Solution Description

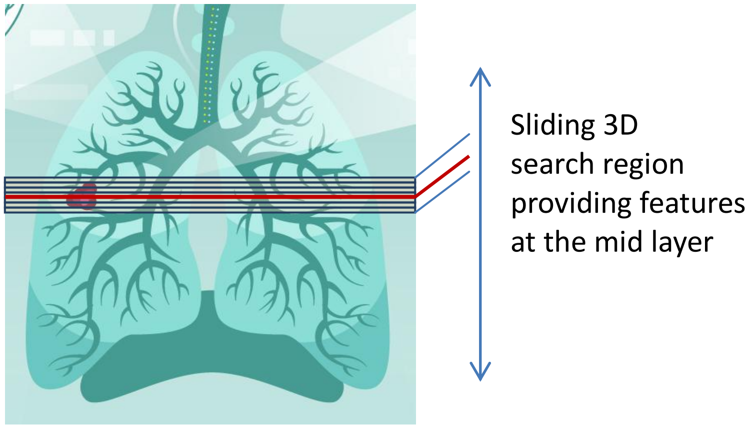
8th Place Solution to Data Science Bowl 2017 Kaggle Competition

Team: Alex|Andre|Gilberto|Shize

1. Summary

In this competition, we are provided with well over a thousand low-dose CT images from high-risk patients, and are challenged to predict whether or not the patient will be diagnosed with lung cancer within one year of the scan being taken.

In our approach, one essential factor is the development of a successful method to identify nodules in the lungs and extract relevant markers or features from the nodules, and the lungs in general, in order to be able to predict lung cancer risk. A sliding 3D data model was custom built to reflect how radiologists review lung CT scans to diagnose cancer risk.

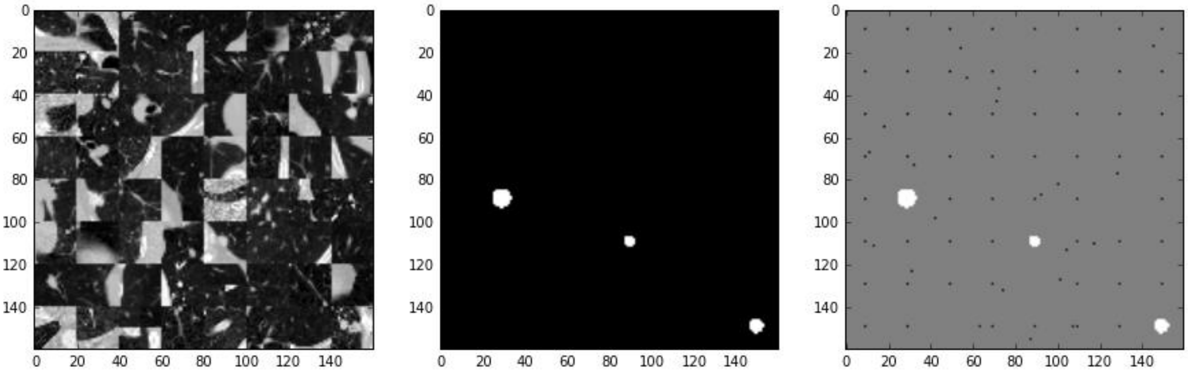


As part of this data model - which allows for any nodule to be analyzed multiple times - a neural network nodule identifier has been implemented and trained using the Luna CT dataset. Non-traditional, unsegmented (i.e. full CT scans) were used for training, in order to ensure no nodules, in particular those on the lung perimeter are missed. Various features were extracted from the individual nodules found by the identifier as well as from the segmented lungs as a whole. Finally, these features have been fed into a tree-based machine learning model to predict whether a person will be diagnosed as having cancer within 1 year of the CT scan being taken.

The key tools included, Keras and Theano for Convolutional Neural Network (with custom extension for the sliding data model) and tree based classifications tools (XGBoost and extraTree). Training some of the nodule models took days using high end 12GB GPUs. Once the features are extracted, we train a couple of xgboost models and extraTree models on different subsets of these features.

