ZNET - LUNG NODULE DETECTION

*Moira Berens, Robbert van der Gugten, Michael de Kaste, Jeroen Manders, and Guido Zuidhof**

*In alphabetical order, all affiliated with Radboud University Nijmegen.

ABSTRACT

The goal of the LUNA16 (Lung Nodule Analysis 2016) challenge is to automatically detect nodules in 888 volumetric CT images. This is a description of the approach used in the *nodule detection track* which consists of two stages: candidate detection and false positive reduction. For the first part we employ the Unet fully convolutional network architecture for candidate selection on axial slices. For the subsequent false positive reduction we feed three orthogonal slices of each candidate to the same wide residual network. Our approach achieved an average FROC-score of 0.8111. In the *false positive track*, which uses the candidate set provided by the competition organizers, a score of 0.758 was achieved.

1. METHOD DESCRIPTION

1.1. Candidate selection via Unet

In the nodule detection track one is required to generate their own candidates from the raw scan files. Initially, we developed a Laplacian of Gaussian based hand-crafted approach, which achieved a high recall rate (around 0.98), but very poor precision (0.000005). Training on such an unwieldy, highly unbalanced dataset would be a challenge. Also, the false positive rate of a classifier trained on this set would probably be higher as it has more opportunities to make these errors.

We then employed the fully convolutional neural network architecture Unet [9] for this task. Please refer to the original paper for more details on this architecture. The original architecture had a tendency to overfit on the train set. To combat this we made some modifications to the Unet architecture:

- The input shape was changed to 512x512. Unet being a fully convolutional network, this also changed the output size, which then becomes 324x324. It was trained on the x,y planes of CT scans, with a uniform spacing of 1mm per voxel. This output size is large enough to contain the full lungs in every image.
- Batch normalization was added in the contraction part of the network. Due to the small batch size, ordinary batch normalization resulted in erratic behavior. To combat this, we set the α parameter to a value of 0.3.
- Dropout makes little sense in a fully convolutional network, hence, we only add it sparsely in specific areas, which are those in the upsampling part of the network, between the convolutional layers.

Even with these changes the network still overfit very rapidly despite extensive data augmentation. Finally, we found a way to prevent this from happening in the form of spatial dropout [3]. It is also called feature dropout, and it involves randomly dropping out entire feature maps instead of individual activations. This makes a lot more sense in a fully convolutional setting as ordinary dropout results is similar to dropping random pixels in intermediate image representations. We implemented this as a custom layer in Lasagne/Theano and added it to the “bridge” connections in the Unet (crossing the gap in the U shape).

As ground truth segmentations we draw spheres at the locations of the nodules with the radius equal to that of the annotated nodule radius. Only slices with at least one pixel being a nodule were presented to the network. To further deal with the class imbalance, the weight of the loss of all the true pixels in a batch was set to equal the weight of all the other pixels. Pixels outside of the segmentation mask after dilation with a 11x11 sphere kernel were assigned a zero weight in this weight map.

The candidates are extracted per probability map output of the Unet (in the x,y plane). First the slice is thresholded and then eroded with a + shaped kernel (3x3). The value to threshold at was determined on the validation subset. After erosion the candidates are grouped using connected components. The coordinates of the combined candidate equals to the center of mass of hits components. See figure 1 for example images of this process.

