Lung Cancer Detection by HU values in CT Scan Images

Ghodratollah Aalipour

ABSTRACT

Lung cancer is the most common type of cancer among various cancers with the highest mortality rate. There is an urgent need for a mechanism that can detect cancerous tumors in the chest cavity through computed tomography (CT) images. Early detection is critical for both patients and insurers and can save millions of dolors and thousands of people lives. In this short note, we will review some techniques that have been applied so far towards this goal. We also evaluate their performance through different metrics. Moreover, we try to make improvement if possible.

1. OVERVIEW

Lung cancer is a common cancer between men and women, and one of the deadliest cancers. According to [1], approximately, 158,000 patients suffering from lung cancer in the United States were expected to die in 2016. According to [2], it strikes 225,000 people every year and costs healthcare providers $12 billion every year. This can be significantly reduced by early detection to give patients the best survival solutions.

Data. We use the database provided kaggle at [6]. This database consists of thousand low-dose computed tomography (CT) images from high-risk patients in DICOM format. Each image contains a series with multiple axial slices of the chest cavity. Each image has a number of 2D cross sectional slices. The number of slices is depended to the machine taking the scan and patient. Each patient has a unique identification number (id) which is available in the header of DICOM file. The header contains id and some other scan parameters such as thickness of each slice. The average number of cross sectional slices for each patient is around 130 slices. Note that the tumor can be anywhere so it might appear only in few first or last slices.

Objective. Using a database of high resolution lung CT scan images provided by National Cancer Institute (NCI), we want to develop an algorithm to predict when lesions in lungs are cancerous.

Approach. Determining a cancerous lesion in the CT scan image is a binary classification problem. We can turn each image into a binary image after applying some filters. In the binary version, the lesion can be detected as a white region through an appropriate thresholding. I expect that extracted feature will be all numeric. These feature can be estimated volume of the lesion or, average length and average width of lesion. So, applying a logistic regression seems plausible. The other classifiers which are good candidate for this purpose are k-NN and SVM. It might be interesting to apply each of these algorithms separately and compare their outcome for our database.

Techniques Studied. So far, several methods have been designed towards this direction of research and their performance have been evaluated by several databases. In this note, we review some of the available methods.

In [1] Anirudh *et al.* exploit 3D convolutional neural networks (CNN) to learn highly discriminative features for nodule detection in lieu of hand-engineered features such as geometric shape or texture. Existing computer aided diagnosis (CAD) methods depend to getting detailed labels for lung nodules, to train models. To overcome this challenge, they present a new method in which the expert needs to provide only a point label, *i.e.*, the central pixel of the nodule, and its largest expected size. They apply an unsupervised segmentation to grow out a 3D region and show that the network trained by these weak labels generate low false positive rates and high sensitivity.

In [2], Dandil *et al* designed a computer aided diagnosis system that is based on segmentation of nodules. It classifies nodules between benign and malignant through an ANN (Artificial Neural Network). They obtained the performance values of 90.63% accuracy, 92.30% sensitivity and 89.47% specificity, where they had 128 CT Scan images obtained from 47 patients.

In [3] Kumar, Wong, and Clausi propose a system based on deep learning which takes features from an autoencoder to classify lung nodules. Their design has an accuracy of 75.01% with a sensitivity of 83.35%. They use a 10 fold cross validation.

In [4] Kuruvilla and Gunavathi develop a lung cancer classification method using computed tomography (CT) scan images through an artificial neural network. They use several statistical parameters like mean, standard deviation,

skewness, kurtosis, fifth central moment and sixth central moment for their classification purpose. They show that for their data the feed forward back propagation network performs better than the feed backward propagation and that the parameter skewness has the highest classification accuracy. Among the already available thirteen training functions of back propagation neural network, the Traingdx function has the maximum classification accuracy of 91.1%. They also introduce two new functions to achieve an accuracy of 93.3%, specificity of 100% and sensitivity of 91.4% and a mean square error of 0.998.

In [5], Mahersia and Zaroug summarize a list of the same techniques by taking into the part other available resources.

