CSIT 6910 A Independent Project

Lung Cancer Detection

31st May 2027

Lujian – jluan@connect.ust.hk

Supervisor: CHUNG, Albert Chi Chung

Table of Contents

Abstract 1

Introduction 1

Data Set Description 2

Data Pre-processing 3

Detection Algorithm 4

Lung Region Extraction 4

Bit-Plane Slicing 4

Lung Border Extraction 7

Flood Fill algorithm and extracted lung 7

Segmentation of Lung Region 8

Mean Shift 8

Generate nodule candidates 9

Analysis of Lung Region 10

Feature Extraction 10

Formulation of Diagnosis rules 12

Training of Model 13

Trees and Forests 13

Running a Random Forest 14

Strengths and weaknesses 14

Model Testing and Evaluation 15

Cross-validation 15

Mean Precision 16

Result 16

Conclusion 17

Reference 17

Abstract

This study aims to highlight the significance of data analytics and machine learning in prognosis in health sciences, particularly in detecting life threatening and terminal diseases like cancer. Here, we consider lung cancer for our study. For this purpose, preexisting lung cancer patients' data are collected to get the desired results. A predictive algorithm is developed to predict the probability of a patient catching lung cancer based on dataset comes from the Data Science Bowl 2017. Data set (in the form of diagnostic images) is run past Matlab for analysis and forecasting. Image processing is employed for this purpose. Medical image segmentation and classification are done to achieve this. Classification depends on features extracted from the images. The emphasis is on the feature extraction stage to yield better classification performance. This information is then fed to machine learning algorithms to discern a pattern that can give some good insights into what combination of features are most likely to result in an abnormality.
KEYWORDS: prognosis, Matlab, CT, feature extraction, machine learning

Introduction

Lung cancer is one of the most common types of cancer, with nearly 225,000 new cases of the disease expected in the U.S. in 2016. Early detection is critical, as it opens a range of treatment options not available when cancer is detected at later, more advanced stages. Low-dose computed tomography (CT) is a potential breakthrough technology for early detection, with the ability to reduce deaths by 20%. Often, suspicious lesions identified in screening are initially assessed as high risk of cancer, but after additional follow-up tests, they turn out to be non-cancerous (false positives from the initial screening). Hopefully, machine learning can reduce the number of radiology exams flagged for potentially unnecessary follow up and avoid patient anxiety. Using a data set of high-resolution scans of lungs provided by the National Cancer Institute, researchers can develop artificial intelligence algorithms to accurately determine when lesions in the lungs are cancerous. This will dramatically reduce the false positive rate that prevents low-dose CT scans from being widely used for lung cancer detection.

For this project, the lung cancer detection system is shown in figure 1. This project initially preprocesses the dataset in order to filter useless data and gets a smaller data set. Next, different image processing techniques is applied such as Bit-plane Slicing, Erosion, Median Filter, Dilation, Outlining, Lung Border Extraction and Flood-Fill algorithms for extraction of lung region [1]. Then for segmentation Mean Shift algorithm is used and for learning and classification Random Forest is used.

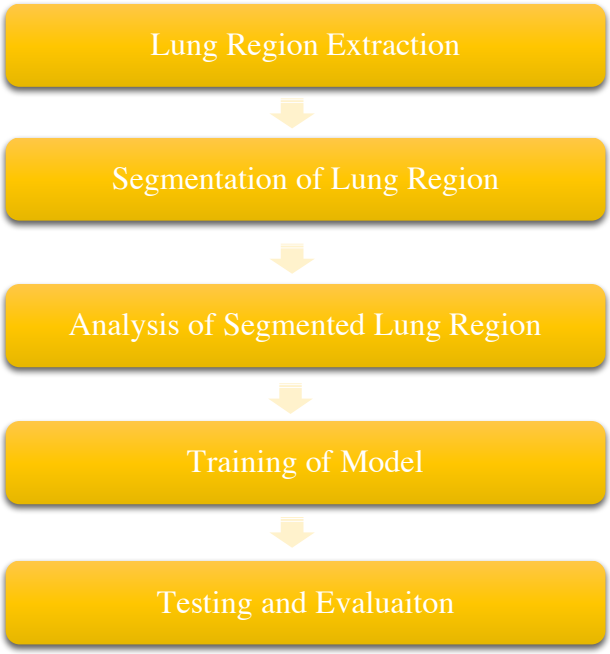

Figure 1. The Lung Cancer Detection System

Data Set Description

For this project I used one main source of data. The major of data I used came from Kaggle's contest. I will briefly describe the data set I found useful.

The full file list & information can be found at:
https://www.kaggle.com/c/data-science-bowl-2017/data

Kaggle contest's file
stage1.7z

This file contains all images for the first stage of the completion, including both training and test set. The training set and test set contain about 1500 patients' data and for each patient there are about 120 CT scan slices.

stage1_labels.csv

This file contains the cancer ground truth for the stage 1 training set images.

stage1_solution.csv

This file contains the caner ground truth for the stage 1 test set images.

Data Pre-processing

The raw data downloaded from kaggle is quite large (about 150 Gbyte) and it contains a lot of useless data. For data pre-processing, after reading the dcm format lung scan image into Matlab I count the dark intensity in the specific lung region and compute a percentage threshold. With the threshold, I can filter a number of useless CT slices and finally pick 10 slices for each patient. As Figure 2 shown, we do not want lung scans like CT3 but prefer lung scans link CT1, CT2 and CT3. After shrinking the data set, I obtain a smaller data set (about 4 Gbyte) that can be processed in the next stage.