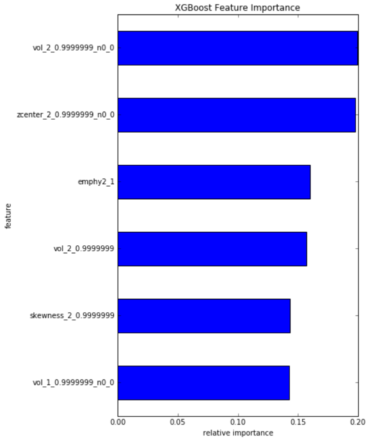
2. Feature Selection / Engineering

Feature engineering (using neural networks):
The most important feature is the existence of nodule(s), followed by their size, location and their other characteristics. For instance, a very significant number of patients for which no nodule has been found, proved to be no cancer cases.
Thus the accuracy of the nodule identifier was essential, and thus special mosaic-type aggregation of training of the nodule identifier has been deployed, as illustrated below.



Mosaic aggregation: 1. Scans (mid layer), 2. Masks, 3. Masks with false positives shown

Manual selection of features was used in the early stages of the competition, as shown on the graph below illustrating some of the key features created.



Key features include existence/size of the largest nodule, and its vertical location, existence of emphysema, volume of all nodules, and their diversity

One of a useful feature transformation was to code the location of nodules versus the segmented lungs centre of gravity, as this reflected well the fact that nodules in the upper part of lungs have higher risk of being cancerous. This approach seems to provide higher significance than just tracking whether nodules are in the upper or lower parts of lungs (as used by some professionals). For the final model, automatic selection of the most useful features offered by the tree based algorithms was used, notably XGBoost.

The key external source of data was the Luna CT dataset. No information on malignancy of nodules has been used, as this did not seem available – an assumption that proved incorrect, to the team's disadvantage (it seems that the top teams have found and used information on which nodules, including in the Luna database, are malignant).

3. Training Methods

With the three extracted feature sets ready, we then train a couple of xgboost and extraTree models on different subsets of the three extracted feature sets.

A significant attempt was also made to use a neural network end-to-end to generate the final predictions and submissions, but this approach was worse than our combined approach submitted, potentially due to the lack of information on the malignancy of nodules, or insufficient training. As outlined, our combined approach uses the neural network as a feature generator and then applying xgboost and extraTree models on the extracted features to generate predictions and submissions.

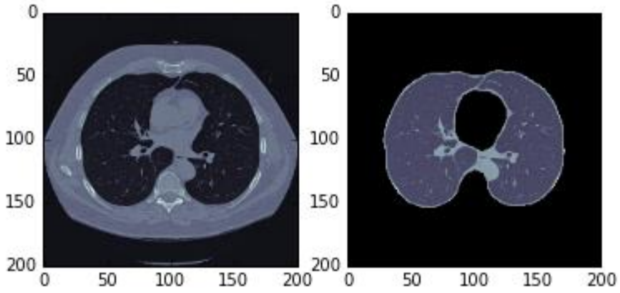
To make the model performance more stable, we also run some of the models with multiple random seeds (e.g., for xgb, use 50 random runs; for extraTree, use 10 random runs) and take the average. Our final winning submission (private LB0.430) is a linear combination of a couple of xgb models and extraTree models.

4. Code Descriptions

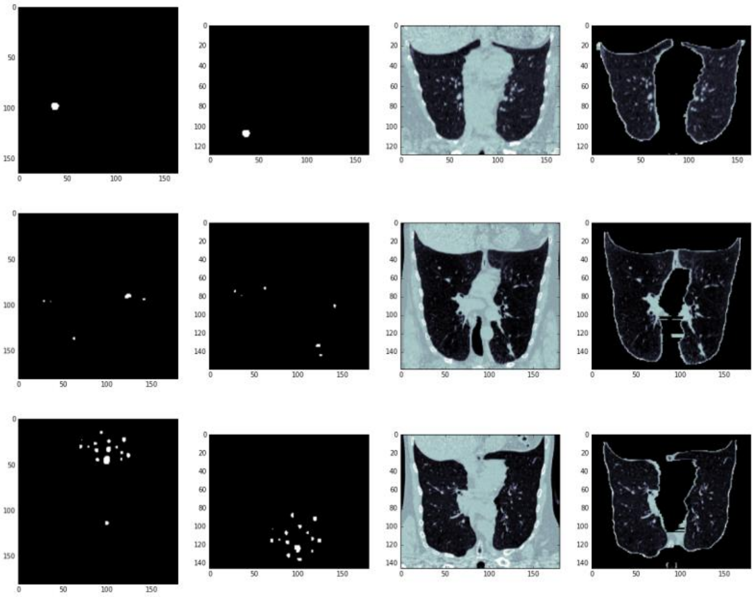
Codes under folder “code/ Andre”

