Capstone Project 2 final report: Data Science Bowl 2017

Zexi Yu

January 20, 2018

Abstract

I will write the abstract when I finish editing this report and have the greenlight for submission from my supervisor. Right now this part serves as a reminder area for myself.

1 Introduction

To be added

2 Problem Description

2.1 Problem Definition

The capstone 2 project comes from the kaggle competition "Data Science Bowl 2017" at the following link: https://www.kaggle.com/c/data-science-bowl-2017. The challenge for this year is detecting lung tumor. More specifically, "Participants will develop algorithms that accurately determine when lesions in the lungs are cancerous, using a data set of thousands of high-resolution lung scans provided by the National Cancer Institute." The scans are provided in the format of CT images.

Given the above description, it is clear that for this project, the objective is:

- Build a model that will detect whether a lung cancer occured.

2.2 Potential Benefit for Client

According to the project description,

"This will dramatically reduce the false positive rate that plagues the current detection technology, get patients earlier access to life-saving interventions, and give radiologists more time to spend with their patients."

Because of that, potential clients for this project could be hospitals and health/cancer department of the goverments.

3 Dataset Description

The data can be found at the kaggle competition site: https://www.kaggle.com/c/data-science-bowl-2017/data.

Main files include:

- stage1.7z: This file contains all the CT images for training stage. Total number of patient is close to 1500. Total size for these files is 150 GB.
- stage1_labels.csv: This file contains the cancer label for each patient, two fields here are patient ID and cancer positive/negative
- stage2.7z: This file contains all the CT images for testing stage. Total number of patient is around 500. Total size for these files is 117 GB.

All CT images come in DICOM format.

For images related to a single patient, it usually comes in a number of slices. Each slice is a vertical view of the 3D image, and in different height. The size for each size is 512 by 512 pixels, and number of slices for a single patient is around 200. The operations we are going to perform in data wrangling are all slice basis, meaning we apply them in one slice, and move on to the next slice.

4 Data Wrangling and Observations

This section describes the data wrangling part. Since medical imaging is unstructured data, and we plan to use convolutional neural network in this project, we cannot really perform inferential statistics in this project. So I plan to visualize this part as clear as possible. Please consider this as my compensation for exploratory data analysis.

4.1 Data preparation

DICOM format does not only save the image. It contains other information supporting medical usage, such as pixel size, patient information, HU scaling.

The images are in different scales. So the first and most important task is transforming all images into standard CT measurement: the Hounsfield Unit (HU). HU measures radiodensity. The steps for the transformation is:

- Fill in the missing data out of CT scanning scope by assuming everything there is air.
- Read scaling slope and intercept, then transform the image into HU measurements

4.2 Morphology Operations

After all images are transformed into HU, it is now time to segment the nodes within the lung. We use morphology operations for this task. It is performed on every slices It involves in six steps. I am also going to take a slice from stage 1 as an example and visualize the whole procedure.

- Convert the image into a binary image
- Remove the blobs connected to the border of the image.
- Label the image, and keep the labels with 2 largest areas (right lung and left lung)
- Erosion operation with a disk of radius 2 to separate the lung nodules attached to the blood vessels.
- Closure operation with a disk of radius 12 to keep nodules attached to the lung wall.
- Fill in the holes inside the binary mask of lungs

The figure for each step including the original slice and the superimposition by the final mask:

After all the steps, the overall 3D plot looks like this:

It looks like large nodes in the lung is successfully segmented. Looks good!

4.3 Sacrifices

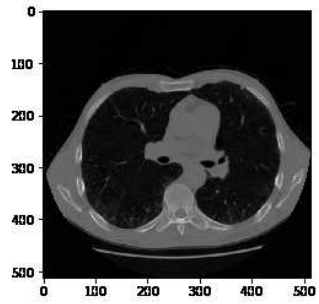
Before we pass the slices to CNN, there are three major sacrifices I make given my limitations on this project. I will list them in small subsections and explain them one by one.