Figure 2. Some scan slices are useless

Detection Algorithm

Lung Region Extraction

The initial stage of the proposed Computer Aided Diagnosing (CAD) [2, 7] techniques is the extraction of lung region from the CT scan image. The basic image processing techniques are utilized for this purpose. The methods and steps involved in the extraction of lung region from CT image are shown in figure 3. The image processing techniques applied in the proposed technique are Bit-Plane Slicing, Erosion, Median Filter, Dilation, Outlining, Lung Border Extraction and Flood-Fill algorithms. Usually, the CT chest image not only contains the lung region, it also contains background, heart, liver and other organs areas. The main aim of this lung region extraction process is to detect the lung region and regions of interest (ROIs) from the CT scan image.

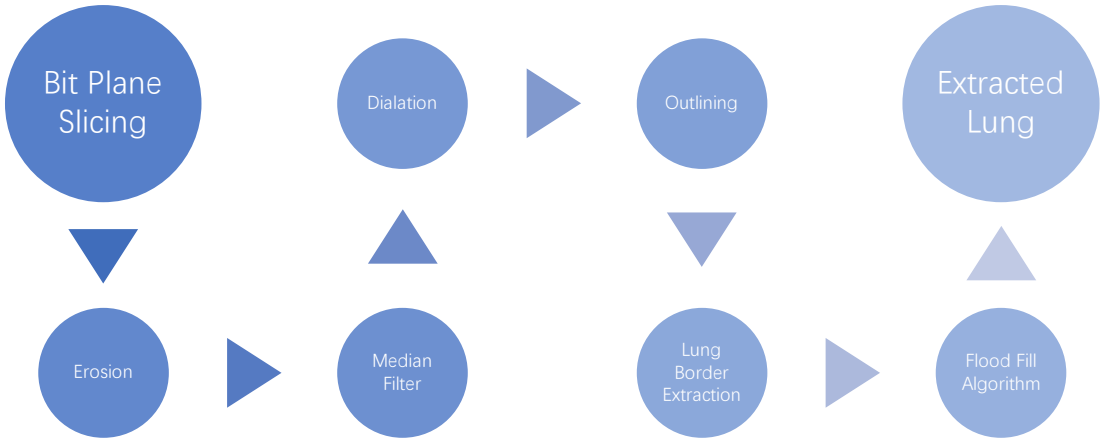

Figure 3. Lung Region Extraction Process

Bit-Plane Slicing

The databases comprise a set of 2D-CT images of normal and abnormal subjects. Each image is stored as a double subscripted array of type unsigned integer (8 bits) and of size 512 x 512 pixels, and each element of the array contains the CT-intensity value of a single pixel in a plane resolution 0.78 x 0.78, slice thickness 2 mm. the 512x512 pixel slices were acquired in the upper, middle, and lower part of the lung of each subject. Figure 4 shows sample of the acquired data from the same subject at different levels

from top to bottom. The eight bit plane representations of the slice at a specific level are given in Figure 5.

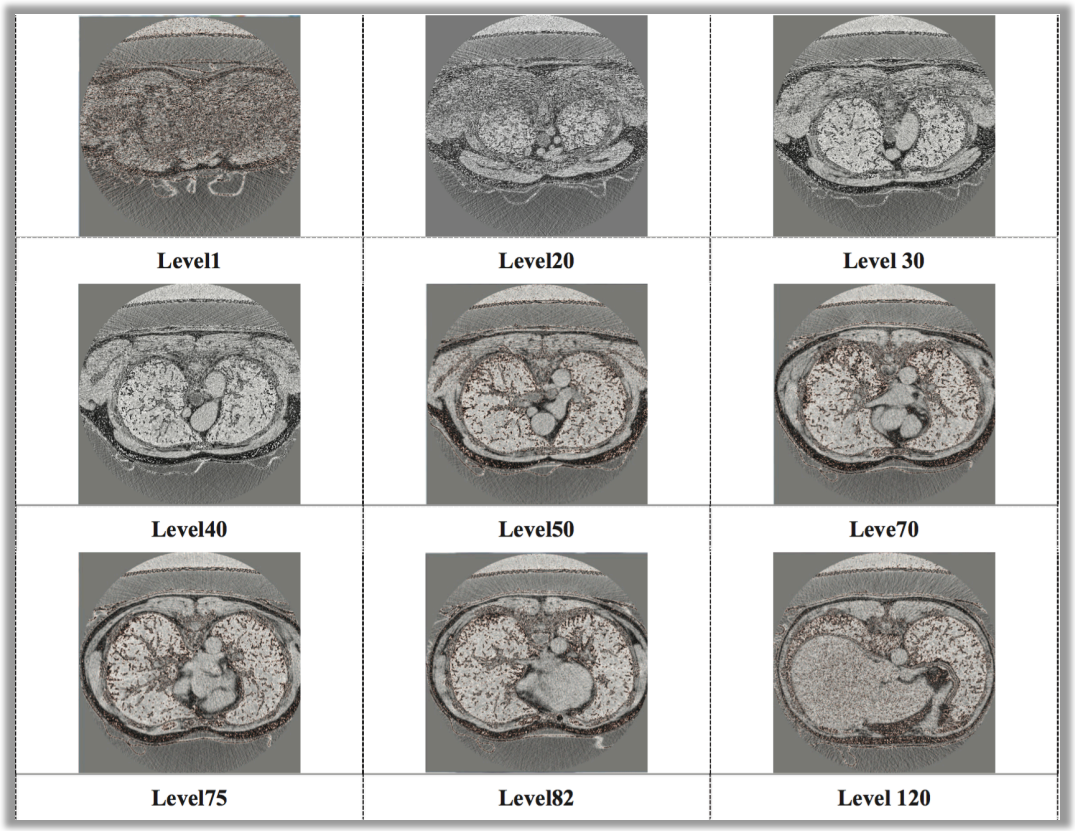

Figure 4. shows samples of the acquired data from the same subject at different levels from top to bottom.