These are codes used to generate the three extracted feature sets by neural networks on the raw competition data and LUNA data. These generated three feature sets that will be later used by the codes under folder “code/Shize” to generate xgb and extraTree model predictions and finally the winning ensemble submission. The solution is very data processing intensive and generally comprises of:

1. Train a nodule identifier on a slicing architecture using Luna dataset or intermediate files created (3 options provided)
2. Resample all patient CT scans, convert them to Hounsfield units, and segment the lungs, as illustrated (mid section view shown)



3. Identify the nodules for each patient using the trained nodule identifier (see the examples below for two cancer and one no cancer patients, illustrating the projections of nodules, and cross sections of lungs and their segmentation created for quality control and visual checks at this step)



4. Extract variety features from the nodules found and lungs in general, and create 3 feature sets for predictions

For further details and to see how to generate the feature sets from the raw data using the code please check the “ReadMe.txt” under the folder.

Codes under folder “code/ Shize”

These are codes used to generate the individual xgboost and extraTree models using the three extracted feature sets by neural networks on the competition image data and LUNA image data. It also includes the ensemble scripts to generate our final winning submission using these individual xgb and extraTree models. Please check the “00ReadMe.txt” under the folder to see how to easily generate the results using the code.

5. Dependencies

The codes under folder **“code/Andre”**

The codes under this folder were run both locally on i7 based Linux systems with 8GB GPUs and on Amazon Web Service P2 systems (with 12GB GPUs). The following packages are needed (all open software) Keras 1.2.2, Theano, Python3, conda, spyder and several popular related packages (e.g. pydicom, cv2, scipy, simpleitk, numpy, and pandas, and Linux environment).

The data dependency is the Luna dataset available at: https://luna16.grand-challenge.org/home/ (10 subsets, and the CSVFILES)

One challenge was significant size of the files and models generated (circa 1 Terabyte).

The codes under folder **“code/Shize”**

All the codes under this folder were run on an Amazon Web Service Linux C3.8 machine. Python2 and the standard popular related python packages (e.g., numpy, pandas, xgboost, sklearn, etc.) are needed.

6. How to Generate the Solution

Step 1: Following the “ReadMe.txt” under the folder “code/andre” to prepare the significant amount of data needed and subsequently generate the three different feature sets by using deep neural networks on the raw competition data and raw LUNA data. Three different options are provided, including a pre-trained nodule identifier that may be used for the fastest way to generate the three feature sets required for Step 2.

Step 2: Copy the three extracted feature sets from Step 1 to the folder "code/Shize". Then follow the "00ReadMe.txt" under the folder "code/Shize" to generate all those individual xgboost and extraTree models by using the three extracted feature sets from Step 1 as input. Then use the ensemble script under "code/Shize" to generate our winning ensemble which is a linear combination of the individual prediction files of these xgboost and extraTree models.

7. Simpler Model

In our winning ensemble, we use a linear combination of quite a couple of different xgb and extraTree models, and some of them were using averaged prediction from 50 or 10 random runs (i.e., using 50 (or 10) different random seeds). So the whole ensemble is a bit complex.

However, we find that a single xgb (to be specific, 0Shize_DSB_feat3_xgb_v5.py) on the extracted three feature sets (this single xgb is also a component in our winning ensemble) is actually enough to give us 8th place (it will score about 0.434 on private LB) standing on the private LB without using any ensemble. This might work as a much simpler solution that has comparable performance.

8. Additional Comments

One key factor to our success in this competition is the development of a new effective approach to extract powerful feature sets from the raw competition data and LUNA data by using neural networks. With this approach using a relatively rough resolution of 2x2x2mm, and 8 layers vertically, was sufficient. A more granular version with resolution of 2x1x1mm, and 16 layers vertically, has also performed well but was not included in submission.
Based on our experience in this competition, it is interesting to see that using neural network end-to-end to generate the submissions doesn't perform as well as using the neural networks as a feature extractor, and then feeding the extracted features to other types of models such as xgboost and extraTree to generate the predictions. At least in this competition, using xgboost and extraTree to generate the final submissions perform much better than using the neural network end-to-end. It is possible that maybe this is simply due to the fact that it will be more time consuming or tricky to tune the neural network classifiers to perform similarly well, or because we have not used information found by some other top teams on malignancy of nodules, or because the location of nodules is an important feature not easily dealt with by neural networks. It is also possible that using other types of models like xgboost or extraTree are more powerful on dealing with those

well processed feature sets than the neural network, even when training is transferred between different neural network workers.

Finally, we appreciate Kaggle and the sponsors to provide such a very challenging and interesting competition.