Improvement of Lung Cancer Detection using Convolution Neural Networks

Avirup Chakraborty (UFID – 5291-4909)

***Abstract*—Lung Cancer has been on the rise and has been a major cause of death in the developed as well as developing countries. In the United States, lung cancer strikes 225,000 people every year, and accounts for $12 billion in health care costs. Early detection is critical to give patients the best chance at recovery and survival. The prognosis for this disease is poor with less than 15% of the people surviving after 5 years of diagnosis.**

Automated Computer-Aided Detection (CADe) has been an important tool in clinical practice and research. State-of-the-art methods often show high sensitivities at the cost of high false-positives (FP) per patient rates for effective cancer detection. However, this high amount of false positive rates lead to unnecessary patient anxiety and lead to additional follow up interventional treatments. This paper proposes a method which tries to improve on the lung cancer detection system by proper segmentation of lung nodules on different slices of the CT scans and then tries to apply deep learning methodology like Convolution Neural Networks (CNN) using TensorFlow framework on those segmented scan slices and discards the unnecessary information in order to narrow down the relevant slices and predict whether the patient has lung cancer in the final layer of the fully connected network. The method is divided into 2 major sub-sections – Pre-processing of the CT scans into a resampled and rescaled 3D segmentation of lungs, the pre-processed image is then fed into a two tier ConvNet which transforms from a coarse to fine cascade framework that leads to reduction of the false positive rate from the Tier I to Tier II of the analysis. Initially, the segmented lung image is fed into the Tier I ConvNet which identifies the necessary region of interest (ROI) in order to remove those CT scan slices that doesn't concern our analysis. This improves the accuracy as well as the computational performance. The selective slice reduction of CT scans acts as input to train the Tier II deep convolution neural network (ConvNet). This second tier behaves as a highly selective process to reject difficult false positives while preserving high sensitivities.

Index Terms*—Computer-Aided Diagnosis, Medical Diagnostic Imaging, Machine Learning, Computed Tomography, Image Segmentation, Pattern Recognition, Convolution Neural Networks, Multi-layer Neural network, Tensor Flow and Deep Learning.

I. Introduction

Lung cancer has been on the rise and has been a major cause of death in the developed as well as developing countries. In the United States, lung cancer strikes 225,000 people every year, and accounts for $12 billion in health care costs. Early detection is critical to give patients the best chance at recovery and survival. However, this is a malignant disease carrying a poor prognosis, with sufferers having an average 5-year survival rate of less than 15% [1]. Patients having locally advanced, or medically inoperable disease are usually treated with concurrent radiotherapy and chemotherapy. Although targeted therapeutics and various chemotherapy regimens are available, locally advanced lung cancer carries a very poor prognosis, with a mean survival time of less than 12 months. Thus, early detection of a lung lesion to improve the complete resection rate (R0 resection) and increase the likelihood of survival rate is important. Chest computed tomography (CT) scan, especially high resolution CT, has been widely accepted for detection of lung tumors. Intriguingly, small lung nodule(s) noted on CT images make the differential diagnosis clinically difficult and may confuse clinical decision-making. Small lung nodules are seldom regarded as malignant, and are also difficult to biopsy or excise, and not reliably characterized by positron emission tomography scan [2]. In standard diagnosis, the American College of Chest Physicians has published a guideline for the diagnosis and management of pulmonary nodules which states that small nodules less than 8 mm could be further surveyed, characterized, or kept under observation according to evidence-based risk estimation [3]. However, the current guideline relies mainly on the size of lung nodules, rendering clinical decision-making difficult and controversial. Clearly, the development of a more informative tool for clinicians to differentiate the nature of pulmonary nodules noted on CT scan remains an urgent task. This is where the Computer Aided Detection(CADe) plays a vital role in early detection of this dreaded disease using lung module segmentation and classification from CT scans using deep neural networks [4]. Even though the State-of-the-art methods often show high sensitivities at the cost of high false-positives (FP) per patient rates for effective cancer detection. However, this high amount of false positive rates lead to unnecessary patient anxiety and lead to additional follow up interventional treatments.