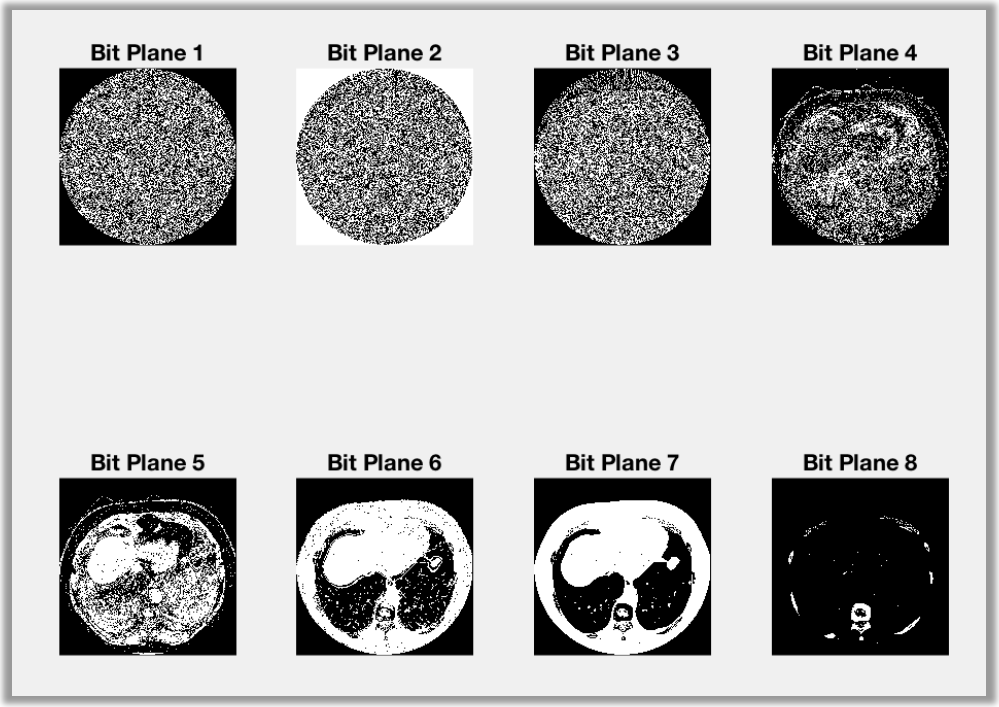

Figure 5. Bit Plane Slicing

The extraction of lung region starts by applying the bit-plane slicing algorithm to the slimed data set from step one. The resulted binary slices are then analyze to choose from among then, the best bit-plane image that may help in extracting automatically the lung region from raw data with a certain degree of accuracy and sharpness.

In my previous work, I have tried that the method [1] declared in the paper gives its best results when used with bit plane7 or bit plane8. Using other bit-planes such as bit-plane5 and bit-plane6 ends with some kinds of problems that were eliminated when using bit-plane7 except the loss of some parts of lung region when interconnected with some other tissue regions on the wall of the lung. However, this fixed choice of one of the bit-plane method may goes wrong when it encounters the problem of image quality variance. In that, with more observation to the information represented in the Bit Planes, I generate 5 modified candidate Bit Planes and choose the best one from them. As Figure 6 shown, the generated method and generated results are presented.

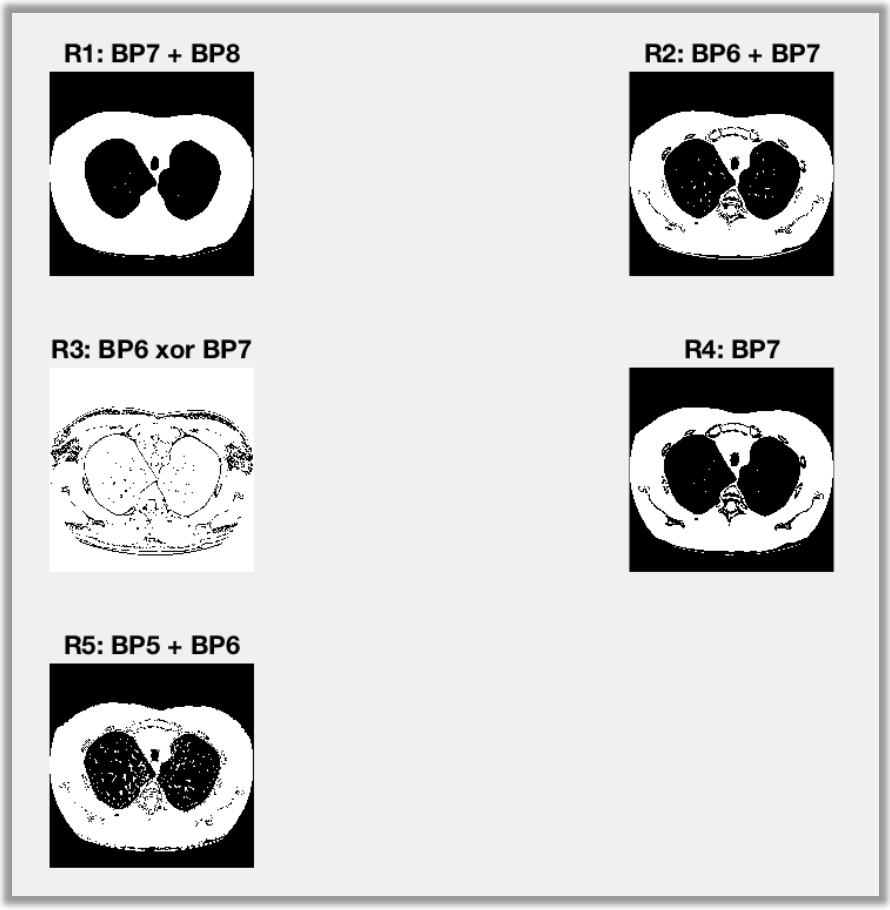

Figure 6. Generated five modified bit planes candidates

The selection strategy is developed baes on two aspects. The first one is to compute the contrast ratio in a scope of lung region. The second one is to compute the ratio of cleanliness of the image, which it means that the cleaner generated bit plane is preferred to be selected. Using Figure 6 as an example, the contrast ratio of R1, R2, R4, R5 will be almost the same but smaller that R3, which it means R3 is discarded. The cleanliness ratio of R2, R4, R5 will be a little bit smaller than R1, which it means R1 having a greater chance to be selected.

