Lung Nodule Classification using Convolutional Neural Networks

Jiayu He
jih093@ucsd.edu

Yiding Qiu
yiqiu@ucsd.edu

Abstract

In this project, we applied deep learning to analyze medical imagery and used CNN to classify lung cancers. We evaluated different preprocessing methods on the dataset and used different neural networks (including ResNet-50 and Inception-v3) to extract features, and then used XGBoost to classify the data using different hyperparameters. In addition, we attempted to train a simple three-layer 3D CNN on the original dataset and tested for its performance. Results show that the best prediction was made by using ResNet and XGBoost with minimum preprocessing with a log loss 0.573 on the validation set. The naive 3D CNN implemented also showed performance that matches that of the pre-trained ResNet, showing its potential in solving the presented problem.

1 Introduction

The advancements in machine learning and deep learning have led to their applications to more and more industries. One of the industries that can be revolutionized by data science is the healthcare industry, as it is characterized by producing an enormously large amount of data every year with high manual operating costs and low efficiency. One of the ways that artificial intelligence can lower the healthcare costs and increases the efficiency is to help make faster and more accurate medical diagnostics. The benefits also exceed beyond lowering the cost of making the diagnosis – early detection maximizes the chances of survival the patient and minimizes the cost of the subsequent treatment.

As "the leading cause of cancer death, and the second most common cancer among both men and women in the United State," lung cancer has been an interest of research for years (Centers for Disease Control and Prevention) [2]. Characterized by abnormal cell growth and metastasis, the ability to invade other body tissues, lung cancer falls under the cases where early interventions will give patients the best chance of defeating the disease. According to the Centers for Disease Control and Prevention, low-dose computed tomography (aka low-dose CT) is the "only recommended screening test" for lung cancer [2]. Given the standardize screening method, and the amount of data can be collected from the population, it is possible to develop algorithms that are more accurate than the current technique which largely relies on radiologists.

Since the prevalence of CT scans, researchers have used different approaches to developing Computer Aided Detection (CAD) algorithms. Lung segmentation methods algorithms reduce the data size and are usually considered as prerequisites for automatic CT scan analysis. They are also useful when dealing with large-scale dataset so that the region of Interest (ROI) will not be damaged during the compression [3, 9, 19]. Different Machine Learning algorithms were tested on lung nodule

classification as well. Multi-view CNN is the most recent trend since it captures different angles of the scan images [7, 10, 16]. Many deep learning techniques also attempted to use 3D CNN, though most of them showed only preliminary results [5, 20]. The difficulty of nodule detecting appears to be multifold. Even with human eyes, the detection of nodules below 8mm was problematic [14]. As for computer vision, the major difficulty is that CT image data size is large, but the sample size of lung cancer patient is relatively small. Also, different CT instruments and settings (for example, some information was not stored properly) may also affect the data training. The obstacles posted and the necessities in detecting nodules require us to further explore the problem and to seek a more efficient solution. It seems that deep learning techniques may be able to fulfill this need.

The present paper attempted to develop such algorithm in hoping to solve the problems stated. We experimented with different models, such as ResNet, Inception-v3 and 3D convolution, and explored different ways of preprocessing the low-dose CT lung scan. In doing so, we hope to contribute understandings of how to approach and solve the problem of cancer detection, and potentially contribute to advancements in the healthcare industry.

2 Dataset and Platforms

2.1 Dataset

To achieve that goal, Data Science Bowl Challenge 2017 open-sourced a lung cancer dataset and presented a challenge to find the algorithm that can predict how likely the scanned patient will develop lung cancer within one year of when the scan was taken [8]. True labels were granted from the pathology diagnosis, and there were in total 362 cancerous examples out of 1397 labeled instances. The unlabeled instances were used as the testing set, though the true labels would not be provided before the competition close. One million prizes will be awarded to the best lung cancer detection algorithm, evaluated by having the lowest log loss function using natural log. Data were provided by the National Cancer Institute in DICOM format and were composed of low-dose CT images collected from 1595 patients. The dataset from each patient consisted a various number of 2D slices of the chest cavity in 512 x 512 pixels, of which can be rendered into a 3D scan. The number of images per patient ranges from 47 to 534, meaning that though the size of each image for all patients is the same, the size of data collected from each patient is different as the depths differ. Figure 1 shows examples of the individual lung slices collected from the same patient at different depths.

