Automatic Lung's Grade Image Classification from CT

Reinaldo Daniswara, Tianyang Zhang
Department of Computer Science
University of California, Los Angeles, 90095

***Abstract*—This paper presents the technique to develop a 3D convolutional neural network model from CT scan images to produce probabilities of individuals to have cancer. Before we develop the model, we need to do image preprocessing using watershed algorithm, which is a creation of internal, external, and watershed marker to segment images. Those preprocessed images then will be used as an input for training or testing model of CNN. After we build the neural network model, we will be able to train and test the dataset that we get from kaggle.com to get the probabilities of individuals to have a lung cancer.**

I. INTRODUCTION

Nowadays, lung cancer has become one of the deadliest cancer in the world. There were 155,870 people die due to lung cancers in 2017 [1]. The authors believe that this number cannot go any higher, and this huge amount of numbers motivates the authors to do a research on doing a lung cancer detection. One of the indication that a patient has a lung cancer is the appearance of nodal inside the lungs. (see Figure 1)

Thus, by doing an image processing on some CT scan images, develops a training model, our goal is providing a probability to detect the risk or a chance of a patient to have cancers using 3D-CNN method. Thus, treatment on early stages could be done faster and more efficient.

II. METHODS

A. Data Collections

In order to obtain data to do this project, we use data that was provided by kaggle.com [2]. The data was used for the competition before. In the webpage, they provided a couple files that we can download and play around with. We provided the link of the data on our website. Unfortunately, today the data are no longer accessible. However, we managed to download the data before the session expired. The dataset that we mainly used are:

Sample_images (1.9 GB), smaller dataset images, which contains 20 patients' CT scan images with DICOM file type for trial and error testing.
Stage1.7z: compressed of 66 GB file for testing and training data.
Stage1_solution.csv: contains the actual result for each patient, 1 means cancer and 0 means not cancer.

B. Image Preprocessing

CT scan is a special examination process using X-Ray to determine any anomaly that occurs inside the body. CT scan will generate very detail images of many types tissues, lungs, bones, and even blood vessels, which is the main advantages of CT scan compare to regular X-Ray [3]. As a result, CT scan is considered as one of the best method to find any strange or anomaly nodule that may occurs inside the lung.

However, the amount of noise that generated by the radiation of CT scan sometimes make it harder to determine whether it is a nodule or just other unwanted substances. Thus, before we could use the data to train the model, we need to do some image processing technique, to remove redundancy across all images, since not all of the images in the whole dataset have the same format. (See figure 2 and 3).

C. Lung Segmentation

Since nodules could appear anywhere on lung tissue, segmentation is required to mask out the bone, air, or any other substance that makes the data noisy. We are using watershed algorithm to segment the lung with the border to avoid any inaccuracy that may occurs for the nodule that appears in the border of the lungs. The idea and the algorithm are provided by kaggle notebook, with the topic of improved lung segmentation using watershed algorithm [4]. The idea is we segment the CT lung image into 3 markers. Internal, external, and watershed marker. Internal marker (figure 4) is the lung tissue, external marker (figure 5) is the outside region of lung tissue, and watershed marker (figure 6) is the

combination of internal and external marker with a different gray scale color.

Now, based on the watershed algorithm, we need to find the precise border that was indicated by the thin black marker between the internal and external marker. In order to do that we need to find the gradient of original images to get the regional minima (See figure 7 as references). In addition to that, we use the function black top hat to avoid missing nodules that located next to the border of the lung.

After that, we will get a segmented lung images. However, we still need to normalize the Hounsfield value for each image to get the enhanced segmented lung. Typically, housenfield value for air is -1000 and bone is 400, thus we set the HU value between that value since we will have a noisy image produced if we do not the threshold for the HU image.

D. 3D-CNN

We are using tensorflow to transform and build a 3D-CNN model from 3D CT scan to create images classifier, which will be used to train the dataset that we have. We choose to pick this method because unlike the traditional detection for the appearance of nodule in lung, 3D convolutional neural networks can learn useful representations directly from a large dataset without the need of extensive preprocessing pipelines.

The basic architecture for our neural network is CT images -> convolution & pooling -> flatten layer -> fully connected layer -> output layer.

Convolution basically the act of taking the original sample of data and create feature maps from it [5]. In addition, pooling is such a down-sampling, where we take the maximum value of a certain region and become a new value for the entire region. In our implementation, we do convolutions and immediately followed by max-pooling method for 3 times, which are 32, 64, and 128 features. After that, before the data can be used as an input for a fully-connected layer, we need to convert the convolution result into a 2D tensor via flatten layer. At the end, we need to connect all the nodes before producing the output, and here is the main role of the fully connected layer.

E. Training and Testing the Model

After that, we just need to train the model and see the result.

III. RESULT

Produced results is a csv file containing the id of the patient with the probability of cancer that it has. The table is too big to include it all in this paper, thus we will attach a separate csv file to look on it. However, here we will provide an example result for the first 5 out of 198 patients. (See table 1)

IV. PERFORMANCE ANALYSIS AND DISCUSSION

From 198 result of test set, we observed that the probability that we get for all of the data is around 0.1 and 0.3. Thus, to validate our data, we compare our result and the solution that provided by the Kaggle. We found out that for every patient ID that diagnosed cancer, it has average probability of 0.2599. On the other hand, for patient ID that does not have cancer, it has average probability of 0.1930. Further research is required to figure out why we did not get a higher value for the cancer patient. However, in term of counting the true positive or false positive for this research, we take a mean from both to be the threshold value or indicator value to determine cancer or not. For example, if a patient has a probability below 0.22 but it actually diagnosed cancer, than it counts as a false positive. Same case happens for patient that does not diagnosed by cancer, and if it has a probability above 0.22 then it is a false positive. In contrast, if the patient has a cancer probability of 0.26 and it was diagnosed cancer, then it is true positive because the probability is above our threshold, which is 0.22.

