DataScienceBowl 2017

Lung cancer prediction

Team

Cowboy Bebop - 11th place

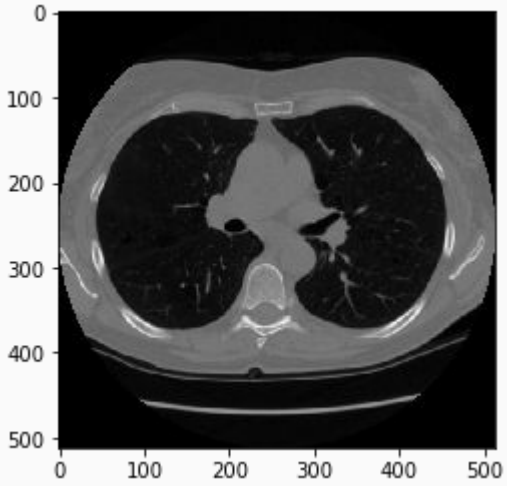
Dmitry Altuhov
Alexander Guschin
Dmitry Ulyanov
Michail Trofimov

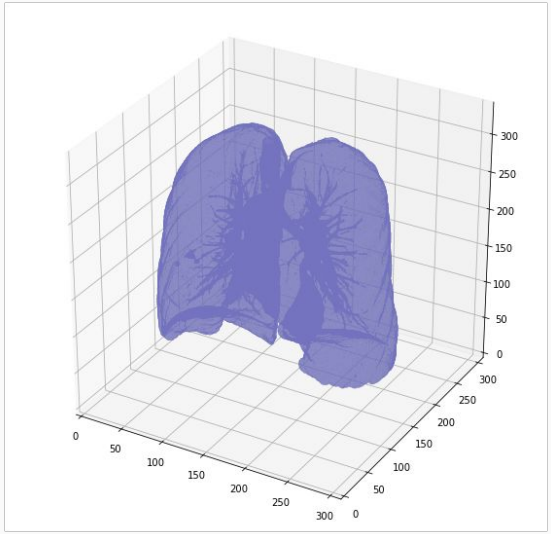
Overview

X: 3d images of lungs (CT)