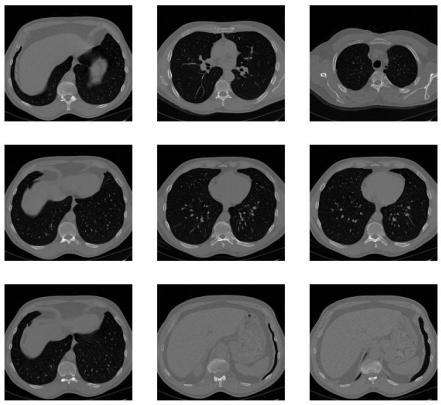

Figure 1. A sample of low-dose CT slices from one patient.

2.2 Platforms

Given the complexity of this problem, the present study used deep learning frameworks first to segment the area of interest, where cancer may develop, and to learn if the nodules found in that area would develop into cancer in one year period. More specifically, deep residual learning framework (ResNet) [6] were used on MXNet [12, 18] and a 3D convolutional neural network was used on

Tensorflow [1, 15]. Given these frameworks are usually used in Python language, the present study also used Python as the only operating language [13].

The present paper chose to use ResNet based on its observed high performances on the ImageNet dataset, and its generalizability on other visual recognition tasks. ResNet was introduced by He, Zhang, Ren, and Sun (2016) as a "deep residual learning framework," [6] in an attempt to address the degradation problem, where the accuracy of the network gets compromised as the network gets deeper. The residual network used "shortcut connection" without adding complexity to the network, thus evaded the problem of degradation that would result from simply stacking layers and was able to get lower training error and higher accuracy. He et al. (2016) tested ResNet on ImageNet dataset 2012 and modeled a 152-layer residual net that achieved 3.57% error on the ImageNet dataset. The net was deeper, yet it had lower complexity than counterpart such as VGG. The high performances of ResNet can also be generalized in other recognition tasks and have led He and colleagues to win several other competitions including ImageNet detection and COCO segmentation.

As its popularity raised, several variations of ResNet had being trained on the ImageNet dataset and were ready to be used for visual segmentation task by the time of the present study. ResNet-50 [6], as one of the example, was trained on ImagenNet 11k and Places 365. Xgboost [4] was applied to both models as an attempt to train the feature obtained. Besides ResNet-50, inception v3 [17] trained from ILSVRC 2012 validation set was also used as a comparison. We also trained a 3D convolutional neural network model with only two convolutional layers using Tensorflow, with a more naive preprocessing that aimed to reduce the size, to see how the performance differ when predicting if a patient will develop lung cancer in a year based on their CT scan. The computational limitation is the major concern with 3-D CNNs, thus present study only experimented it with shallow layers.

3. Method

3.1 Using MXNet with pre-trained neural network and XGboost

Two different pre-trained neural networks were used, ResNet and Inception-v3, to extract the feature of the CT-scan images. Then we used XGBoost classifier to train on features and to differentiate cancerous from non-cancerous cases.

3.1.1 Preprocessing. One preprocessing method was adopted from "Full Preprocessing Tutorial" [21] provided by Guido Zuidhof which was made available on GitHub. The pixel values were first converted to Hounsfield Units (HU) using the rescale slope and intercept [11], with filling (i.e. area outside the scanning bounds) of -2000 adjusted to zero to correspond to air. Resampling was done to achieve a constant pixel spacing of 1mm x 1mm x 1mm to ensure the slices were equally spaced. Visual segmentation was applied to segment the lungs to reduce feature dimensions. This was done using connected component analysis. Air pocket was labeled and eliminated, with only the HU of the soft tissues left for further processing. No mask was used in an attempt to lower information lost. Data were then normalized into the range of -1000 to 400 to capture mostly the tissues instead of the bones' radiodensity and were zero centered using the pixel mean of 0.25.