In [8] the authors use a neural network technique along with the genetic algorithm to build a classifier of lung nodules without computing the shape and texture features. Their algorithm has a performance with the best sensitivity of 94.66%, specificity of 95.14%, accuracy of 94.78% and area under the ROC curve of 0.949.

2. DESIGN CONSIDERATIONS

In this section, we consider the important parameters that we take into the account for our classification purpose. There are several images, over one hundred, for each patient. They are cross-sectional slices from the chest cavity. So they start from stomach and go up to the area below the throat. We have associated meta data for images that can give us interesting information.

Middle Slices and Thickness. The thickness of slices varies from patient to patient. It seems that the best slices are the middle ones. To get the middle slices, we sort the images based on their location. Then we choose the middle images in our sorted list. For each patient we can find the thickness of slices from the meta data of images. This can be done by the following MATLAB pseudo-code :

```
file = dicom-file.dcm;
imdic = dicominfo(fn);
sliceloc = imdic.SliceLocation;
slice-thickness = abs( sliceloc(1) - sliceloc(2) );
```

Hounsfield Unit. The **Hounsfield unit (HU)** is a quantity that is usually used in CT images in standard format, see [7]. Hounsfield units, created by and named after Sir Godfrey Hounsfield, are obtained from a linear transformation of the measured attenuation coefficients.

$$HU = (pixelvalue) * RescaleSlope + RescaleIntercept$$

Again, using meta data available on each image header, We find that RescaleIntercept = imdic.RescaleIntercept = -1024 and RescaleSlope = imdic.RescaleSlope= +1. Thus

$$HU = pixelvalue - 1024$$

The scale is defined in Hounsfield units (symbol HU), running from air at -1000 HU, through water at 0 HU, and up to dense cortical bone at +1000 HU and more for bones. This transformation is based on the arbitrary definitions of air and water. These value are significantly helpful for our segmentation as they act similar to color spaces. For instance, in evaluation of tumors, an adrenal tumor with a radiodensity of less than 10 HU is rather fatty in composition and almost certainly a benign adrenal adenoma. The values of several tissue of human bodies and other items are listed in the Table 1. We use the thresholds in Table 1 later for our segmentation goals. According to [7], Hounsfield units are measured and reported in a variety of clinical applications including:

Substance	HU Interval
Air	[-1020, -1000]
Lung	[-900, -320]
Fat	[-100, -50]
Blood	[30, 45]
Water	[0,4]
Bone	[170, 3000]
Liver	[40, 62]
Muscle	[10, 40]

Table 1: HU values for substances

- Measuring the amount of fat content in the liver,
- Assessing bone mineral density (BMD),
- Predicting the presence of anemia,
- Guiding the management of kidney stones.

2.1 Segmentation

One of the most important parts of image processing projects is segmenting the foreground from background. Using thresholding and the values of HU, we can segment different tissue. For instance we may obtain the following segmentations:

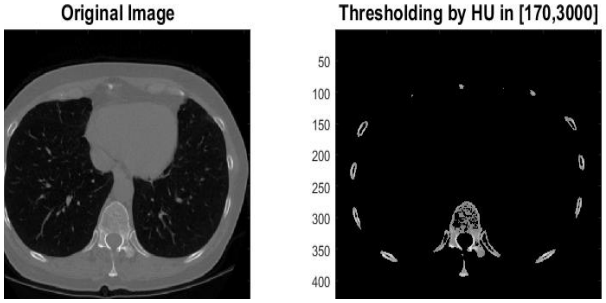

Figure 1: Segmenting bones by thresholding.

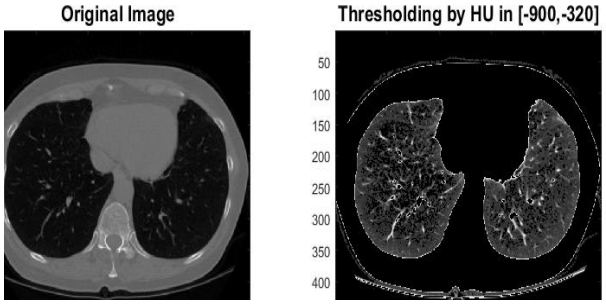

Figure 2: Segmenting lungs by thresholding.