Nodule segmentation is a challenging step in digital pathology that identifies cell regions from micro-slide images and is fundamental for further process like classifying sub-type of tumors or survival prediction [5]. The hand-crafted features that depends on shape features, signal intensity etc. has couple of major drawbacks 1) these techniques require several expertise of parameter selection manually which puts additional pressure on radiologists. 2) Generalization of shape and morphological features is difficult to achieve due to lack of symmetry and irregularity of different types of cancer cells.

In this paper, a solution is explained for effective nodule segmentation using convolution neural networks which extracts important features from the CT scans and trains itself to classify each patient for detecting lung cancer.

II. Solution Description and Implementation Details

A. Data Collection and Data Analysis

The dataset used for this analysis has been collected from Kaggle of size 66GB approx. which includes the high-resolution CT scans along with image labels for 1593 patients but there are labels missing for certain patients which reduced the dataset to approx. 1300 patients. The dataset consists of labels classified as 0 for non-cancerous patient and 1 for cancerous patient but no label is provided at the slice level of each patient. The patient label has been assigned directly to each slice label which runs of risk of some misleading information but it's worth a try.

B. Data Pre-Processing

The pre-processing of the high-resolution CT Scan images is one of the most important steps prior to fitting the images to any Convolution Neural Network. This is the step where only the most relevant embedded information is extracted and the unnecessary information and positions of the image is discarded which helps the Neural Network to better learn about the images and improve the detection with high accuracy. The framework that is used is a combination of OpenCV and Scikit-Image in order to get the best of both worlds. Many Pre-Processing approaches were tried in order to analyze and compare the results of each approach. First, the CT Scan DICOM images were loaded and the missing metadata information of slice thickness was added to each slice by calculating the difference between corresponding slice image positions or the slice locations found in the metadata of the DICOM files [7]. The unit of measurement of CT Scans is Hounsfield Units(HU) which gives the measure of radio density. The standard CT scanners are calibrated to measure the radio density accurately in this scale which represents different organs of a body with a standard Hounsfield Unit Value, e.g. the lung is represented by the value of -500 HU while the kidney is represented by the value 30 HU [6]. So, the original pixel intensities of the CT scan slices are converted into the Hounsfield unit (HU) scale which is a linear transformation of the original linear attenuation coefficient measurement into one in which the radio density of distilled water at standard pressure and temperature (STP) is defined as Zero Hounsfield units (HU), while the radio density of air at STP is defined as -1000 HU [6].

The converted slices are resampled into a standard isomorphic resolution in order to remove variance in scanner resolution by setting a standard setting of 1mm x 1mm x 1mm pixels which helps the 3D convolution network to train itself without worrying about learning zoom/slice thickness invariance. Then, a lung segmentation mask is applied to select the lung section of the slices by using the range of greater than -320 HU.

After this step, there were 2 approaches by which the selective filtering of the slices having possible nodules was done for all the slices. In one approach, the HU distribution of the slices was studied by a histogram and then the lung mask was identified based on the HU range threshold of greater than 20 frequency count which bypassed the non-nodule section of the lungs. In the 2nd approach, the CT scan image was normalized and zero centered. Then, the kernel smoothing operation was done by applying Gaussian function on the normalized image in order to smoothen the noise. Then, the peak detection technique was applied on the Gaussian curve in order to detect all the peaks. Both the approaches were used to filter the slices by checking the multi-modal distribution of the pixel intensities where the other smaller peaks in the histogram represents possible presence of a nodule and used for fitting into the Convolution Neural Network. After selective filtering step, each segmented image slices were cropped and rescaled to a standard size of 512 x 512 pixels in order to maintain a standard shape for all the images having the lungs [7].

Summary of the Pre-Processing Algorithm: -

a) Load each DICOM Image file and calculate the slice thickness using the difference of ImagePositionPatient metadata between two slices and if the ImagePositionPatient value is missing from the slice, then, difference of the slice location is used for the calculation.