Figure 2 provided visual illustrations of how this preprocessing procedure can be used to segment different areas of interest.

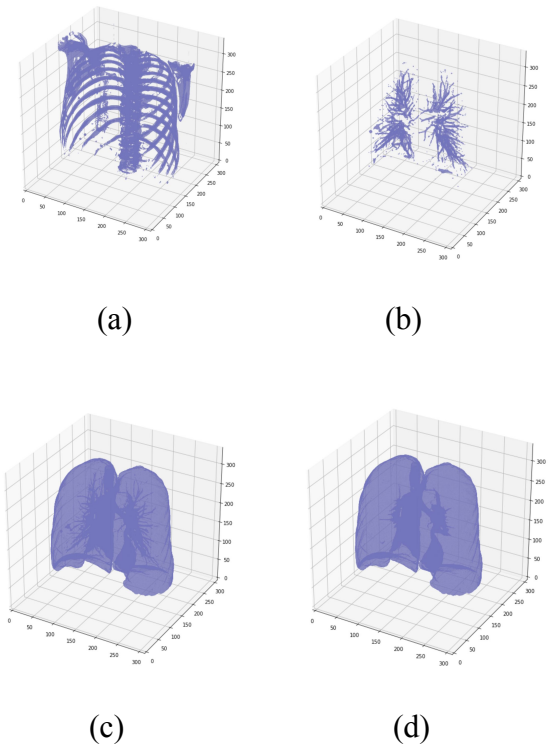

Figure 2. (a) Segmentation of bones using 400 threshold. (b) Segmentation of blood vessels. (c) Segmentation that gets rid of air in the lung except the largest air pocket. (d) Segmented lung with air fill.

3.1.2 Residual Networks. Three experiments in total were done using different ResNet models on MXNet, all with XGBoost. More specifically, one were done using ResNet 50 with minimum preprocessing, and two were preprocessed but used different hyperparameters for XGBoost. For feature extraction, we used the last layer ("flatten0") before the fully connected layer to extract the feature of our image. The batch size is set to 32, and the image is resized to fit the model (1*3*224*224). Since the model used 3 color channels of RGB, and there was limitation computational power, three consecutive images were chunk together and fed it into the model.

3.1.3 Inception-v3. One model used a pre-trained Inception-v3 with XGBoost and was tested on the preprocessed dataset. The hyperparameter of XGBoost is described in the next section. We also used the last layer ("flatten") before the fully connected layer to extract the feature of our image. We used the same method as with Resnet and resize images to 1*3*299*299.

Inception-v3 was adopted from the TensorFlow pre-trained Inception-V3 model from Szegedy, Vanhoucke, loffe, Shlens, and Wojna (2015) [17]. The model was trained on the ILSVRC 2012 classification challenge validation set and achieved an error rate of 3.46%.

3.1.4 XGBoost. Two models of ResNet 50 were experimented with two sets of parameters of XGBoost. Both models use the same learning rate of 0.05, the number of the thread of eight, the same subsampling ratio and column-wise subsampling ratio of 0.8, and seed number of 4242. The first model used tree booster of maximum depth of ten, a minimum nine instance weight need in a child, and 1500 estimators. The second model doubled the maximum depth and minimum child weight which led the minimum sum of instance weight of a child went up to 20, and used ten times as much estimators, listed in Table 1. The first model was also applied to the preprocessed dataset using the method described before, to see if preprocessing helped improved the overall performance of the model. Grid search was attempted to find the optimal parameter, and the first set of parameters appeared to have obtained smaller loss.

Hyper-parameter	Max. Depth	Min. Child Weight	Num. Estimators
xgboost_1	10	9	1500
xgboost_2	20	20	15000

Table 1. Parameters used in XGBoost.