Lung Border Extraction

The lung border extraction is based on the result of applying outlining algorithm images. In Matlab, I use `bwboundaries` function to outline the objects in the input images. The `bwboundaries` function implements the Moore-Neighbor tracing algorithm modified by Jacob's stopping criteria. This function is based on the boundaries function presented in the first edition of Digital Image Processing Using MATLAB, by Gonzalez, R. C., R. E. Woods, and S. L. Eddins, New Jersey, Pearson Prentice Hall, 2004. As Figure 7 shown, I need to delete useless outlines and finally get a result like the right one. Based on the return value of `bwboundaries` function, I know the number of points of each object in the outlining image. Thus I can set a threshold to filter the small circles region. Plus, I can filter the outer ring encircling the two lungs by designed detecting algorithm.

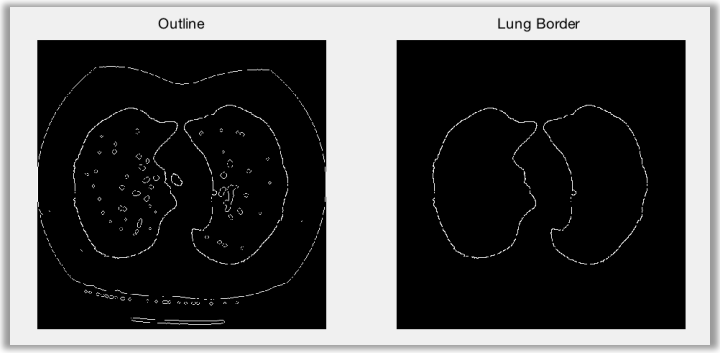

Figure 7. The result of lung border extraction

Flood Fill algorithm and extracted lung

Finally, flood fill algorithm is applied to fill the obtained lung border with the lung region. After applying these algorithms, the lung region is extracted from the CT scan

image. This obtained lung region is further used for segmentation in order to detect the cancer nodule.

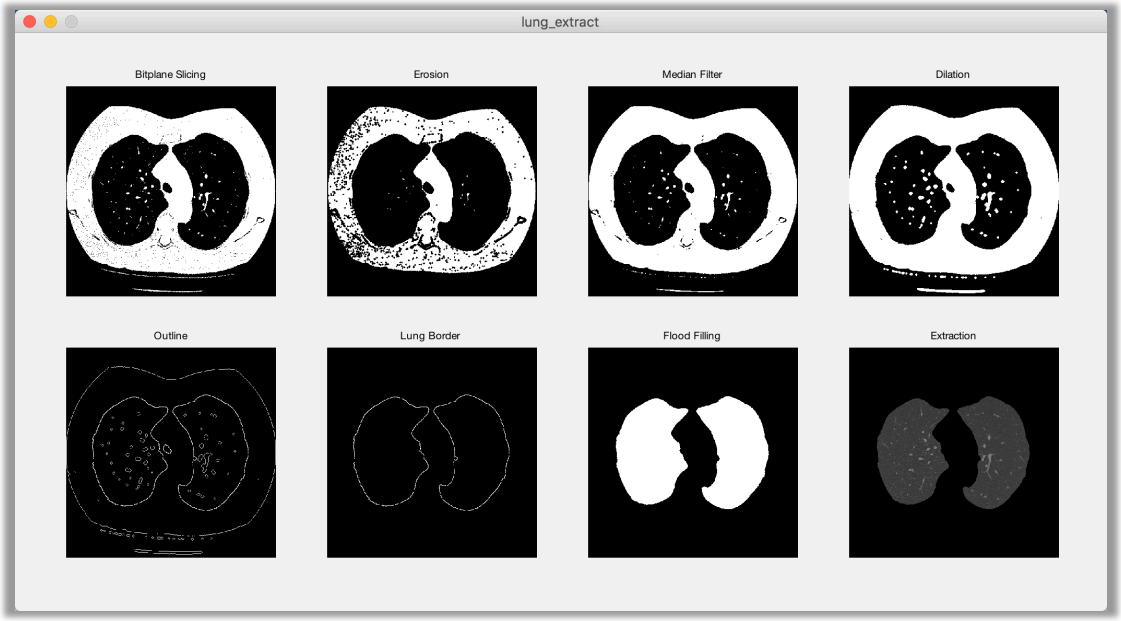

Figure 8. Lung region extraction algorithm

Segmentation of Lung Region

After the lung region is detected, the next process is segmentation of lung region in order to find the cancer nodules. This step will identify the region of interest (ROIs) which helps in determining the cancer region. In this project, Mean Shift is implemented for segmentation.

Mean Shift

Introduction

Mean Shift is a powerful and versatile non parametric iterative algorithm that can be used for lot of purposes like finding modes, clustering etc. Mean Shift was introduced in Fukunaga and Hostetler [14] and has been extended to be applicable in other fields like Computer Vision. This document will provide a discussion of Mean Shift, prove its convergence and slightly discuss its important applications.