Y: 1 if (lung cancer was diagnosed during 1 year after scan) else 0

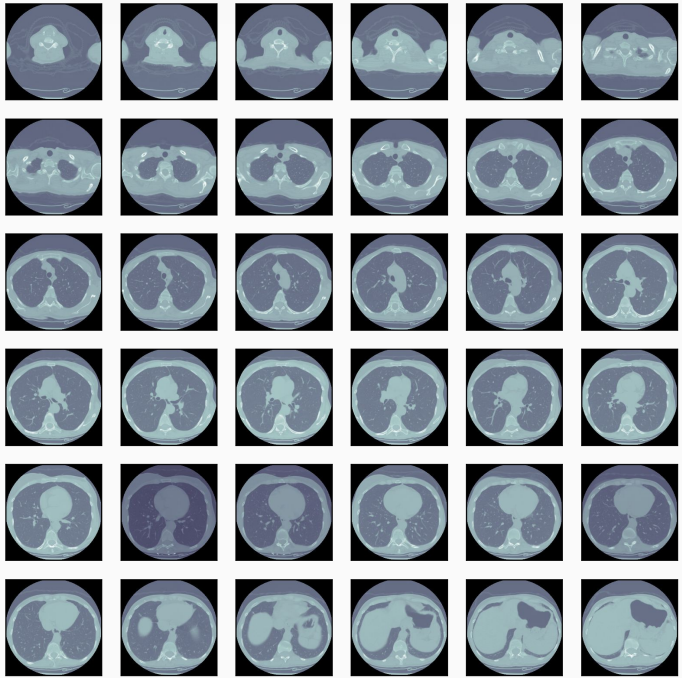
Metric: Logloss



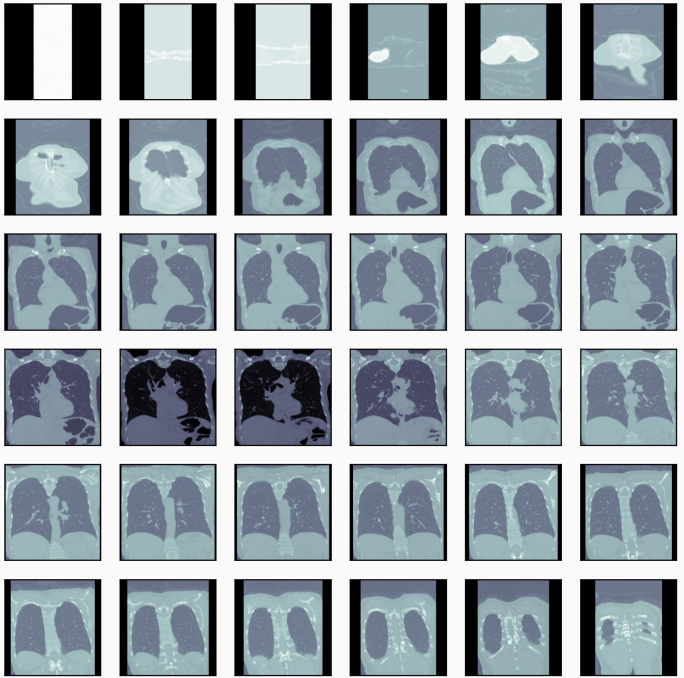


Example

(X,Y) slices of 3d image



(Z,Y) slices of 3d image



Overview

X: 3d images of lungs (CT)
Y: 1 if (lung cancer was diagnosed during 1 year after scan) else 0

Additional data: **Luna 2016 challenge**

1. 3d images of lungs (CT)
2. Nodules candidates (detected automatically)
3. Candidates assesment (by radiologists)

A candidate == (x,y,z)

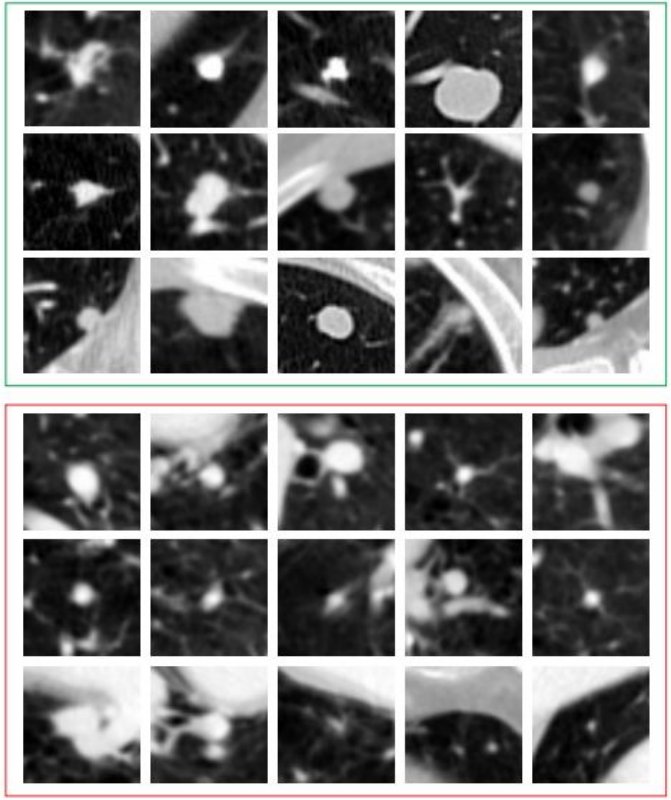
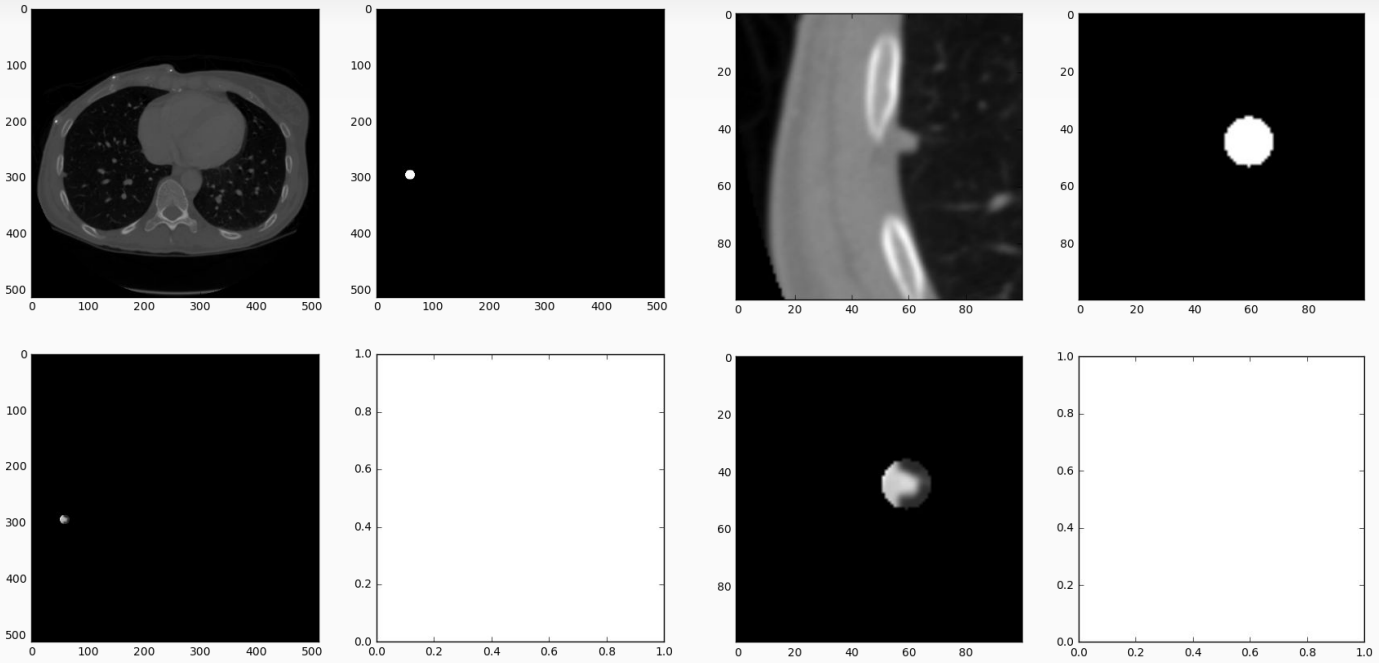


