
 - Radiologen negeren uiteinden?
- Lichaamssegmentatie werkt niet in 3D (memoryerrors e.d.) -> on hold -> 2D m.b.v masks (**44%**)
 - Enkel intensiteiten voor alle voxels geeft al MemoryError in classifier.
- Code geoptimaliseerd om rekentijd te verlagen en memory errors te vermijden
 - Bv. lijsten van **pixelcoordinaten -> masks**
- Classifier getraind op tiental datasets
 - Cascada 1 = intensiteiten (**32%**)
 - Cascade 2 = (x,y,z) -> **z (niet!)** -> entropy -> **memory errors in classifier**
 - Positie in pixelcoordinaten vanuit linkerbovenhoek
 - Moeilijkheden scanpositie, grootte, te specifiek
 - Cascade 3 = ???

Planning komende week

- Cascaded classifier uitbreiden: meer en betere features
- Validatie (FP, accuracy, sensitivity, specificity,... berekenen en vergelijken)
- Verder werken aan verslag

Vragen & Opmerkingen

- Hoe validatie in de praktijk brengen
- **Features suggesties: zoveel mogelijk**

Volgende meeting

30/04/2014 @ MIRC, 17u

Verslag Meeting VII

Wo 30/04/2014

Aanwezig

- Dr.ir. Pieter Slagmolen
- Ir. David Robben
- Ir. Sven Van Hove
- Ir. Kim Nuyts

Agenda

Memory errors

Alles nalopen: ergens onregelmatigheden? Nee! Dus waarschijnlijk een probleem van 32-bit python i.p.v. 64-bit.

Cascaded classifier

- Laplacian
 - Edges: zerocrossings (min-max): zelfde max als nodules?
 - Multiscale: sigma's aangepast aan blobgrootte -> 4D piramide
 - Sobel na Laplacian om edges weg te werken? -> smoothin nodig
 - LET OP! Detecteert met uniforme sigma's ellipsoïde en geen bol DUS aanpassen aan voxelafmetingen
- Vergelijken met SVM? (probleem gecorreleerde features)
- Parameters trees:
 - Niet te diep: overfitten (moet nog generaliseerbaar zijn)
 - In laatste leaf nog bv 100 samples
 - Crossvalidatie: sklearn: gridsearchcv (fit doen) -> diepte boom optimaal -> daarna parameters hard coderen

Verslag

- Toon dat je alles begrijpt
- Summary!
- Aantal trainingsdata; features; beperkingen; ...

Klassendiagram gemaakt: niet veel in verslag

Vooruitgang sinds vorige meeting

- Verslag → literatuurstudie nalezen?
- Featurevectors in chunks inladen
- Segmentatie: thresholded – reconstruction = longen (50% -> 20%)
 - Dilatatie om rand + nodules op te nemen

- Masks Pieter ingeladen
 - Minder memory errors
 - Region growing -> nodules in mask?
- Design geoptimaliseerd + klassediagram
 - Variabelen in stack bekijken, niets onregelmatig
 - Bespreken in eindverslag?
- Cascade 2: Laplacian (ndimage.filters.gaussian_laplace)
 - Edges = low->zero->high → thorax wall blijft
 - Sigma: 1-5
 - Gaussian scale pyramids: 3D -> 4D
 - $\sigma^2 = t \rightarrow \text{"extent"} \sqrt{2t} = \sqrt{2}\sigma$

Planning komende week

- Cascaded classifier uitbreiden: meer en betere features
- Validatie (FP, accuracy, sensitivity, specificity,... berekenen en vergelijken)
- Verder werken aan verslag

Vragen & Opmerkingen

- Cascading: "data het zware werk laten doen" vs cherry-picking

Volgende meeting

N/A