Intuitive Idea

Mean shift considers feature space as an empirical probability density function. If the input is a set of points then Mean shift considers them as sampled from the underlying

probability density function. If dense regions (or clusters) are present in the feature space, then they correspond to the mode (or local maxima) of the probability density function. We can also identify clusters associated with the given mode using Mean Shift. For each data point, Mean shift associates it with the nearby peak of the dataset's probability density function. For each data point, Mean shift defines a window around it and computes the mean of the data point . Then it shifts the center of the window to the mean and repeats the algorithm till it converges. After each iteration, we can consider that the window shifts to a more denser region of the dataset.

At the high level, we can specify Mean Shift as follows :

1. Fix a window around each data point.
2. Compute the mean of data within the window.
3. Shift the window to the mean and repeat till convergence.

Kernel Density Estimation

Kernel density estimation is a non-parametric way to estimate the density function of a random variable. This is usually called as the Parzen window technique. Given a kernel K, bandwidth parameter h , Kernel density estimator for a given set of d-dimensional points is

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

Generate nodule candidates

With the input of lung region extraction images, Mean Shift is applied to cluster nodule candidates and the result is shown in Figure 9. After the segmentation is performed to the lung region, the feature extraction and cancer diagnosis can be performed with the segmented image.

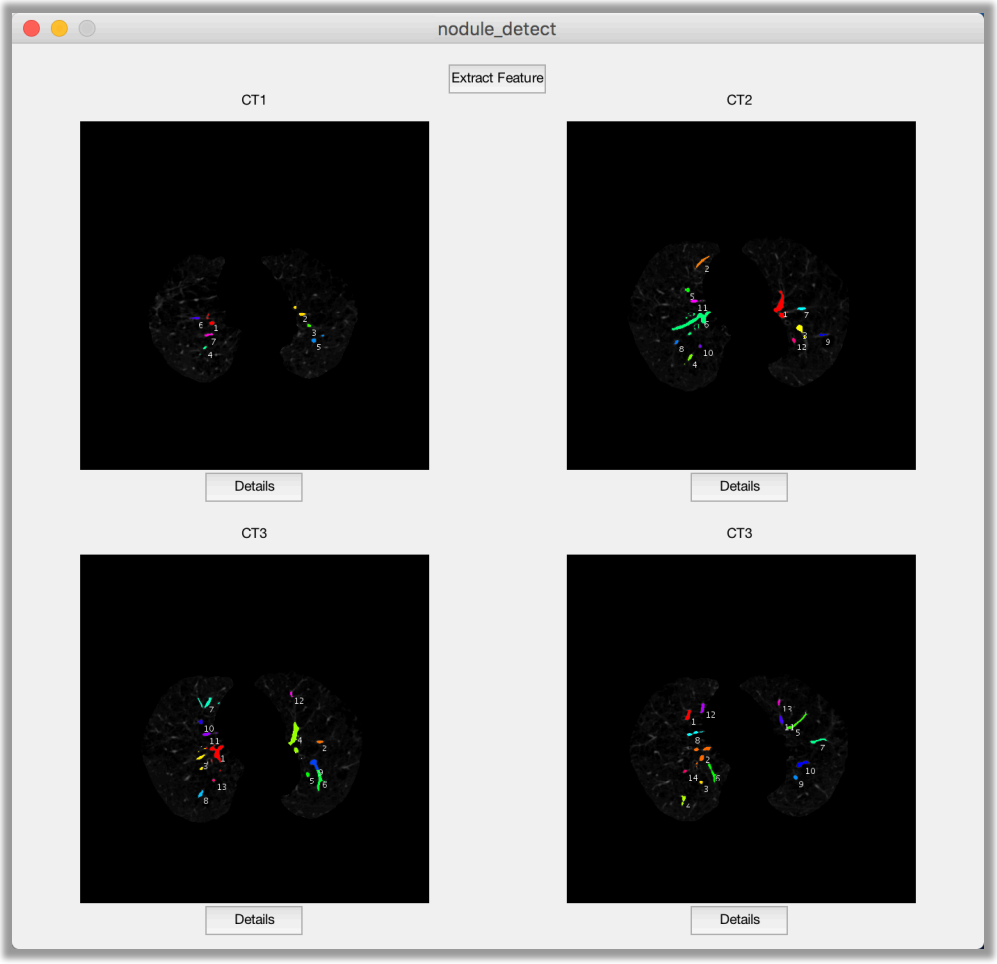

Figure 9. Nodule candidates generated result

As Figure 9 shown, each lung region extraction image can generate from ten to thirty nodule candidates. Consequently, each patient generates more than one hundred nodules candidates, which they needed to be filter since not all of them are useful.

Analysis of Lung Region

After the segmentation is performed to the lung region, the features can be obtained from it and the diagnosis can be designed to exactly detect the cancer nodules in the lungs. The diagnosis rules can eliminate the false detection of cancer nodules resulted in segmentation and provides better diagnosis.

Feature Extraction

The features that are used in this project in order to generate diagnosis rules are:

- Area of the candidate region
- The Maximum Drawable Circle (MDC) inside the candidate region

- Mean intensity percentage of candidate region

Area of the candidate region

This feature can be used here in order to

- Eliminate isolated pixels
- Eliminate very small candidate object

With the help of this feature, the detected region that do not have the chance to form cancer nodule are detected and can be eliminated. This helps in reducing the processing in the further steps and also reducing times taken by further steps.

Maximum Drawable Circle (MDC)

This feature is used to indicate the candidate regions with its maximum drawable circle (MDC). All the pixels inside the candidate region is considered as center point for drawing the circle. The obtained circle within the region is taken for consideration. Initially radius of the circle is chosen as one pixel and then the radius is incremented by one pixel every time until no circle can be drawn with that radius. Maximum drawable circle helps in the diagnostic procedure to remove more and more false positive cancerous candidates.