Fig. 1. Examples of the pulmonary nodules with various sizes, shapes and locations (green rectangle), and the false positive candidates (red rectangle) which carry similar appearance and make the task challenging. Each example is a representative 2D transverse plane extracted from a location.

Example: nodule



Data size

DataScienceBowl:

3d images of lungs (CT) x 1600 patients (stage1) ~ 120GB
3d images of lungs (CT) x 500 patients (stage2) ~ 60GB

Additional data: **Luna 2016 challenge**

3d images of lungs (CT) x 1000 patients ~ 80GB

Common pipeline

1. Preprocess data
    a. Rescale to 1 voxel == 1mm^3
    b. Refine data: segmentation
2. Train networks on Luna Dataset
    a. Classify candidate/annotation (3d convnet)
    b. Segmentation candidate/annotation (3d Unet)
    c. Classify nodule's malignancy
3. Use networks on DSB to "preprocess" data
    a. "Probability" map
4. Train
    a. 3d convnet
    b. Xgboost

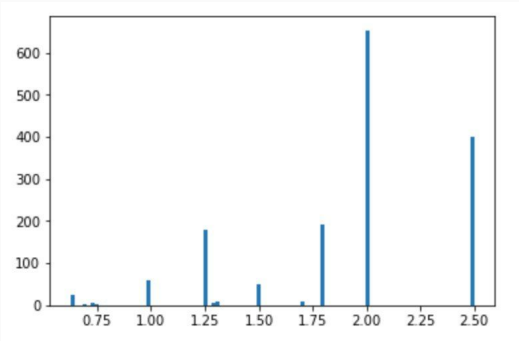
1. Preprocess: rescaling

Different spacing between slices

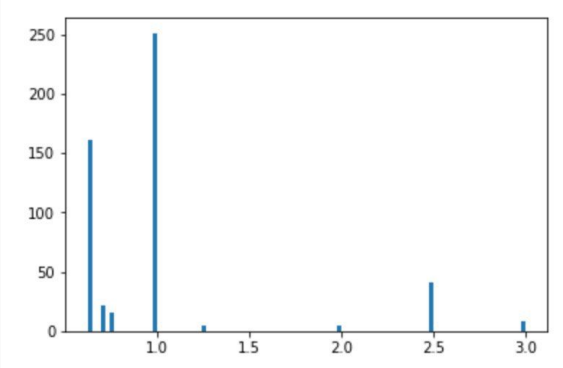
1. Between different patients in one dataset
2. Between different datasets

Stage2 have smaller spacings => higher quality data

Stage1:



Stage2:



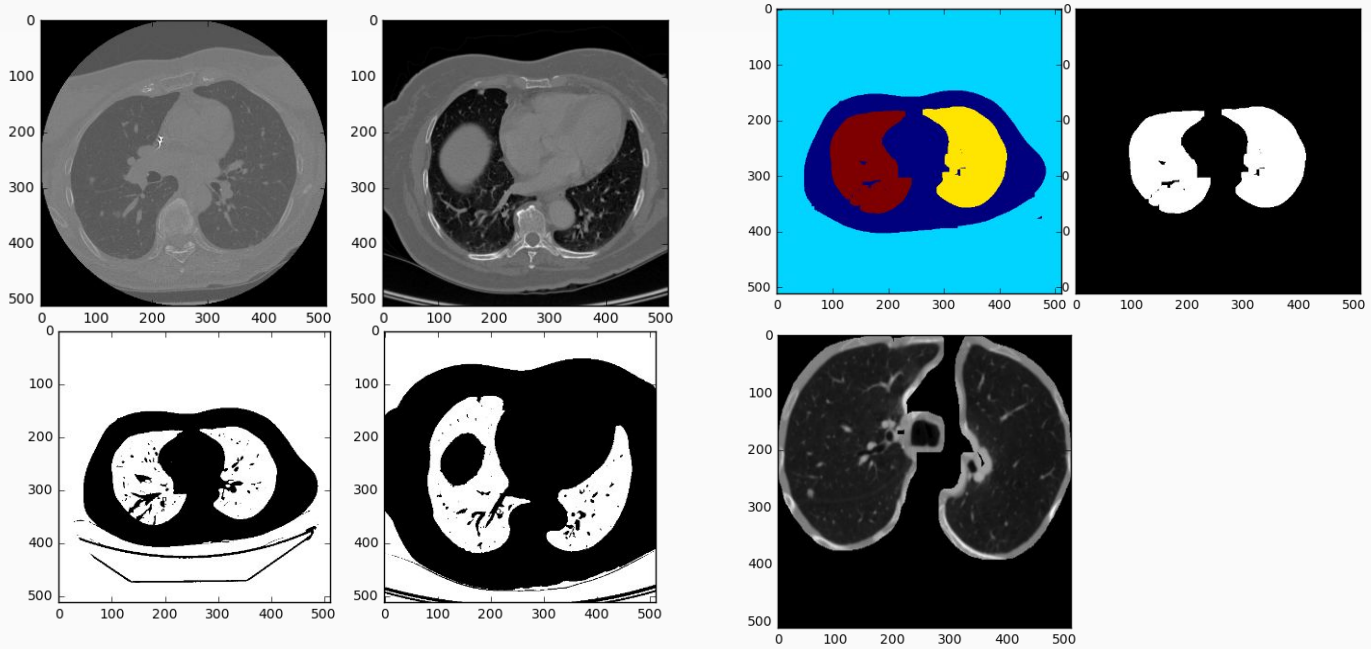
1. Preprocess: segmentation

Two goals:

1. Remove redundant parts
2. Refine useful parts

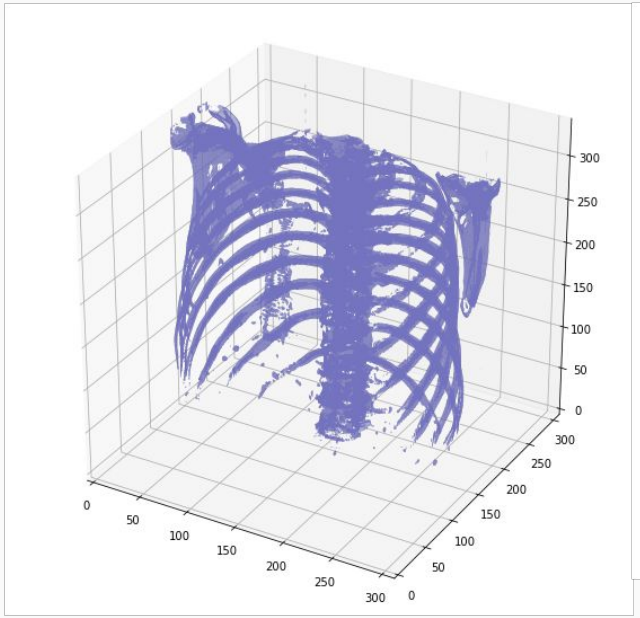
1. Preprocess: segmentation

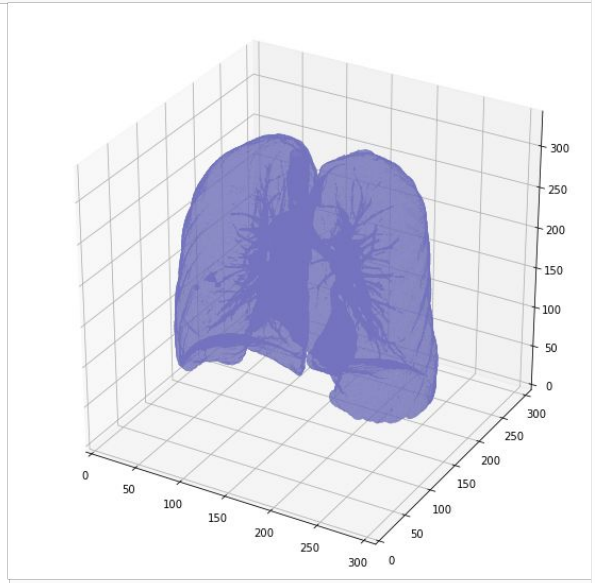
Examples

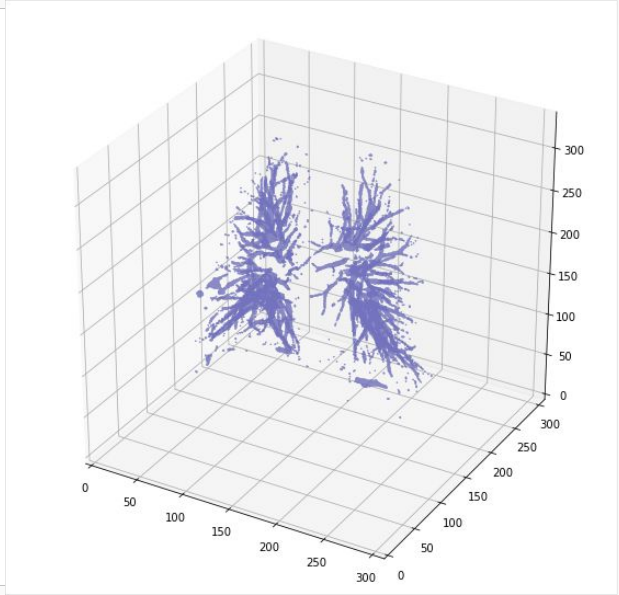


1. Preprocess: segmentation

Examples







2. Train networks on Luna dataset

Differences in network architecture. In input:

1. Different receptive fields: 32^3, 64^3, etc
2. Some use 2d conv networks (see 7th place report)

In target:

1. Classification: candidate/annotation or malignancy (XML)
2. Segmentation using 3d-contour (XML)
3. Regression: nodule size

2. Train networks on Luna

Multi-level Contextual 3D CNNs for False Positive Reduction in Pulmonary Nodule Detection

Qi Dou, *Student Member, IEEE,* Hao Chen, *Student Member, IEEE,* Lequan Yu, *Student Member, IEEE,* Jing Qin, *Member, IEEE,* and Pheng-Ann Heng, *Senior Member, IEEE.*

TABLE I
ARCHITECTURES OF THE MULTI-LEVEL CONTEXTUAL 3D CNNS.

Archi-1			Archi-2			Archi-3		
layer	kernel	channel	layer	kernel	channel	layer	kernel	channel
Input	-	1	Input	-	1	Input	-	1
C1	$5\times5\times3$	64	C1	$5\times5\times3$	64	C1	$5\times5\times3$	64
M1	$1\times1\times1$	64	M1	$2\times2\times1$	64	M1	$2\times2\times2$	64
C2	$5\times5\times3$	64	C2	$5\times5\times3$	64	C2	$5\times5\times3$	64
C3	$5\times5\times1$	64	C3	$5\times5\times3$	64	C3	$5\times5\times3$	64
FC1	-	150	FC1	-	250	FC1	-	250
FC2	-	2	FC2	-	2	FC2	-	2
Softmax	-	2	Softmax	-	2	Sofrmax	-	2

C: convolution, M: max-pooling, FC: fully-connected

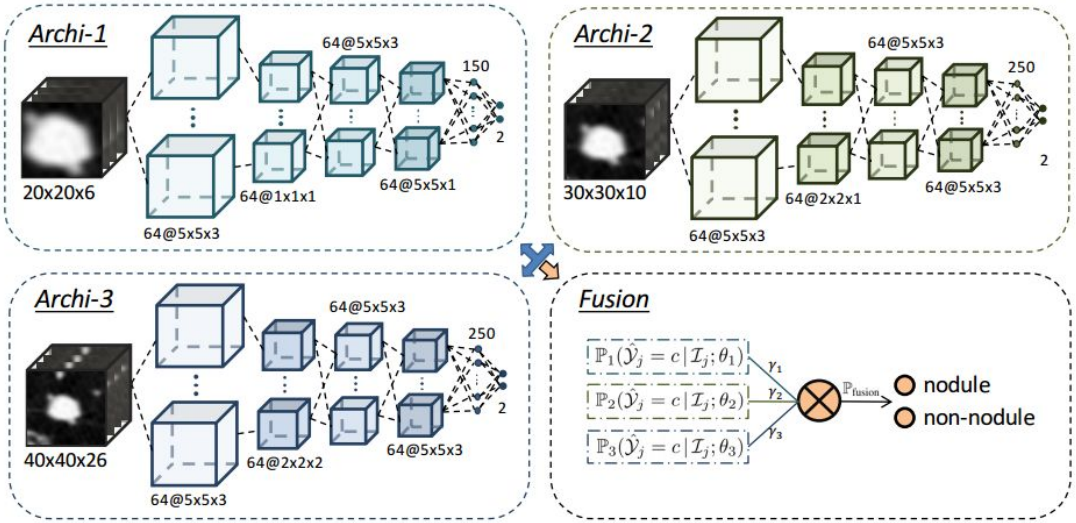


Fig. 2. The framework of the proposed method. We design three 3D convolutional networks incorporating different levels of contextual information. The posterior predictions of these networks are fused to produce the final classification result.