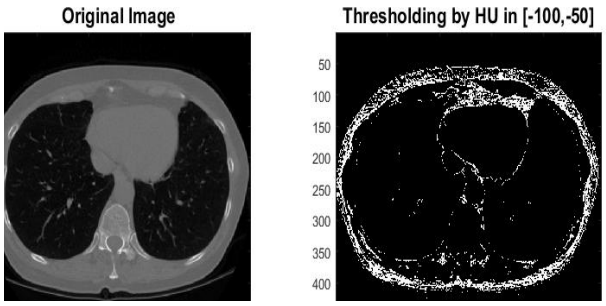

Figure 3: Segmenting Fat by thresholding.

Since we want to design an algorithm for lung cancer detection, we focus our attention on lung. As it can be seen from Figure 2, using thresholding method we get extra pixels that are related to the body or the imaging facility. To get rid of them we apply thresholding followed by a sequence of morphological operations to segment exactly lungs.

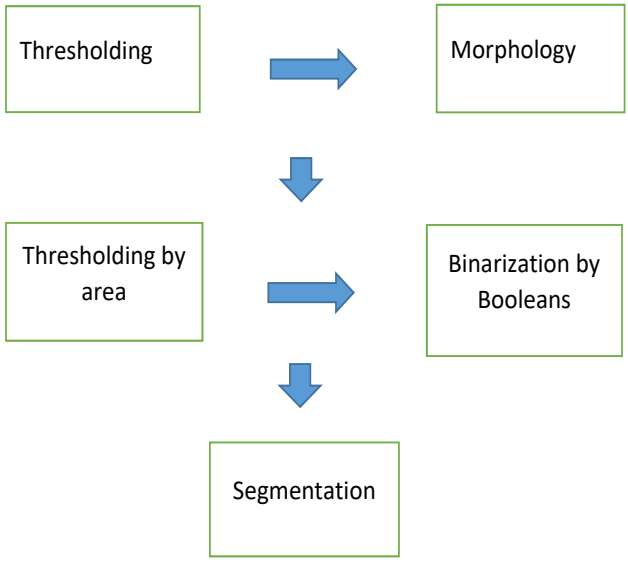

Figure 4: The process for lung segmentation.

In Figure 2.1, we have the original image presented next to the segmented image. We illustrate the steps above through the following example.

3. FEATURE EXTRACTION

Now that we have pre-processed and normalized our images, we can start extracting some features from the images and then feed them to some classification algorithm. One of the features that we include among our set of features is the area of regions with specific pixel values. These regions could be cancer tumors. So we check their areas if this is bigger than a particular threshold then by a 1R we might deduce some conclusions.

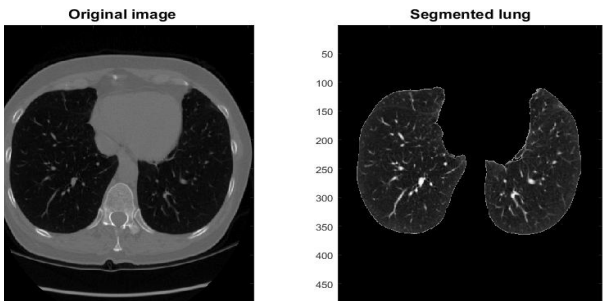

Figure 5: The process for lung segmentation.

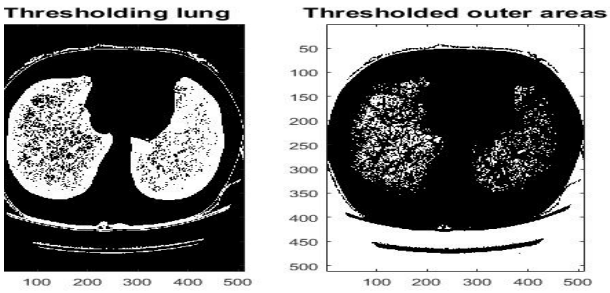

Figure 6: Thresholding applied for segmentation.