3.2 Using TensorFlow to train 3D-CNN

3.2.1 Preprocessing. A different preprocessing was done when modeling the 3D convolutional neural network on TensorFlow. The main goal was to uniformly downsize the images into sizes within the computation constraint of the convolutional neural network. This preprocessing involved first downsizing the shape to 50 x 50 and chunking the data into the same number of slices of 20, as they started with different numbers of slices per patient. The resulting data for each patient is 50 x 50 x 20 to allow training in a timely fashion.

3.2.2 Modeling. The Neural network contains three layers. The first and second layers were convolutional layers with Stride 1 and ReLu as the activation functions. Three dimension max polling with 2*2*2 kernel and 2*2*2 strides were applied. The fully connected layer contains 54080*1024 inputs. We used Adam Optimizer and used learning rate 10^-3. A total of 150 epochs were used and the loss stayed stable after 80 epochs.

4 Result

The obtained results are shown in Table 2. Through tuning the hyperparameters, ResNet 50 with XGBoost using the preprocessed data was able to achieve a training error as low as 0.242857. However, the lowest log loss of 0.572531 was achieved using the first ResNet 50 with XGBoost model with data of minimal preprocessing. The highest area under the curve (AUC) of ROC was obtained using Resnet 50 on unprocessed data, with figure 0.602078. The ResNet 50 on preprocessed data, on the other hand, had the lowest AUC out of all the models attempted in the experiment.

Though inception-v3 showed higher training error than both model two of ResNet 50 and model one ResNet 50 with processed data, the log loss of it was lower and it AUC was higher than both of them, indicating a better performance. Though 3D CNN showed the highest training error among all the models attempted here, its log loss was lower than all of the other models except the ResNet 50 with data that were not specifically preprocessed.

	Logloss	error	AUC
ResNet 50 + xgboost_1	0.572531	0.257143	0.602078
ResNet 50 + xgboost_2	0.58568	0.246429	0.547482
(Preprocessed) ResNet 50 + xgboost_1	0.59338	0.242857	0.543048
(Preprocessed) Inception v3 + xgboost_1	0.584733	0.264286	0.571769
3D CNN	0.574796	0.280000	-

Table 2. Results obtained from different models.

5 Discussion

Based on our preliminary experimentations on the dataset using ResNet 50, inception v3, 3D CNN and XGBoost, the best result was obtained using Resnet 50 without data being preprocessed. Yet the highest

AUC of 0.602078 was still considered low. Surprisingly, the 3D CNN trained using three layers had the highest training error, but a better performance than all the other models except the ResNet 50 on unprocessed data. This indicates the models with lower training error were overfitting.

The error rates were all lower than 0.5, which is the chance level. However, it did not mean that these models worked and were better than random guessing, as the dataset had an uneven number of labels. Out of 1397, there were in total 362 cancerous cases, meaning that even if the model made the same guess throughout, the training error could still be as low as 0.259127. In this case, log loss and AUC might be better when evaluating the performance.

One surprising finding is that the model one of ResNet 50 showed better performance when the data were not preprocessed, compared to when the data were processed. The goal of the preprocessing technique used was to resize and normalize the data and reduce noise to some extents, yet the intention was not achieved. This suggests that some information that might be essential to the prediction of cancer was lost during the process. It could also be that the technique was not implemented properly. The decisions about threshold value, image size, o slices depth, might have been poorly made. More experimentations needed to be done to test if the particular processing technique is useful when dealing with CT scans.

The performance of the 3D CNN was unexpected, as it was better than most of the models used, consider the data were in 3D, had been largely downsized, and the network only had two layers. The 3D CNN showed potentials and more experimentations should be done using the 3D CNN to see if fine tuning can bring more performance gain.

Overall, even though the present study did not find a model of high performance, we saw the potential of 3D convolutional neural network and believe that future study should explore more of its potentials.

Github:
https://github.com/joyceheucsd2017/cog181_final

* Bonus Points

We think our work deserves bonus points because 1. The dataset we were using is comparatively large (~150GB). It took a large amount of time to preprocess the data and to train the neural network. 2. The problem we are trying to solve is revolutionary to the medical industry. The technique can be applied not only to detect lung cancer but other cancers as well. 3. The 3D-CNN algorithm is relatively new and is not easy to get demos or sample codes. We had to spend the time to figure it out. 4. We've never used MXnet before, but we had the courage to try it instead of using a platform that we've already know.