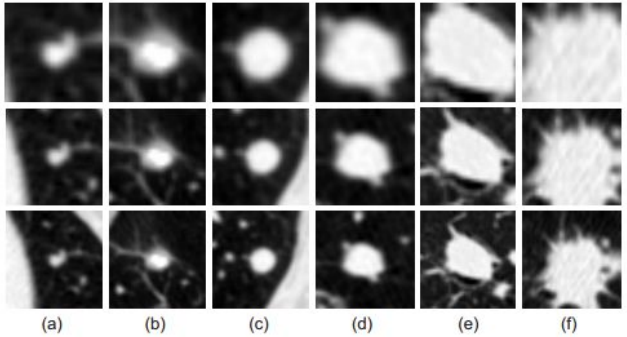
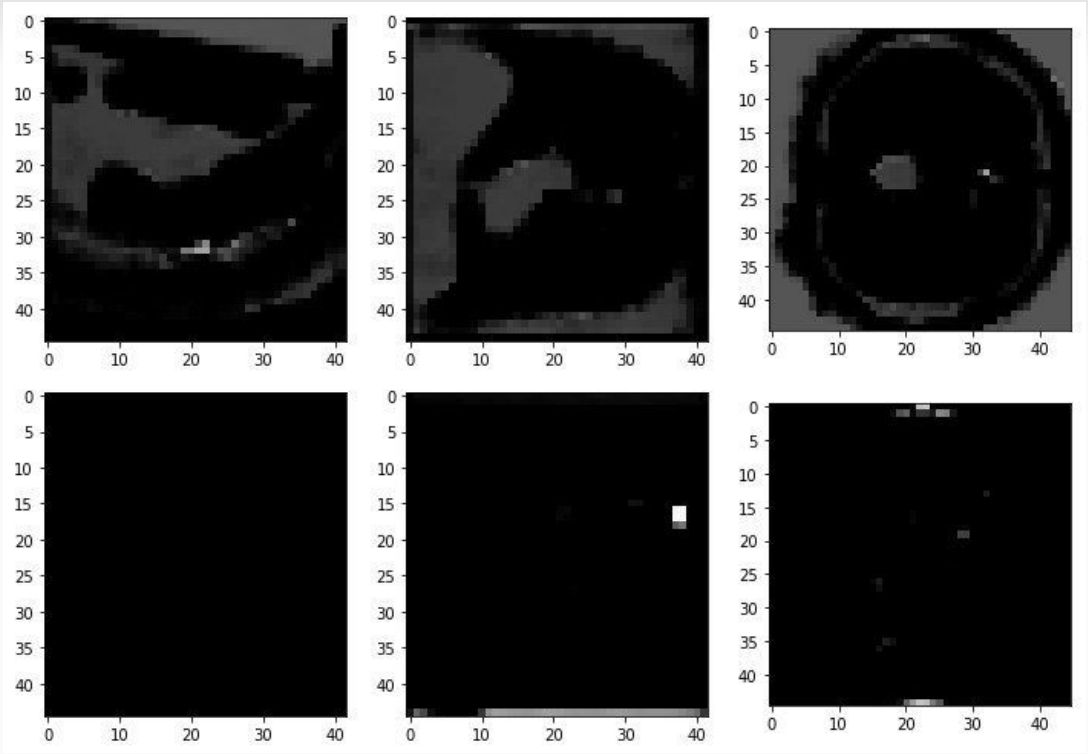


Fig. 4. Illustration of multi-level contextual information surrounding nodules. The patch sizes are $20\times20\times6$, $30\times30\times10$ and $40\times40\times26$ for the first, second and third row, respectively. We show the transverse plane only, and all patches are scaled to the same image resolution for clear visualization. The examples (a) and (b) are small nodules with diameter lower than 7 mm, (c-e) are middle sized nodules with diameter between $9\sim16$ mm, (f) is a large nodule with a diameter of over 24 mm.

