

COMPUTER AIDED DIAGNOSIS LUNA16: CANDIDATE DETECTION

*Luc Nies (s4136748), Tom van de Poll (s4106512), Harmen Prins (s4132297),
Steven Reitsma (s4132343) & Inez Wijnands (s4149696)*
Radboud University Nijmegen

ABSTRACT

Write your abstract here.

1. INTRODUCTION

In the *Lung Nodule Analysis 2016 (LUNA16)* challenge, we aim to detect pulmonary nodules in low-dose thoracic CT images. Firstly, we have segmented the lungs from the scans. The next step is identifying candidates for lung nodules. Our aim is to detect enough candidates to include all actual lung nodules. In other words, be as sensitive as possible. The future step is of course to remove the false positives, but first we strive towards including the annotated nodules.

We continued working on the deep learning approach we used for the previous phase. In that phase we worked on lung segmentation and for this phase we treated candidate detection again as a segmentation problem. Secondly, we also used an image processing approach where we use the features of nodules such as blobness for candidate detection. These approaches and their results are explained more detailed in the following sections.

Table 1. Caption here.

text	text	text
text	text	text

You can include figures as follows:

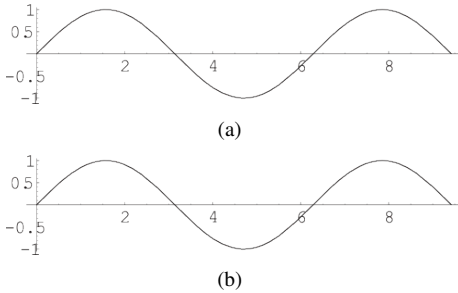


Fig. 1. Caption here.

2. METHOD

2.1. Fully convolutional networks

2.1.1. Lung segmentation

During the previous phase, we used fully convolutional networks [1] to segment the lungs from the background. When we wrote the previous report, we were unable to get this method to work in time, but since then we have correctly finished the implementation. We report our results with this method on the lung segmentation task first. Since this method can also be applied to nodule candidate detection, we will also report the results on this task.

Our final network architecture is based on the VGG net network [2], as Long et al. [1] reported that this architecture worked best for image segmentation. We adapted this network to fit our problem. This resulted in a 12 layer network:

1. Conv, 12 filters, 3x3 pixels
2. Conv, 24 filters, 3x3 pixels
3. Max pooling, 2x2 pixels
4. Conv, 48 filters, 3x3 pixels
5. Conv, 48 filters, 3x3 pixels
6. Max pooling, 2x2 pixels
7. Conv, 96 filters, 3x3 pixels
8. Conv, 96 filters, 3x3 pixels
9. Max pooling, 2x2 pixels
10. Conv, 1024 filters, 4x4 pixels, dropout 0.5
11. Conv, 1024 filters, 1x1 pixels, dropout 0.5
12. Conv, 2 filters, 1x1 pixels

In Figure 2 you can see the successful result of using the full lung slice as input of the fully convolutional network. The result is an image that is 57x57 in size. Due to the pooling layers and the lack of padding in the convolutional layers, the image is reduced in size compared to its input. To upscale the image, we employ the shift-and-stitch method as proposed

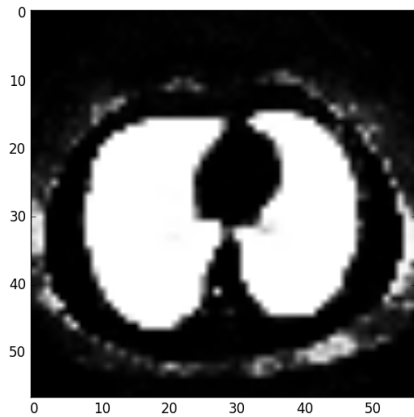


Fig. 2. The result of using the full lung slice image (size 512x512) as input in the network.

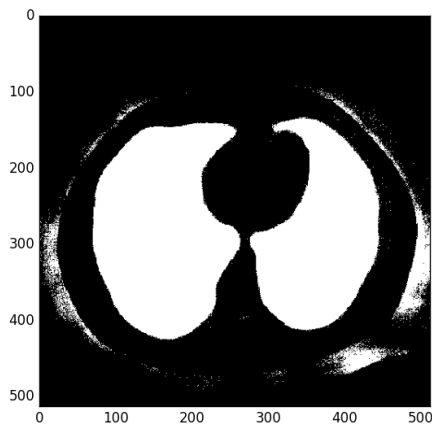


Fig. 3. The result after performing shift-and-stitch.

in [1]. The result of the shift and stitch method is shown in Figure 3. We can still see some parts besides the lungs that are marked as lung. However, this is something that can easily be fixed by using connected components analysis with a minimum volume filter. With this method, we attain a dice score of 0.985 on subset 9 (our validation set). Note that the network was trained on only four subsets (0-3) to speed up training. We expect that this method will perform even better when trained on all data sets.