6 Reference

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Ghemawat, S. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

[2] Centers for Disease Control and Prevention. (2017). Lung Cancer. Retrieved from https://www.cdc.gov/cancer/lung/

[3] Chae, S. H., Moon, H. M., Chung, Y., Shin, J., & Pan, S. B. (2016). Automatic lung segmentation for large-scale medical image management. *Multimedia Tools and Applications*, *75*(23), 15347-15363.

[4] Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). ACM.

[5] Hamidiana, S., Sahinerb, B., Petrickb, N., Pezeshkb, A., & Spring, M. D. (2017, March). 3D Convolutional Neural Network for Automatic Detection of Lung Nodules in Chest CT. In SPIE Medical Imaging (pp. 1013409-1013409). International Society for Optics and Photonics.

[6] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778).

[7] Hussein, S., Gillies, R., Cao, K., Song, Q., & Bagci, U. (2017). TumorNet: Lung Nodule Characterization Using Multi-View Convolutional Neural Network with Gaussian Process. *arXiv preprint arXiv:1703.00645*.

[8] Kaggle. (2017). Data Science Bowl 2017 [DICOM]. Retrieved from https://www.kaggle.com/c/data-science-bowl-2017/data

[9] Korfiatis, P., Skiadopoulos, S., Sakellaropoulos, P., Kalogeropoulou, C., & Costaridou, L. (2014). Combining 2D wavelet edge highlighting and 3D thresholding for lung segmentation in thin-slice CT. *The British journal of radiology*.

[10] Liu, K., & Kang, G. (2017). Multiview convolutional neural networks for lung nodule classification. *International Journal of Imaging Systems and Technology*, *27*(1), 12-22.

[11] McCormick, M. (2014, October). DICOM Rescale Intercept / Rescale Slope and ITK. Retrieved from https://blog.kitware.com/dicom-rescale-intercept-rescale-slope-and-itk/

[12] n01z3. (2017, February). Mxnet + xgboost baseline [LB: 0.57]. Retrieved from https://www.kaggle.com/drn01z3/data-science-bowl-2017/mxnet-xgboost-baseline-lb-0-57

[13] Python Software Foundation. Python Language Reference, version 2.7. Available at http://www.python.org

[14] Rubin, G. D. (2015). Lung Nodule and Cancer Detection in CT Screening. *Journal of thoracic imaging*, *30*(2), 130.

[15] Sentdex. (2017, January). First pass through Data w/ 3D ConvNet. Retrieved from https://www.kaggle.com/sentdex/data-science-bowl-2017/first-pass-through-data-w-3d-convnet

[16] Shen, W., Zhou, M., Yang, F., Yang, C., & Tian, J. (2015, June). Multi-scale convolutional neural networks for lung nodule classification. In *International Conference on Information Processing in Medical Imaging* (pp. 588-599). Springer International Publishing.

[17] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2818-2826).

[18] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems. In Neural Information Processing

Systems, Workshop on Machine Learning Systems, 2015

[19] van Rikxoort, E. M., de Hoop, B., Viergever, M. A., Prokop, M., & van Ginneken, B. (2009). Automatic lung segmentation from thoracic computed tomography scans using a hybrid approach with error detection. *Medical physics*, *36*(7), 2934-2947.

[20] Yan, X., Pang, J., Qi, H., Zhu, Y., Bai, C., Geng, X., ... & Ding, X. (2016, November). Classification of Lung Nodule Malignancy Risk on Computed Tomography Images Using Convolutional Neural Network: A Comparison Between 2D and 3D Strategies. In *Asian Conference on Computer Vision* (pp. 91-101). Springer, Cham.

[21] Zuidhof, G. (2017, February). Full Processing Tutorial. Retrieved from https://www.kaggle.com/gzuidhof/data-science-bowl-2017/full-preprocessing-tutorial/run/928737