3. Use networks on DSB to “preprocess” data

After “preprocessing”



4. Train on DSB after “processing”

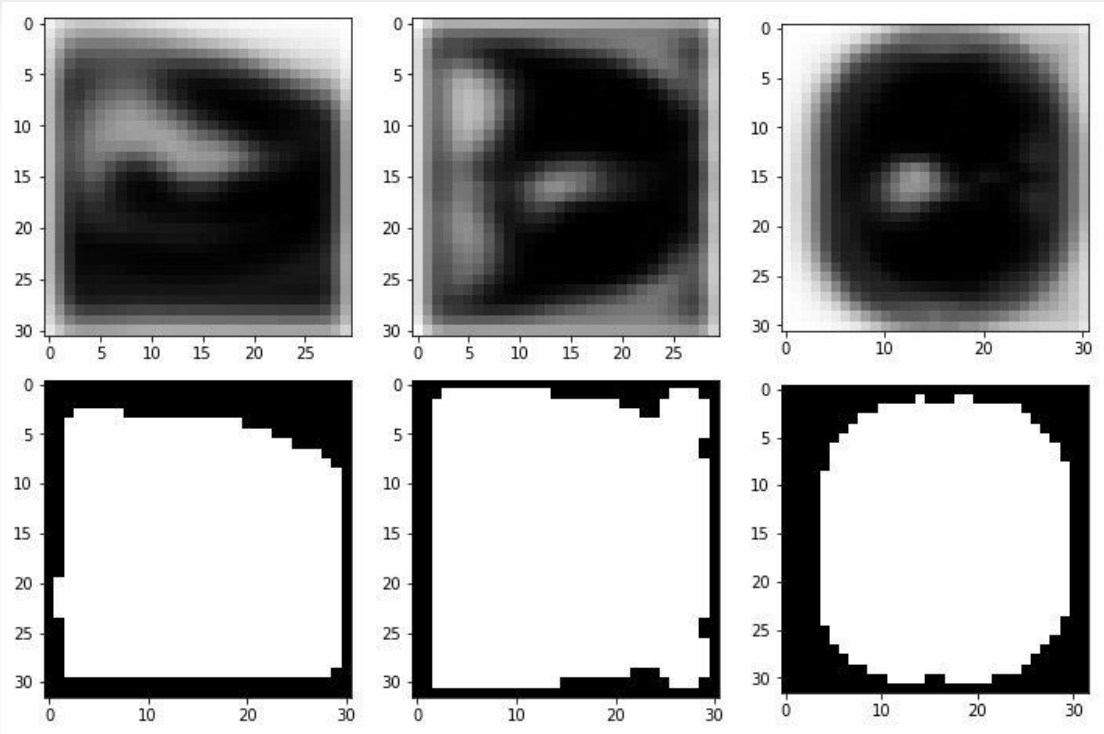
1. 3d-Convnet
2. Xgboost
    a. Features: max, np.where(patch == patch.max()), relative coordinates
    ~ 0.405 cv (5 folds), ~0.44 lb

    b. Preprocessing: mask
    ~ 0.405 cv (5 folds), ~0.435 lb

    c. Features for predictions from 3 zoom levels (2nd place)
    1 voxel == 1mm^3, 1.5mm^3, 2mm^3

4. Train on DSB after “preprocessing”

Use mask to clip redundant parts of images



Pipeline overview: 2nd and 11th places

1. Preprocess data
   a. Rescale to 1 voxel == 1mm^3
   b. Refine data: segmentation
2. Train networks on Luna Dataset
   a. Classify candidate/annotation (3d convnet)
   b. Segmentation candidate/annotation (3d Unet)
   c. Classify nodule's malignancy
3. Use networks on DSB to "preprocess" data
   a. "Probability" map
4. Train
   a. 3d convnet
   b. Xgboost

Pipeline overview: 7th place

1. Preprocess data
   a. Rescale to 1 voxel == 1mm^3
   b. Refine data: segmentation
2. Train networks on Luna Dataset
   a. Classify candidate/annotation (3d convnet)
   b. Segmentation candidate/annotation (**2d** Unet)
   c. Classify nodule's malignancy
3. Use networks on DSB to "preprocess" data
   a. "Probability" map **+ Clustering**
4. **Train**
   a. **Top-20 most suspicious clusters > 3d convnet classification (labels from patients)**
   b. **Max probability from top-20 > patient prediction**

Pipeline overview: 9th place

1. Preprocess data
   a. Rescale to 1 voxel == 1mm^3
   b. Refine data: segmentation
2. Train networks on Luna Dataset
   a. Classify candidate/annotation (3d convnet)
   b. Segmentation candidate/annotation (**3d** Unet)
   c. Classify nodule's malignancy
3. Use networks on DSB to "preprocess" data
   a. Find candidates by 2b > False positive reduction by 2a and 2c
4. **Train**
   a. Top-N most suspicious nodules > 3d convnet classification (by networks like 2a/2c)
   b. Aggregating probabilities from top-N > patient prediction