2.1.2. Nodule detection with Fully Convolutional Networks

We employ the exact same technique for nodule detection. The only difference is in the labels. Whereas we use the segmentation masks as the labels in the lung segmentation task, we now use nodule masks. We parsed the annotations.csv file

and from this generated 3D masks that can be used as labels in the fully convolutional network. One extra challenge here is the extreme class imbalance. Even in the lung segmentation task, the class imbalance was something that we could not ignore. In that task, we employed class balancing by enforcing our minibatches to have a 50/50 split in terms of positive and negative samples. For the nodule detection task, this is harder, since we have even less positive samples. Even with data augmentation, the amount of positive samples is still too few. Due to time constraints and bugs left over in our implementation from the previous phase, we were as of yet unable to get this method to work. We will continue working on it during the next week. We know where the problem is (class imbalance) so we are positive we will get this method to work within a week.

2.2. Extracting candidates using Laplacian of Gaussian

We used a second approach for detection lung nodule candidates which is the more conventional approach of using their blob-like structure. The first step is applying the lung segmentation as a mask on the lung CT images to reduce the search space to the actual lungs. The algorithm for the actual candidate detection is as follows:

- Apply two different sized Laplacian of Gaussian (LoG) filters. A LoG filter is circular-symmetric and thus widely used for blob detection, but has a maximum response to a particular blob size. We combined two filters with normalized kernels (IS THIS NORMALIZED?), the first with $\sigma = 3$ and *kernel extent* = 7 and the second with $\sigma = 5$ and *kernel extent* = 11 CHECK VALUES. We chose these parameters based on the paper of Van de Leemput et al. [3], who used LoG filters for pulmonary nodule detection in the *ANODE09* challenge.
- Take the pixel-wise maximum of the results of the LoGs to combine them to one result.
- Threshold the resulting image with value `THRESHOLDVALUEAFTERLOGS`. See Figure 4 for an example result. This value is established by sampling the values of known nodules and healthy lung tissue.

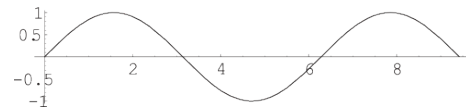


Fig. 4. The thresholded image after taking the maximum of the two LoGs.

- Do a connected component analysis on the resulting image, and select only components with values between `CONNECTEDCOMPONENTSVOLUMEVAL-`

UES. This identifies the initial candidates. See Figure 5 for an example result.

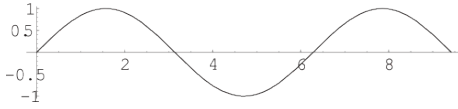


Fig. 5. The result of applying a connected component analysis on the image shown in Figure 4.

- Threshold the resulting image again with value THRESHOLD2VALUE.
- Apply dilation with a kernel of KERNELVALUE to increase the radius of the predicted candidates, i.e. predicted candidates will better correspond to lung nodule location. For example, if the prediction is only ten pixels removed from the actual nodule, this is really close in an image of $512 \times 512 \times \text{amount of slices}$ pixels. We still want this candidate to match with the ground truth. See Figure 6 for an example result.

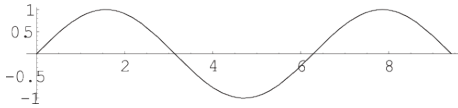


Fig. 6. An example result after applying dilation.

ADD IMAGES

3. RESULTS

3.1. Candidate segmentation using the FCN

None as of yet.

3.2. Candidate extraction using Laplacian of Gaussian

The results are based on the first 30 (AMOUNT) images in subset 9 of the dataset. The results are evaluated using the recall of the ground truth candidates given for this phase of the assignment, compared to our predicted nodule candidates. The reason that the results are only examined on SUBSET images is that this method is very time consuming (over ten minutes per image) and we continued tuning parameters and thereby improving our recall score relatively close to the deadline.

The average recall obtained by this method is MEANRECALLSCORE, with a standard deviation of STDRECALLSCORE. This score is relatively low, especially considering the high standard deviation. Additionally, increasing the recall drastically increased the total amount of predicted candidates which means there are a lot of false positives. However, the

main goal of this method is to include all nodules in our candidates. Thus, it was important to reach a recall as high as possible, and we will worry about reducing the false positives later.

4. DISCUSSION

A. CONTRIBUTIONS

Luc Nies: Final adjustments to the fully convolutional network so that it worked on the lung segmentation. Created the FCN pipeline, tested a lot of possible network and input parameters. Final adjustments to shift and stitch, and reading in of data. Wrote about the network architecture and parts of the nodule detection with FCN section.

Tom van de Poll:

Harmen Prins:

Steven Reitsma: Helped building the fully convolutional network to work correctly. Fixed buggy shift and stitch method. Created 3D nodule masks. Wrote report sections on fully convolutional network.

Inez Wijnands: Contributed to creating the MeVisLab candidate detection method using blob detection. Wrote introduction section and sections on the conventional approach in report.

B. REFERENCES

- [1] J. Long, E. Shelhamer & T. Darrell (2015). Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*: 3431–3440.
- [2] K. Simonyan & A. Zisserman (2014). Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*
- [3] S. van de Leemput, F. Dorssers & B.E. Bejnordi (2015). A novel spherical shell filter for reducing false positives in automatic detection of pulmonary nodules in thoracic CT scans. *Proceedings SPIE Medical Imaging*, 9414:94142P.