From the graph (see graph 1), we know that there is 25.25% false positive. In our opinion, the model is not completely perfect, but it is good enough to detect the probability of cancer.

V. CONCLUSION AND FUTURE WORK

To conclude, this paper writes a method on how to develop a deep learning method to create a lung cancer detection. First, we need to do the image preprocessing to mask out any substances that we do not want, such as air and bone. After that, we need to develop a neural network model to train our dataset to predict the probability of each patient to have a cancer. Lastly, we do the testing to generate the desired result. Since the probability that we generated only in range of 0.1 until 0.3, we set a threshold 0.22 as a threshold to determine a patient had a cancer or not. As a result, our experiment generates 25.25% false positive. It is obviously not a perfect model and there are some rooms of improvements in the future, such as we may need to tune the parameter to determine the threshold to find more accurate representation. In addition to that, the lack of training data also contributes a lot in determining the accuracy of the model.

REFERENCES

[1] "Lung Cancer Facts - Bonnie J. Addario Lung Cancer Foundation." *Bonnie J Addario Lung Cancer Foundation*, www.lungcancerfoundation.org/about-us/lung-cancer-facts/.

[2] "Data Science Bowl 2017 | Kaggle." Countries of the World | Kaggle, www.kaggle.com/c/data-science-bowl-2017/.

[3] Radiological Society of North America, et al. "CAT Scan (CT) - Chest." *Patient Safety - Contrast Material*, www.radiologyinfo.org/en/info.cfm?pg=chestct.

[4] "Improved Lung Segmentation Using Watershed | Kaggle." *Countries of the World | Kaggle*, www.kaggle.com/ankasor/improved-lung-segmentation-using-watershed/notebook.

[5] "Convolutional Neural Network (CNN) Basics." *Python Programming Tutorials*, pythonprogramming.net/convolutional-neural-network-cnn-machine-learning-tutorial/.

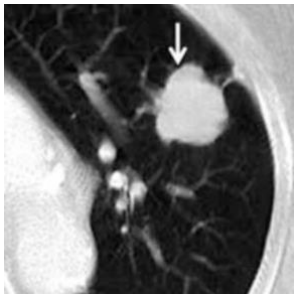
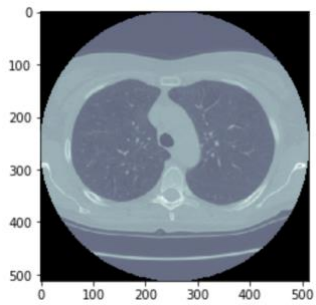


Figure 1. Example of nodal that appears in lung tissues.



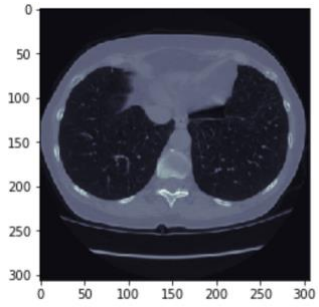


Figure 2. Original Images (left) and Figure 3. Images after rescaling the pixel and HU values

Figure 4. Top Left, Figure 5. Top Right, Figure 6. Middle Below

Figure 7. Gradient image (left) and segmented images with border (right)

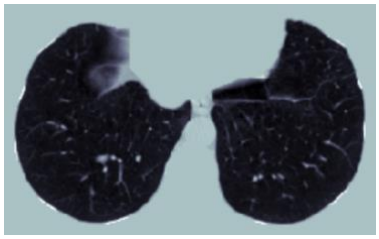
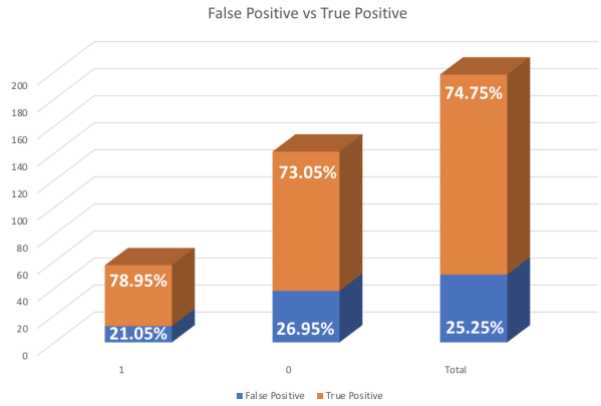


Figure 8. Segmented lung images

ID	Cancer Prob.
026470d51482c93efc18b9803159c960	0.247615
031b7ec4fe96a3b035a8196264a8c8c3	0.248471
03bd22ed5858039af223c04993e9eb22	0.085951
06a90409e4fcea3e634748b967993531	0.186083
07b1defcfae5873ee1f03c90255eb170	0.195519

Table 1. Example of the first 5 result.



Graph 1. The comparison between true positive and false positive