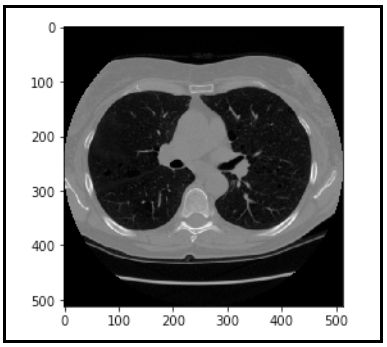

Fig 1. *Original CT Scan DICOM Image*

b) Convert the Original Pixel Intensities into the Standard Hounsfield Unit (HU) by rescaling the slope and intercept of each slice pixels.

c) Resample the slices by re-computing the original pixel spacing by adding the slice thickness in step (a) which creates a new resize factor. Then, interpolate the image intensities with the new refactoring vector to generate the standardized pixel spacing of 1mm x 1mm x 1mm.

d) A lung segmentation mask is generated by selecting the slices greater than -320 HU and applied to the original image to segment the lungs from the rest of the organs.

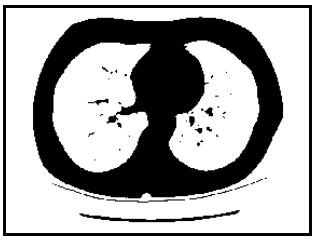

Fig 2. *Segmented Lung Mask*

e) Histogram of this lung segmented image is studied and 2 approaches are applied for nodule detection: -

i) Frequency Count of 20 is selected for detection of bi-modal distribution in the histogram which denotes the nodules.

ii) Gaussian Kernel Smoothing is applied on the Histogram and peak detection is done on the Gaussian curve. The slices having multiple peaks are selected as the

multiple peaks denote the possible nodules.

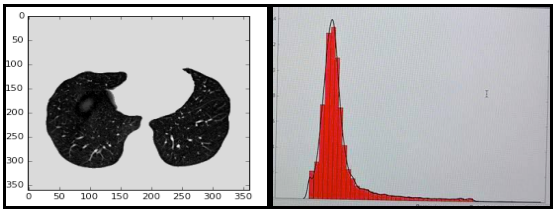

Fig-3. a CT Scan image with no nodules.

Fig-3. b Unimodal distribution of the slice with no nodules based on approach 2 using Gaussian Distribution.

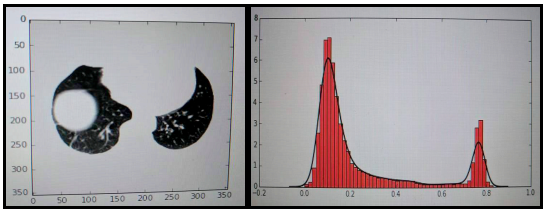

Fig-4. a CT Scan image with visible nodules.

Fig-4. b Bi-Modal distribution of the slice with visible nodules based on approach 2 using Gaussian Distribution.

f) The selected slices are then cropped and rescaled to a standard size of 512 x 512 dimensions and is saved in a pickle file for using in the next steps.

Code Snapshot of the selective filtering approach:

```
def tagSlicesNodules(image):

    try:
        selected_slices = list()

        for i in range(len(image)):

            sliceX = image[i]
            mode_val = stats.mode(sliceX.flatten())
            sliceX = sliceX[sliceX != mode_val]
            sliceX = sliceX[sliceX != 0.0]
            if (len(sliceX.flatten()) == 0):
                continue

            #Kernel Smoothing
            kde = sm.nonparametric.KDEUnivariate(sliceX.flatten())
            kde.fit(kernel='gau', bw='scott', fft=True)

            #Peak Detection for bimodal distribution
            indexes = peakutils.indexes(kde.density)

            if (len(indexes)>1):
                cropped_image=
image[i][min(np.where(image[i]!=0)[0]):max(np.where(image[i]!=0)[0]),min(np
.where(image[i]!=0)[1]):max(np.where(image[i]!=0)[1])]
                rescaled_image = misc.imresize(cropped_image, size=(512,512))
                re_dimensioned_image = cv2.cvtColor(rescaled_image,
cv2.COLOR_GRAY2BGR)
                selected_slices.append(re_dimensioned_image)


        print ("Number of Filtered Slices:" + str(len(selected_slices)))
        return selected_slices
    except BaseException as e:
        print str(e)
```