Mean intensity percentage of the candidate region

In this feature, the mean intensity percentage for the candidate region is calculated which helps in rejecting the further regions which does not indicate cancer nodule. The mean intensity percentage indicates the average intensity percentage of all the pixels that belong to the same region and is calculated using the formula:

$$\text{MeanPtg(j)} = \frac{\sum_{i=1}^{n} Intensity(i)/n}{Max}$$

where j characterizes the region index and range from the number of candidate regions in the whole image, which we specify as N. Intensity(i) indicates the CT intensity value of pixel i, and i ranges from 1 to n, where n is the total number of pixels belonging to region j. Max indicates the maximum intensity values of all of the candidate regions.

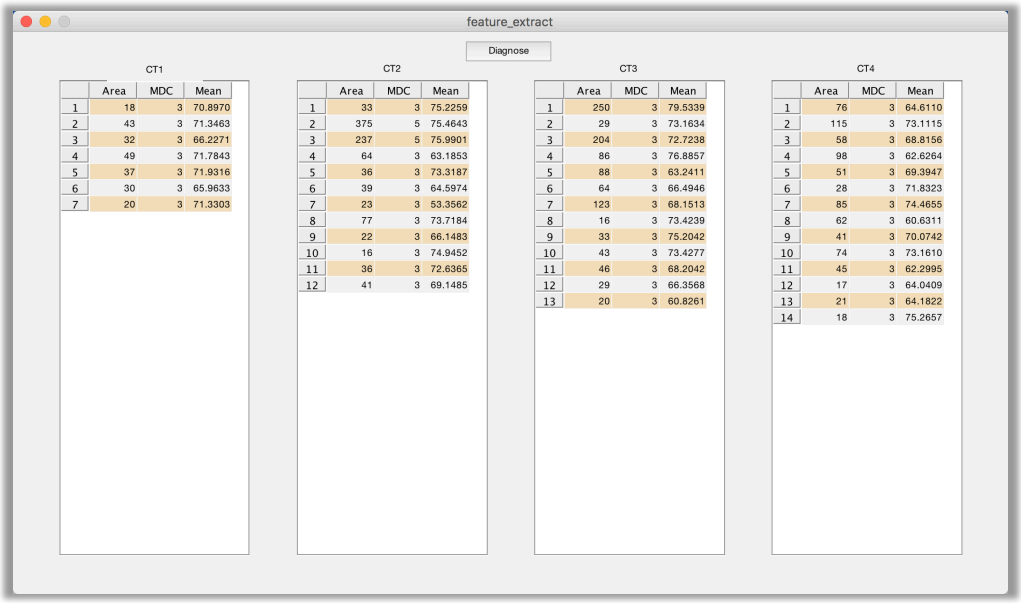

Figure 10. Feature extraction result

Formulation of Diagnosis rules

After the necessary features are extracted, the following diagnosis rules can be applied to detect the occurrence of cancer nodule. There are three rules which are involved are as follows:

Rule 1: Initially the threshold value T1 is set for area of region. If the area of candidate region is lower than the threshold value, then it is eliminated for further consideration. This rule will helps in reducing the steps and time necessary for the upcoming steps.

Rule 2: In this rule maximum drawable circle (MDC) is considered. The threshold T2 is defined for value of maximum drawable circle (MDC). If the radius of the drawable circle for the candidate region is less than the threshold T2, then that is region is considered as non-cancerous nodule and is eliminated for further consideration. Applying this rule has the effect of rejecting large number of vessels, which in general have a thin oblong, or line shape.

Rule 3: In this, the rage of value T3 and T4 are set as threshold for the mean intensity percentage of candidate region. Then the mean intensity percentage for the candidate regions are calculated. If the mean intensity percentage of candidate region goes below minimum threshold or goes beyond maximum threshold, then that region is assumed as non-cancerous region.

By implementing all the above rules, the maximum of regions which does not considered as cancerous nodules are eliminated. The remaining candidate regions are considered as cancerous regions. This CAD system helps in neglecting all the false positive cancer regions and helps in detecting the cancer regions more accurately. These rules can be passed to the Random Forest Classifier in order to detect the cancer nodules for the supplied lung image.

Training of Model

In this training section, I use Random Forest model to train and test the data.

Trees and Forests

The random forest starts with a standard machine learning technique called 'Decision Tree' which, in ensemble terms, corresponds to the weak learners. In a decision tree, an input is entered at the top and as it traverses down the tree the data gets bucketed into smaller and smaller sets. The random forest takes this notion to the next level by combining trees with the notion of ensemble. Thus, the ensemble terms, the trees are weak learners and random forest is a strong learner.