Threshold. Selecting th threshold for each substance is significantly important for our purpose. We can leave it as an artificial neural network project to let it run for all appropriate threshold and then choose those ones that minimize our misclassification error or other types of errors. As running the MATLAB code for each patient takes between 40s to 90s, depending on the number of images, it would be much time consuming, we skip this method and applied some thresholds manually to the images that have parts of tumors.

We apply three sets of thresholds to our images:

1. **Cut off threshold**. After sorting images, we skip initial and terminal images. These are the images that are either very close to neck (cervical) or very close the tailbone. We skip the first 20 top images and the last 45 bottom images from the set of images for each patient.

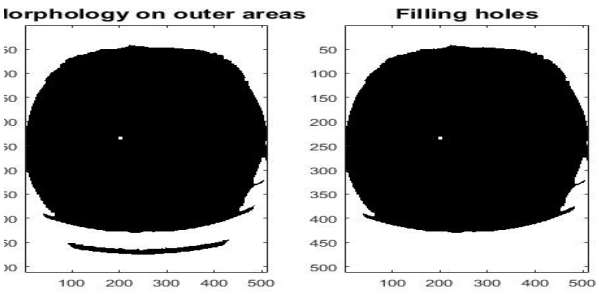

Figure 7: Morphology applied for segmentation.

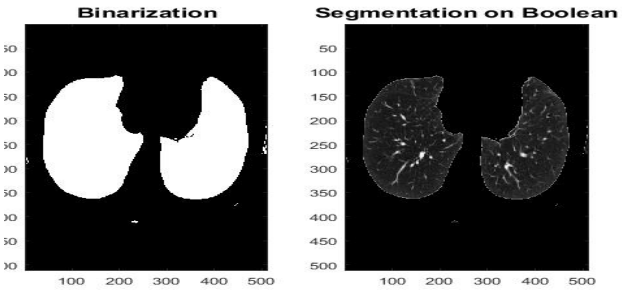

Figure 8: Connected Component Analysis.

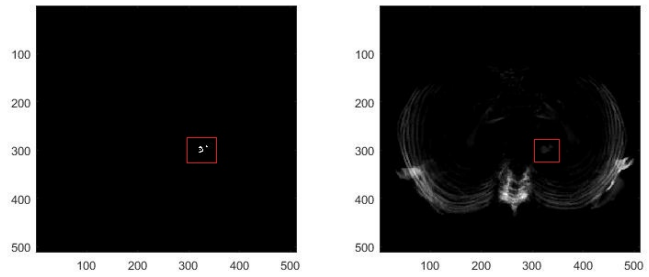

Figure 9: Chest Cavity: Project_Tumor on the left and Project_Bone on the right

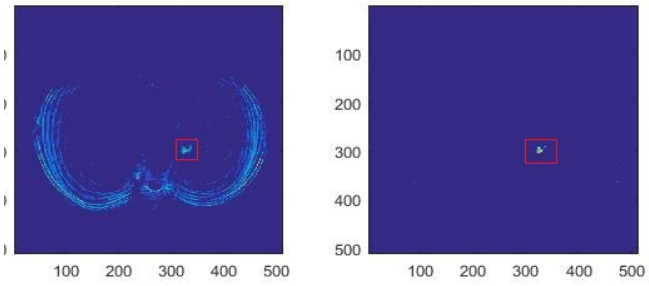

Figure 10: Lung and the Tumor inside: ProjectLung on the left and Project_HU on the right.

2. **HU threshold for tumors**. This is the threshold we choose for values of tumors. Looking at the images manually, and also other CT images from cancerous tumors suggests that the HU value for them is very high, above 2000. So we selected the threshold 2000 for HU value.

3. **HU threshold for lung**. This is the threshold by which we select only those pixels in the lung area (derived from the previous step) that have a value more than this threshold. This approach helps us find tumors in the area of the lung. We select the threshold 700 for HU.

Now, corresponding to each of the last two thresholds above, we associate several images that are either Boolean or sub-sampled from HU images.