C. Architectural Overview of the solution using ConvNet and TensorFlow Framework

Convolution Neural Networks take the advantage of the input which is an image and they constrain the architecture in a way so that there are neurons arranged in the ConvNet in the form of 3 dimensions (width, height, depth) unlike a regular neural network. The ConvNet is divided into 3 distinct layers - Convolutional Layer, Pooling Layer, and Fully-Connected Layer like the regular neural network. The Convolutional Layer computes the output of the neurons based on the dot product of the weights of the local filters along with the input values connected to the local filters. The Pooling layer does a down sampling operation along the spatial dimension of height and width. Finally, the Fully Connected Layer computes the class scores resulting in a single vector of binary values based on the input categories. Each layer of the ConvNet learns certain features and reduces the dimensions of the original input to a single vector of class scores which are used to classify the two types of patients: Cancerous or Non-Cancerous. The number of layers and the choice of the hyper parameters varied based on trial and error and whatever works best on the ImageNet [8]. The TensorFlow framework is used for its five major advantages: a) Flexibility to represent multiple versions of the same convolution model or simultaneously provides multiple models. b) Portability to deploy the trained model to any device irrespective of any additional hardware support. c) Research and Production for reusing the same set of modules for different case studies without any code alterations. d) Auto Differentiation capabilities which benefits gradient based machine learning algorithms. The computational architecture of the predictive model can be defined and combined with any objective function and data. TensorFlow manages the derivatives of computing processes automatically. e) Performance enhancement due to its advanced support for threads, asynchronous computation, and queues. TensorFlow can manage all the independent copies of computational elements of the flow graph on different devices automatically. It also facilitates with the language options to execute the computational graph [9].

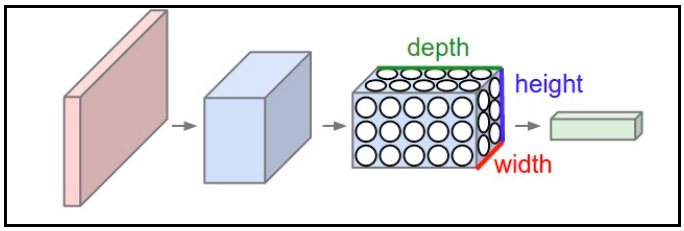

Fig 5. *Basic Convolution Neural Network Architecture*

D. CNN Architecture used for training the model

There are many CNN architectures that are state of the art for pulmonary nodule detection and cancer prediction with CT Imaging. However, in all the architectures, the data set had the exact position of the nodules annotated by the radiologists which helped the model to detect the nodules easily. One of the state of the art architecture is the ReCTNet which uses the nodule annotations to detect any area of interest and generates 3D probability maps for highlighting them [11].

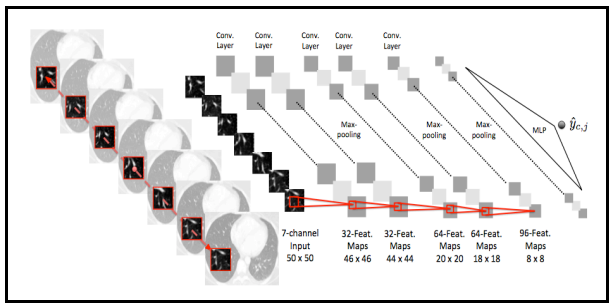

Fig 6. *Original ReCTNet Architecture*

But the challenge in this data set was the lack of information about the nodules. I applied a modified version of the original ReCTNet architecture by varying the layers in order to compare the performances as mentioned in the summary below:

Summary of the modified ReCTNet CNN Architecture:

a) Convolution Layer of 32 filters, each of size 5.
b) Convolution Layer of 32 filters, each of size 3.
c) Max Pooling Layer of size 2.
d) Convolution Layer of 64 filters, each of size 3.
e) Convolution Layer of 64 filters, each of size 3.
f) Max Pooling Layer of size 2.
g) Convolution Layer of 96 filters, each of size 3.
h) Max Pooling Layer of size 2.
i) Fully Connected Layer of 512 Nodes with a rectified linear unit activation function.
j) Drop out layer with a keep rate of 50 % and 80%.
k) Fully Connected Layer of 2 Nodes with a softmax activation function generating 2 outputs for binary classification.

The model was trained using dynamic batch size which helped the model to identify and recalculate the weights within a patient only. This dynamic batch size was used since the labels in the dataset was for each patient and not for each slice.

E. Training, Testing Data Set

The 80% of the entire Kaggle dataset was used for training the model and 20% for testing [10]. An external LUNA-16 dataset having nodule annotations were also used to compare the performances of both the datasets [12].

F. Performance Evaluation and Results

The existing ReCTNet architecture provides an accuracy of 90.5% using provided the nodule annotations are given by the radiologists. However, if the key information of the nodules is not provided, there has not been much success in any of the existing algorithms. The nodule annotation is one of the key information which helps in nodule detection of random CT scans. The architecture which was used had 1000 training patients and 300 test set which gave an accuracy of only 35% with approach 1 of selective filtering. However, the results improved slightly to 39% with approach 2 when the threshold selection was randomized with automated peak detection. The approach works well with other data set which has nodule annotation and improves the accuracy to around approx. 86%.

Study	Sensitivity	FPs/scan	Training scans	Test scans
Golosio et al. (2009)	79.0%	4.0	84†	84
Messay et al. (2010)	82.7%	3.0	84†	84
Tan et al. (2011)	87.5%	4.0	235	125
Torres et al. (2015)	80.0%	8.0	94*	949
Teramoto and Fujita (2012)	87.0%	4.2	84†	84
ReCTnet	**90.5%**	**4.5**	600	150
	85.6%	3.5	600	150
CNNs	81.8%	6.8	600	150
	70.0%	3.0	600	150

Fig 7: *Results of different architectures with nodule annotations in data set*

Technique	Sensitivity	Training Size	Testing Size
Histogram Threshold (20)	30%	1000	300
Kernel Smoothing and Peak Detection with no nodule annotations	39%	1000	300
Kernel Smoothing and Peak Detection with nodule annotations (LUNA dataset)	86%	1000	300

Table-1: *Results of modified ReCTNet Architecture used in current study*

Patient ID	Cancer	Predicted Cancer
0015ceb851d7251b8f399e39779d1e7d	1	1
00cba091fa4ad62cc3200a657aeb957e	0	1
00edff4f51a893d80dae2d42a7f45ad1	1	1
024efb7a1e67dc820eb61cbdaa090166	0	0

Table-2: *Snapshot of the output of the current study*

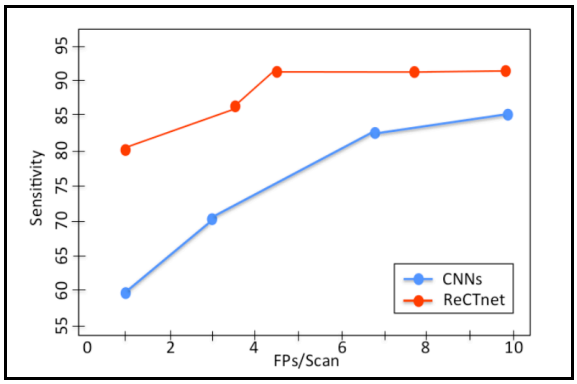

Fig 8. *Comparison between ReCTNet and CNNs architecture*

There were several runtime and storage optimization techniques which were applied to improve execution performance. The multiprocessing library was used with 4 and 8 parallel threads were implemented which significantly improved the pre-processing task. The results are given in the following table:

Technique	Sample Size (Patients)	Total Pre-Processing Time (Days)
Sequential	1300	5
Parallel (4 threads)	1300	3.5
Parallel (8 threads)	1300	2

Table-3: *Runtime Performance of the Pre-Processing task*

Initially, the results of the pre-processing were stored directly to one pickle file per patient which led to a storage requirement of around 1 TB in total but this was a huge overhead in terms of storage availability. In order to reduce the file size, file compression technique using the GzipFile python library which drastically reduced each file size from ~ 1 GB to around ~ 1 Mb. The training of the model took around 2 days for training 1000 patients which was optimized using Adam optimizer instead of the original Momentum optimizer.

File Storage	Sample Size (Patients)	Average File Size/ patient	Total File Size for all patients
Uncompressed	1300	1.3 GB	1690 GB
Gzip Compression	1300	1.67 MB	2.12 GB

Table-4: *File Storage optimization of the Pre-Processing Files*

III. Conclusion and Future Work

In this study, we have presented a modified convolution neural architecture which can identify nodules and other anatomical objects of interests in CT scans and can help in cancer classification by selective filtering technique. This modified ReCTNet has been tested on the challenging task of lung cancer identification in CT scans using a large dataset available on Kaggle website and has achieved a moderate sensitivity with the lack of the nodule annotations. In future, I would like to apply another pre-processing technique of 50% peak filtering which would remove the slices with distorted histogram distribution. This would further improve the sensitivity of the system and would remove any kind of local bias and reduce the number of false positive even further.

References

[1] Siegel R, Naishadham D, Jemal A. *Cancer statistics, 2013. CA Cancer J Clin. 2013;*63(1):11–30.

[2] Gomez Leon N, Escalona S, Bandres B, et al. *F-fluorodeoxyglucose positron emission tomography/computed tomography accuracy in the staging of non-small cell lung cancer: review and cost-effectiveness.* Radiol Res Pract. 2014:135934.

[3] Gould MK, Donington J, Lynch WR, et al. *Evaluation of individuals with pulmonary nodules: when is it lung cancer? Diagnosis and management of lung cancer, 3rd ed*: *American College of Chest Physicians evidence-based clinical practice guidelines.* Chest. 2013;143(5 Suppl): e93S–e120S.

[4] Hua, Kai-Lung, et al. *"Computer-aided classification of lung nodules on computed tomography images via deep learning technique."* OncoTargets and therapy 8 (2014): 2015-2022.

[5] Anirudh, Rushil, et al. *"Lung nodule detection using 3D convolutional neural networks trained on weakly labeled data."* SPIE Medical Imaging. International Society for Optics and Photonics, 2016.

[6] *https://en.wikipedia.org/wiki/Hounsfield_scale*

[7] *https://www.kaggle.com/gzuidhof/data-science-bowl-2017/full-preprocessing-tutorial*

[8] *http://cs231n.github.io/convolutional-networks*

[9] *http://www.softwebsolutions.com/resources/tensorflow-googles-artificial-intelligence-system.html*

[10] *https://www.kaggle.com/c/data-science-bowl-2017/data*

[11] Ypsilantis, Petros-Pavlos, and Giovanni Montana. *"Recurrent Convolutional Networks for Pulmonary Nodule Detection in CT Imaging."* arXiv preprint arXiv:1609.09143 (2016).

[12] *https://luna16.grand-challenge.org/download/*

Acknowledgments

I would like thank Dr. Dapeng Wu for giving me this opportunity to work on this challenging project and David Ojika for providing the insights into the performance aspects which helped in algorithm optimization. I would also like say special thanks to my friends Sabyasachi Bandyopadhyay and Estefany Suarez for sharing their expertise in the field of biomedical engineering.