4.3.1 Down Sampling

Since the images are so large, it has to be down sampled giving I only have 4GB RAM on my GPU. For each patient, the image is down-sampled into 50 pixels by 50 pixels by 20 slices. The slices are evenly selected for the down sampling. This operation can surely cause problem for detecting any tumor with diameter less than 6mm, because it may not appear on the down-sampled image at all.

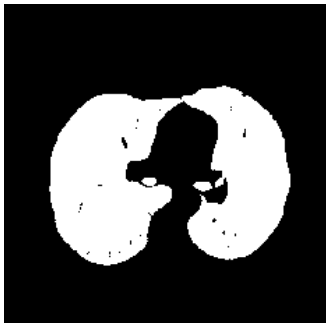
4.3.2 No Further Node Segment

Node segmentation is usually an alternative method for the down sampling. It means further segmenting the nodes from the morphology result, making them into small figures, and then pass it alone to neural network. In this way, algorithms can still detect small tumors, and do not have to handle the whole images at once. This is usually the standard procedure in medical image processing, but I do not perform it here.

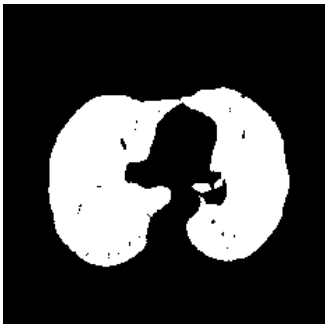


(a) Original Image

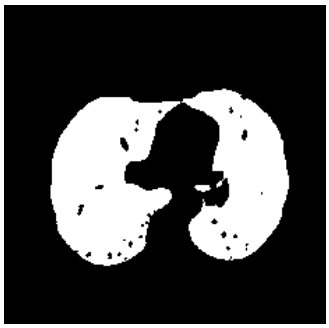
(b) Step 1



(c) Step 2



(d) Step 3



(e) Step 4

(f) Step 5

(g) Step 6

(h) Superimposition

Figure 1: The morphology steps

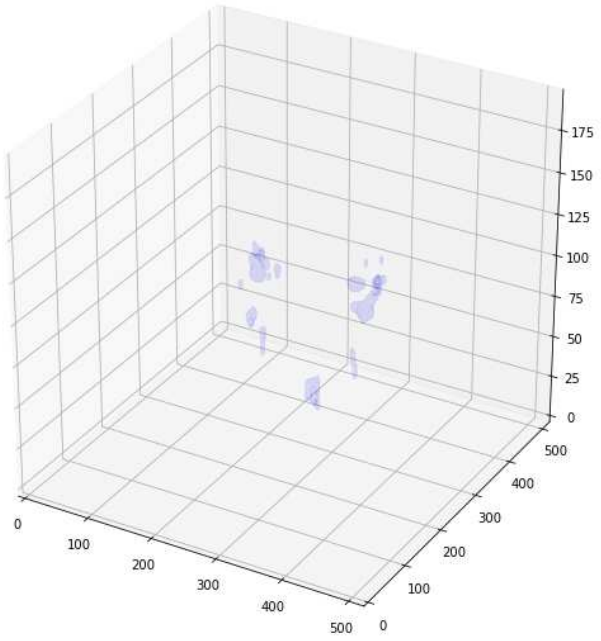


Figure 2: 3D plot for one patient

The reason for me to skip this procedure and use down sampling is that you have to know if each segmented node is cancer or not in order to successfully train the neural network. The top finish for this competition manually labeled every node in the training set. Due to the time limitation for me (because after all this is a capstone project for springboard learning course), I do not perform the manual labeling. But this is definitely in the future work, in order to improve the performance.