For boolean image L_i that is the region corresponding to lung for the i-th image I_i

$$(Project\_Tumor)_i = HU \geq 2000;$$

$$(Project\_Bone)_i = (HU \geq 700 \ \& \ HU \leq 3000);$$

$$(Project\_Lung)_i = (HU > 700) \ \& \ L_i;$$

$$(Project\_HU)_i = HU.^*(HU \geq 1600);$$

We determine each ProjectTumor, ProjectLung, and ProjectHU for each patient. Now if a patient p has n CT images in our database, then we define the following images (matrices) for this patient p as follows:

$$Project\_Tumor = \frac{1}{n}\sum_{i=1}^{n}(Project\_Tumor)_i;$$

$$Project\_Bone = \frac{1}{n}\sum_{i=1}^{n}(Project\_Bone)_i;$$

$$Project\_Lung = \frac{1}{n}\sum_{i=1}^{n}(Project\_Lung)_i;$$

$$Project\_HU = \frac{1}{n}\sum_{i=1}^{n}(Project\_HU)_i;$$

For a patient with id 0015ceb851d7251b8f399e39779d1e7d, we get the following images:

Now, we are ready to extract our features from these images. We define the following features.

$$avg\_area = sum(Project\_Tumor(:))/n; \quad (1)$$

$$avg\_max = max(Project\_Tumor(:))/n; \quad (2)$$

$$avg\_HU = max(Project\_HU(:))/n; \quad (3)$$

$$\mu = mean(Project\_Lung(:)); \quad (4)$$

$$\sigma = std(Project\_Lung(:)); \quad (5)$$

For simplicity, we call avg_max and avg_area as max and area. For 631 patients in our database, we have the following results:

id	n	area	max	HU	μ	σ	cancer
0	195	26.22	0.43	44.68	0.17	0.79	1
1	265	1.15	0.15	22.41	0.11	0.55	0
2	233	0.00	0.00	20.68	0.13	0.74	0
3	173	0.00	0.00	9.40	0.08	0.52	1
4	146	16.3	0.54	42.06	0.06	0.43	1
5	171	3.60	0.16	33.26	0.08	0.52	0
6	123	0.00	0.00	0.00	0.03	0.23	0
7	134	0.00	0.00	0.00	0.13	0.73	0
8	135	0.00	0.00	0.00	0.04	0.38	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
628	267	111.06	0.55	138.09	0.27	1.40	0
629	259	0.00	0.00	6.39	0.23	1.22	1
630	159	6.83	0.49	50.43	0.20	0.85	0

4. PROCESSING DATA

By looking at the values in this table, we realize that we need some data pre-processing operations before applying any analytical tools. A simple query shows that

	area	max	HU
min	0	0	0
max	3324.375	0.811428571	1374.512987

we can apply the Dynamic Ranging MinMax() built in Python to normalize our data. This method projects our data to interval $[0, 1]$. Since we have a wide range for area and HU features, we multiply the resulting value by 100 to map every data values to the interval $[0, 100]$. We still have many data points close to the origin. As an example of data spread, we consider the following graph showing the distribution of data points based on their

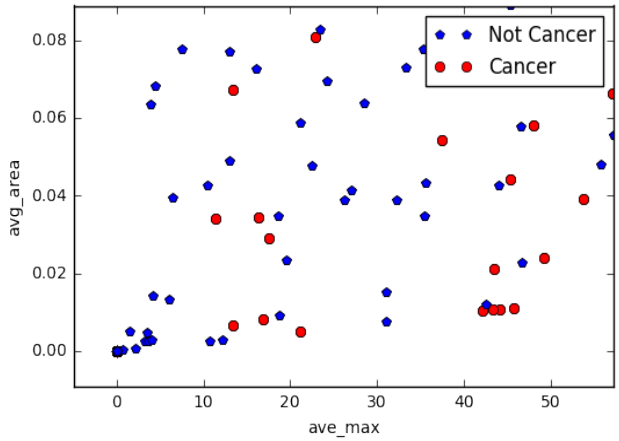

Figure 11: Plotting avg_max vs avg_area

This figure suggests us to use a distance-based algorithm for our classification purpose. One of the most famous and strong classification algorithm with this property is k-NN.