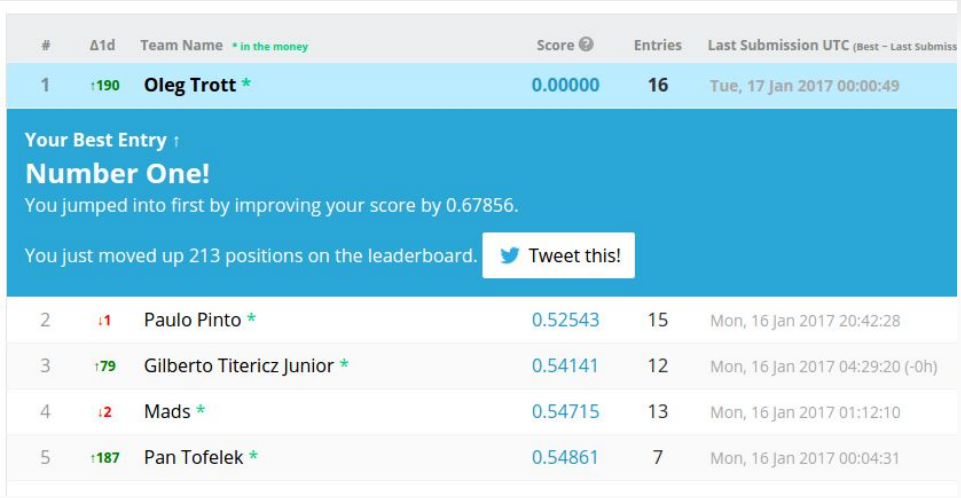
Technical details

The heaviest network trains up to 12 hours on 4x Nvidia M60 for networks on Luna dataset

Pytorch

Fun moments

Perfect score script <Oleg Trott>



#	Δ1d	Team Name * in the money	Score	Entries	Last Submission UTC (Best – Last Submiss
1	↑190	**Oleg Trott** *	0.00000	16	Tue, 17 Jan 2017 00:00:49

Your Best Entry ↑

Number One!

You jumped into first by improving your score by 0.67856.

You just moved up 213 positions on the leaderboard. Tweet this!

2	↓1	Paulo Pinto *	0.52543	15	Mon, 16 Jan 2017 20:42:28
3	↑79	Gilberto Titericz Junior *	0.54141	12	Mon, 16 Jan 2017 04:29:20 (-0h)
4	↓2	Mads *	0.54715	13	Mon, 16 Jan 2017 01:12:10
5	↑187	Pan Tofelek *	0.54861	7	Mon, 16 Jan 2017 00:04:31

The core algebraic insight needed here is that if we choose 15 probabilities to be

sigmoid(- n * epsilon * 2 ** i)

where n=198, 0 <= i < 15, and epsilon = 1.05e-5 for example, and choose the rest of the probabilities to be 0.5, then the 15 labels corresponding to those 15 probabilities are easily discoverable from the score we get, because all 32768 possible label combinations lead to different scores.

Fun moments

Using “pretrained” convolutions from Luna winners paper

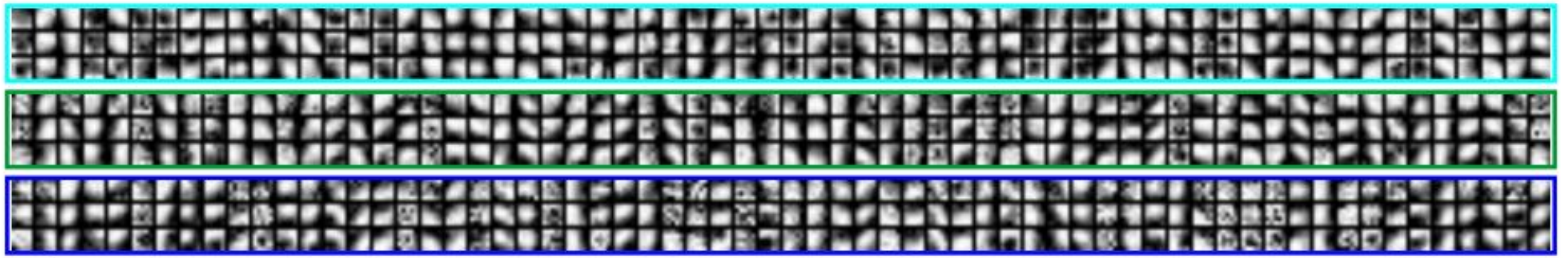


Fig. 7. Visualization of the learned 3D kernels in the first layer of the networks incorporating different levels of contextual information. Each $5 \times 5 \times 3$ kernel is embedded as three 5×5 maps presented in a column. The rectangles with color cyan, green and blue correspond to *Archi-1*, *Archi-2* and *Archi-3*, respectively.

Pictures are taken both from our work and from competition forum/kernels:

https://www.kaggle.com/c/data-science-bowl-2017