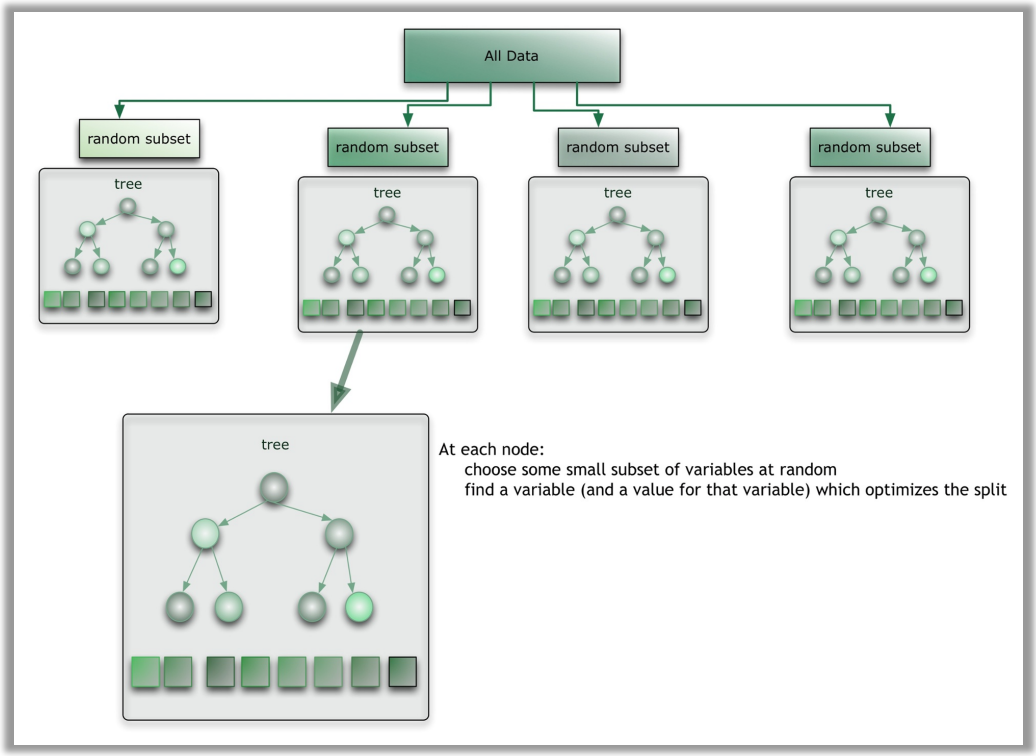

Figure 11. Random forest model description

Here is how such a system is trained; for some number of trees *T*:

1. Sample *N* cases at random with replacement to create a subset of the data (see top layer of figure above). The subset should be about 66% of the total set.
2. At each node:
    a) For some number *m* (see below), *m* predictor variables are selected at random from all the predictor variables.
    b) The predictor variable that provides the best split, according to some objective function, is used to do a binary split on that node.
    c) At the next node, choose another *m* variables at random from all predictor variables and do the same.

Depending upon the value of *m*, there are three slightly different systems:

- Random splitter selection: $m = 1$
- Breiman's bagger: *m* = total number of predictor variables
- Random forest: $m \ll$ number of predictor variables. Brieman suggests three possible values for m: $\frac{1}{2}\sqrt{m}$, $\sqrt{m}$, and $2\sqrt{m}$

Running a Random Forest

When a new input is entered into the system, it is run down all of the trees. The result may either be an average or weighted average of all of the terminal nodes that are reached, or, in the case of categorical variables, a voting majority.

- With a large number of predictors, the eligible predictor set will be quite different from node to node.
- The greater the inter-tree correlation, the greater the random forest error rate, so one pressure on the model is to have the trees as uncorrelated as possible.
- As *m* goes down, both inter-tree correlation and the strength of individual trees go down. So some optimal value of *m* must be discovered.

Strengths and weaknesses

Random forest runtimes are quite fast, and they are able to deal with unbalanced and missing data. Random Forest weaknesses are that when used for regression they cannot predict beyond the range in the training data, and that they may over-fit data sets that are particularly noisy. Of course, the best test of any algorithm is how well it works upon your own data set.

Model Testing and Evaluation

In this this project, I evaluate the random forest model with two methods:

- Cross validation
- Mean precision

Cross-validation

We test our classifier using a technique called “cross-validation”: train the classifier on all projects except for one. Here the projects mean the partition of the data set. Of course, we also know the ground truth for this held-out project, so we can see how well the classifier does on it, without cheating by training the classifier on the hold-out. We do this in turn with each project. See figure below.

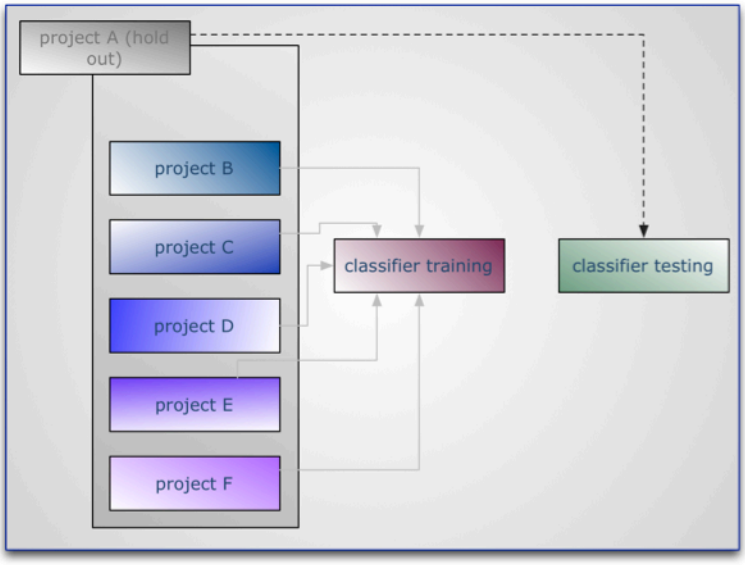

Figure 12. Cross validation

Let's say we have projects A through F. We train a classifier on projects A through E, and test on F. Then we train on A and C through F, and test on B. We do this in turn, holding out each project, until we train on A through E and test on F. Each project is a partition of our data set in our experiment, with subjects A through F.

In this project, I will use `fitensemble` function to generate an ensemble training model. And compare its classification error rate with the normal one I apply.

Mean Precision

The effectiveness of the classifier is the distance between the two means, which does not vary as threshold changes. One way of measuring the effectiveness of the classifier is the "precision". Precision is the number of truly correct items ("hits") divided by the number of items that the classifier says are correct (hits + false alarms). "Mean precision" takes into account the issues with choosing a threshold, noted above, by performing this calculation at a range of thresholds and taking the mean.

In Matlab, it offers several function helps in evaluating the performance of the training model, which I get help from `kfoldLoss`, `oobLoss`, `loss`, `classperf` functions.