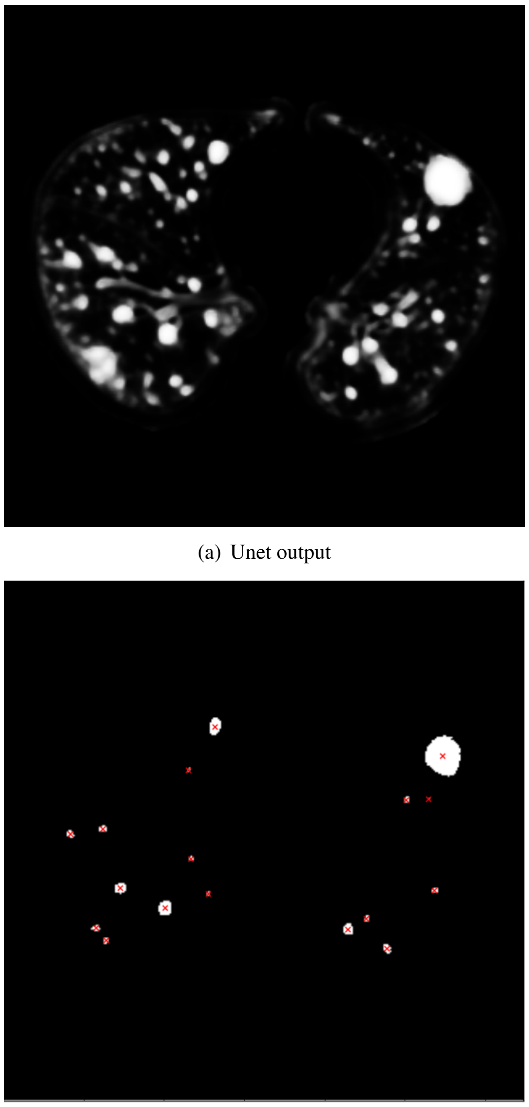
1.1.1. Candidate Merging

After the candidates are selected with Unet, various filtering and merging steps are applied in this specific order:

- Distance merging
- Nodule merging

Distance Merging

Because the candidates are extracted from a 2D environment, in the most probable case, it will find candidates in the third dimension very close to each other. In order to reduce these candidates, a distance-based merging algorithm is applied on the base candidates extracted from Unet output. The distance is euclidean and the merging is done on world-coordinates in millimeters. The candidates that are too close to one another are replaced by one candidate at the mean of these candidates.


(a) Unet output

(b) Candidates

Fig. 1. From Unet output to candidates (before merging). The above image (a) is the raw output from the Unet CNN, (b) shows this image after thresholding and erosion. The red crosses indicate the coordinates of each candidate, which is at the center of mass for every connected component.

TP- and FP-based merging
This merging step filters the candidates based on distance to true positive-candidates (nodule) within the provided annotations file. If a candidate is 'too close' to an annotation, it is discarded. Here, we throw away candidates that are on the very edge of a nodule (within 30mm from the center of the nodule). Candidates that are within the same annotation (within the radius of a nodule) are merged into one candidate by taking the mean coordinate.

Precision & Recall
In order to evaluate the performance of the approaches the 888 slides are divided into 10 subsets. Using these sets, a 10-fold cross validation can be performed. Here a method is developed and tweaked on 9 of the 10 subsets, the other subset is then predicted using the model trained on the first 9. This is repeated for all 10 subsets, until all subsets have been predicted.

Unet was trained and validated on 8 subsets at a time (so the other 2 splits could be predicted), hence it was a 5-fold cross validation. The final result of the candidate selection is a candidate set with a precision of **0.00509** and a recall of **0.95946**.

1.2. Residual Convolutional Neural Networks

For the false positive reduction step in the pipeline we make use of deep convolutional neural networks. As input data we fed our models with 2D images. For every candidate three different slices were created. One slice over the x and y axis, one slice over the x and z axis, and one slice of the y and z axis, with the candidate centered the middle of each slice. These planes have a size of 64 by 64 pixels. All these slice were given separately to a single network, there was no fusion within the network. The predicted output values of the network for these three different slices were ensembled in the end by taking the mean. This has a regularizing effect, one could view it as three separate networks with tied weights for all three orientations.

The scaling factor of all scans is known, and because having a scale invariant method is not a goal in this challenge, all scans were resampled such that the scaling factor was the same for all slices via linear interpolation. A uniform 0.5mm per voxel scaling was chosen in all dimensions. The 64x64 patches thus correspond to 32x32 millimeters in the real world.

We employed wide residual networks as proposed by Zagoruyko et al. [2]. They showed that wider and less deep residual networks outperform their deeper and thinner counterparts both in accuracy and efficiency. Also, they showed that in such an architecture - contrary to ordinary residual networks - dropout can be utilized to prevent overfitting [8]. Figure 2 provides an overview of the architecture. We used N=5, k=6 for the final submissions, resulting in a network with 32 convolutional layers.

group name	output size	block type = $B(3,3)$
conv1	64 x 64	[3×3, 16]
conv2	64 x 64	$\begin{bmatrix} 3\times3, 16\times k \\ 3\times3, 16\times k \end{bmatrix} \times N$
conv3	32 x 32	$\begin{bmatrix} 3\times3, 32\times k \\ 3\times3, 32\times k \end{bmatrix} \times N$
conv4	16 x 16	$\begin{bmatrix} 3\times3, 64\times k \\ 3\times3, 64\times k \end{bmatrix} \times N$
avg-pool	1 x 1	$[8 \times 8]$

Fig. 2. Structure of the wide residual networks. Network width is determined by factor k. Groups of convolutions are shown in brackets, where N is a number of blocks in group. The final softmax classification layer is omitted for clearance.

Xavier initialization [6] was used for weight initialization, ADAM as optimization method [7]. Leaky Rectified Linear Units were used as nonlinearities throughout the network.