4.3.3 Not training on full training set

There are 1500 patients in the training set, and I only used around 240 of them to train the neural network. The reason for it is again the time limitation. In my estimation, it takes 5 days to apply the morphology operations on full training sets. Since I have to perform the operations on each patient on the testing set (stage 2), I cut down training set to make my project finish in time. In the future after I finish the springboard course, I will run the training on all the patients in the training set and update this report.

5 Learning Method: Convolutional Neural Network

After morphology operations, the images are down-sampled and send to convolutional neural network (CNN). CNN is getting more and more attention in deep learning, and is considered as the weapon of choice when dealing with recognition/classification task on images.

When it comes to recognition/classification tasks in image processing, traditionally people looks for features that best describes the characteristic of the image, and then the task become a traditional machine learning problem that can be solved by SVM or decision trees. For example, given pictures of rocks and eggs, the best way to distinguish egg is to find smooth curves for the boundaries. Traditionally, people find curves or other kind of shapes by designing a small image chunk (kernel) based on the shape he/she is looking for, and perform convolution multiplication using this kernel to find the particular shape in the image.

The idea for convolutional neural network is quite simple: Instead of deciding the features ourselves, we let neural network decides the best set of features for us. From the algorithm point of view, that means instead of designing the kernels for particular shape, let the algorithm decide the kernels based on training data.

A typical CNN usually includes multiple convolution layers (conv layers in my figure), each layer contains multiple random initialized kernels and the training data is passed multiple times for the network to decide the best kernels to characterize the classification target. Stacking multiple layers help catching all characteristic from small to large scale (meaning it can catch shapes like circle and the curves on the circle at the same time).

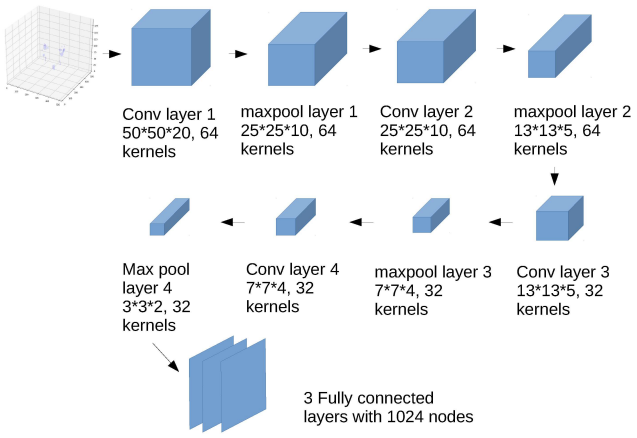


Figure 3: CNN structure

It also includes maxpool layers to down-sample the outcome from previous layers and prevent the algorithm from overfitting. Finally, multiple fully connected normal neural network layers are used to give the classification result.

5.1 CNN Structure

The structure I used for this project is shown in the following figure:

It uses four convolutional layers to fully capture the characteristics in the image from small to large scale, and then use 3 fully connected normal neural network layers to output the result. I also put maxpool layers between every convolutional layers. I also put one dropout layer in the very last to prevent overfitting.

6 Results and Performance

The result constantly pass the brenchmark set for the competition, but only in 1 of the 4 times it will get close to a bronze medal. On average, it ranked around 220 out of 394 teams.

We do not know true label for the images in stage 2. Therefore we cannot know for sure how our algorithm performs. One thing for sure is that it definitely overfits. I observe several times during the training period where the loss on training set drops to zero (indicating serious overfitting). On the other hand, given the cancer rate in the training set is around 30%, it is quite possible that the true cancer rate in the testing set is 30% as well. Compared to that, the

10% cancer rate given by our CNN algorithm suggests we missed a lot of cancer cases, either because of overfitting, or because of the lost of small tumors caused by the downsampling.

7 Future works

The future work definitly contains two points I mentioned in the "Sacrifice" subsection. They are:

- Train the CNN using all the training set instead of 1/5 of them and see if the performance improves.
- Manually label the nodes in the training set, perform node segmentation and train and change CNN structure according to the single node images.

8 Conclusion

To be added