Result

As figure 13 shown, the following result presents a 10-fold cross-validated bagged ensemble and we examine the cross-validation loss as a function of the number of trees in the ensemble. With the number of trees increased by 50, the cross-validation model's classification error rate remains steady at 26%. While the normal random forest model's error rate fluctuates in the area between 26% and 28% when the number of trees increased from forty.

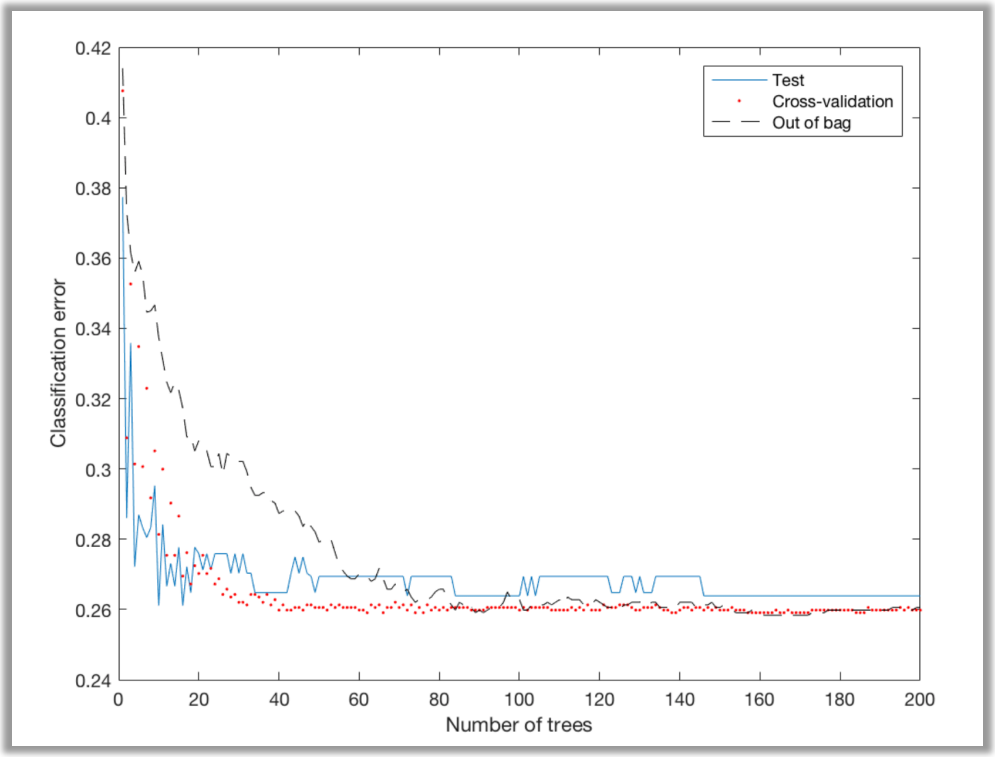

Figure 13. Cross validation and normal random forest model

The diagram also shows us that the mean accuracy of the normal random forest model is about 73%. With the help of trained model, we can compute a posterior probability of lung cancer for a patient automatically based on the detection algorithm depicted before.

Conclusion

This project presents the better Computer Aided Diagnosing (CAD) system for automatic detection of lung cancer. The initial process is lung region detection by applying basic image processing techniques such as Bit-Plane Slicing, Erosion, Median Filter, Dilation, Outlining, Lung Border Extraction and Flood-Fill algorithms to the CT scan images. After the lung region is detected, the segmentation is carried out with the help of Mean Shift clustering algorithm. With these, the features are extracted and the diagnosis rules are generated. These rules are then used for learning with the help of Random Forest. The experimentation is performed with 15, 000 images obtained from the kaggle contest. The experimental result shows that the proposed CAD system can able to tell the posterior probability of lung cancer for a patient based on the detection algorithm. Also the usage of Random Forest will increase the accuracy of detecting the cancer nodules.

Reference

[1] M.Gomathi, Dr.P.Thangarj, "A Computer Aided Diagonsis System For Detection of Lung Caner Nodules Using Extreme Learning Machine", ISSN:0975-5462, Vol. 2(10), 2010

[2] R. Wiemker, P. Rogalla, T. Blaffert, D. Sifri, O. Hay, Y. Srinivas and R. Truyen "Computer-aided detection (CAD) and volumetry of pulmonary nodules on high-resolution CT data", 2003.

[3] D. Lin and C. Yan, "Lung nodules identification rules extraction with neural fuzzy network", IEEE, Neural Information Processing, vol. 4, 2002.

[4] S. G. Armato, M. L. Giger and H. MacMahon, "Automated detection of lung nodules in CT scans: Preliminary results", Med. Phys., Vol. 28, pp. 1552–1561, 2001.

[5] B.V. Ginneken, B. M. Romeny and M. A. Viergever, "Computer-aided diagnosis in chest radiography: a survey", IEEE, transactions on medical imaging, vol. 20, no. 12, 2001.

[6] M. Fiebich, D. Wormanns and W. Heindel, "Improvement of method for computer-assisted detection of pulmonary nodules in CT of the chest", Proc SPIE Medical Imaging Conference,

vol. 4322, pp. 702–709, 2001.

[7] M. N. Gurcan, B. Sahiner, N. Petrick, H. Chan, E. A. Kazerooni, P. N. Cascade and L. Hadjiiski, “Lung nodule detection on thoracic computed tomography images: preliminary evaluation of a computer-aided diagnosis system”, Medical Physics, vol. 29, no. 11, pp. 2552-2558, 2002.

[13] R. Wiemker, P. Rogalla and R. Zwartkruis, T. Blaffert, “Computer aided lung nodule detection on high resolution CT data”, Medical Imaging, Image Processing, Proceedings of SPIE, vol. 4684, 2002.

[14] Fukunaga and Hostetler, "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition", IEEE Transactions on Information Theory vol 21 , pp 32-40 ,1975