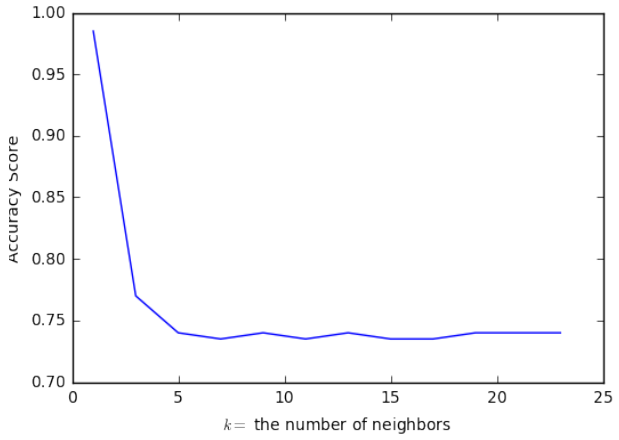

Figure 12: Accuracy vs k

This graph suggests that $k = 1$ is the best number of neighbors with a accuracy of 0.98 to use in k-NN.

5. CONCLUSION

In this note, we applied the HU value for images to detect cancerous tumors through several projection. Selecting thresholds for each substance was critical in order to complete segmentation. Then to project our images for a patient into a single image we applied other set of thresholds. Selecting the right thresholds here was challenging as well.

After selecting the thresholds in the previous step, we generated other images which are concentrate of all CT images of a patient. The we extracted six features from the derived images. After normalizing our data,, we applied a k-NN algorithm to our data to build a classifier.

We could apply other types of classifiers such as a Decision Tree. I ignored using a decision tree because of two reasons: (i) The range for features are very wide and so it is likely to overfit data, (ii) Plotting the data points suggests that a distance-based classifier seems to work better.

We may think to use other approaches such as a 1R approach. We can also consider momentum of images and add them to the list of images as discussed in [4]. For a data vector X, which can be an image, its k-th momentum is defined by

$$m_k(X) = \sum_{i=1}^{n} \frac{(x_i - \mu_i)^k}{\sigma}$$

So the first few momentums can be included among our features. Another approach is a combination two or more classifiers. We may alos use cross-correlation. For a patient, we choose one or few middle images and for each image we find its cross co-relation with other images. Then we vote and check which images gain the highest votes. Then among this odd number of images, we check the highest votes and assign the corresponding class to that patient.

6. REFERENCES

[1] Rushil Anirudh, Jayaraman J. Thiagarajan, Timo Bremer, and Hyojin Kim, SPIE Medical Imaging, International Society for Optics and Photonics, 2016.

[2] Emre Dandil, Murat Cakiroglu, Ziya Eksi, Murat Ozkan, Ozlem Kar Kurt, Arzu Canan, Artificial Neural Network-Based Classification System for Lung Nodules on Computed Tomography Scans, Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of. IEEE, 2014.

[3] D. Kumar., A. Wong, D. A. Clausi, Lung nodule classification using deep features in ct images, Computer and Robot Vision (CRV), 2015 12th Conference on, 133–138, IEEE (2015).

[4] Jinsa KuruvillaâĹŮ, K. Gunavathi, Lung cancer classification using neural networks for CT images, computer methods and programs in biomedicine 113 (2014) 202–209.

[5] H. Mahersia, M. Zaroug, L. Gabralla, Lung Cancer Detection on CT Scan Images: A Review on the Analysis Techniques, International Journal of Advanced Research in Artificial Intelligence(IJARAI), 4(4), 2015.

[6] https://www.kaggle.com/c/data-science-bowl-2017/data, 14:30 EDT April 22, 2017.

[7] https://radiopaedia.org/articles/hounsfield-unit, 19:26 EDT April 25, 2017.

[8] Giovanni L. F. da Silva, Otilio P. da Silva, Aristofanes C. Silva, Anselmo C. de Paiva, Marcelo Gattass, Lung nodules diagnosis based on evolutionary convolutional neural network, Multimedia Tools and Applications (2017) 1–17.