1.3. Data augmentation

Data augmentation is a key aspect in improving the prediction scores since it acts as a regularizer and therefore prevents overfitting. For Unet and ResNet we used the data augmentations as shown in table 1. The augmentation values are randomly determined for each image every epoch between the ranges shown in the table.
Additionally to improve the test set scores, test time augmentation was used. For every candidate in the test set a number of images with different augmentations is returned, of which the predicted probabilities averaged. For the final results only flips were used due to time constraints, resulting in 4 images per original image. Empirically evaluated on one subset, it yielded an improvement of our average FROC score from 0.902 to 0.911.

Table 1. Augmentations used in the various networks, all with uniform probability distributions.

	Unet	**Wide ResNet**
Flip (axes)	X	X,Y
Rotation (degrees)	-20,20	-20,20
Zoom (factor)	0.9,1.1	0.9,1.1
Translation (pixels)	-3,3	-3,3
Gaussian noise (std)	0.05	-

1.4. Learning rate reduction

Reducing the learning rate over time can help settle into a nice local minimum instead of continuously stepping over it due to a too large update step. Following in the footsteps of the original authors of ResNet we applied a learning schedule where the learning rate is decreased by 90% after epoch 80 and epoch 125. We did not further optimize these values.

For UNet we apply a simple rule: if the loss had not decreased the last 6 epochs, the learning rate is decreased by 90%.

1.5. Prediction equalization

Simply concatenating the predictions of the models for the splits results in a poor FROC score at the false positive rates used for evaluation. This is due the difference in distributions of predictions for different models. One model may have a false positive rate of 1 per scan at decision boundary 0.9, whereas for another model this may be at 0.95. Calculating the score for the individual models we attain much higher scores than after simply merging the predictions.

To deal with this, a simple postprocessing step was applied to the output of the individual networks. For every model the 200th highest probability is determined, which is subtracted from all predictions, which are then divided by the highest probability.

2. IMPLEMENTATION AND HARDWARE

All convolutional networks were implemented using a combination of the Lasagne and Theano libraries[4][5] for Python, which allow for employing the GPU to train these networks. We trained on a large range of CUDA enabled graphics cards including the *Tesla K40M, Titan X, GTX 980, GTX 970, GTX 760* and the *GTX 950M* laptop graphics card. All were capable of training the rather modest wide ResNet incarnations, albeit with different batch sizes. For training the UNet architecture a graphics card memory size greater than 2GB is required.

The full implementation of our method is available online[1].

3. RESULTS AND DISCUSSION

The competition leaderboard score is the average FROC value at false positive rates of 1/8, 1/4, 1/2, 1, 2, 4 and 8 false positives per scan.

We trained a Wide ResNet with N and k values of 5 and 6, and achieved a performance of **0.812** on the competition leaderboard in the nodule detection track. A score of **0.758** was achieved in the false positive reduction track, which involves applying the false positive reduction network on a candidate set provided by the competition organizers.

We think that with a different fusion strategy for the information in the three slices per candidate this can be improved. For instance, encoding the three orientations in color channels of the input image of the network. Another possible approach

[1]https://www.github.com/gzuidhof/luna16

is a late fusion architecture with different branches for the input orientations.

Also, we feel that the processing of the output of the candidate selection CNN can be improved. A simple improvement could be using 3D connected components instead of 2D connected components on the different slices. This might make the merging by distance step obsolete.

4. REFERENCES

[1] http://luna16.grand-challenge.org/

[2] Zagoruyko, S., & Komodakis, N. (2016). Wide Residual Networks. arXiv preprint arXiv:1605.07146.

[3] Oliveira, G. L., Valada, A., Bollen, C., Burgard, W., & Brox, T. (2016). Deep Learning for Human Part Discovery in Images. In IEEE International Conference on Robotics and Automation (ICRA).

[4] Dieleman, S., Schlter, J., Raffel, C., Olson, E., Snderby, S. K., Nouri, D., & Kelly, J. (2015). Lasagne: First Release. Zenodo: Geneva, Switzerland. ISO 690

[5] Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., & Bengio, Y. (2010, June). Theano: A CPU and GPU math compiler in Python. In Proc. 9th Python in Science Conf (pp. 1-7).

[6] Glorot, X., & Bengio, Y. (2010, May). Understanding the difficulty of training deep feedforward neural networks. In Aistats (Vol. 9, pp. 249-256).

[7] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., & Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. arXiv preprint arXiv:1502.03044, 2(3), 5.

[8] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385.

[9] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 234-241). Springer International Publishing.