Deliverable Three

Cancer, a disease caused by an uncontrolled division of abnormal cells in a part of the body. To be more specific, this paper will be in reference to lung cancer. This paper will serve as our team's initial project analysis for this semester's project. According to the Center for Disease Control and Prevention (CDC) more people in the United States die from lung cancer than any other type of cancer, both men and women alike. Roughly 225,000 people in the United States die every year from lung cancer which accounts for $12 billion in health care costs.

The initial problem we wish to address, and hopefully help resolve, is a two-step issue. Starting with the issue of early detection. Currently, the detection of the cancer tends to mainly occur when the patient already has cancer, in various stages. This poses an issue, because roughly 20% of the patients who would die due to lung cancer would have been saved due to early detection. Additionally, health cost is another issue to resolve. Utilizing a program to help detect potential lung cancer patients would greatly reduce the expenses for the patient's and the overall burden for their families.

The target user for this program is of course a doctor, more specifically, an oncologist. An oncologist is a medical practitioner qualified to diagnose and treat tumors. Currently, oncologists utilize a computerized tomography (CT) scan, which allows the oncologist to view the patient's lungs. Due to the sheer number of individuals who die of lung cancer each year in the United States alone, the number of CT scans that must be viewed by the oncologist, no matter the result found (normal or otherwise), could reduce the response time of actually removing the cancer. Especially those individuals that could be found at an early stage of cancer, but gets seen at a later date, potentially allowing the cancer to worsen.

As specified in the previous paragraph, current methods allow needless risk to the patients due to potential missed opportunities by the doctor, that is, the current methods run the risk of not detecting lung cancer early and therefore, increase cost to the patient and decrease the patient's life expectancy. With the aid of the program we are attempting to create, we hope to reduce or completely eliminate this risk. Overall improving the likelyness of the patient's longevity and financial burden due to health care costs.

The use of the program would allow the user, the oncologist, to utilize their time more effectively. Meaning, the program would serve as the assistant or aid in reading the CT scans that come in on a day to day basis. The program would go through each new scan and determine potential lung cancer risks. Instead of the oncologist taking the time to view each and every CT scan that comes in, the program will show only the scans to the oncologist in which it believes a patient runs a high risk for lung cancer. Allowing the oncologist to review the scan and respond more effectively. Additionally, due to the faster response time of the oncologist, the health care cost that normally would accumulate for those going for treatment, those already diagnosed with cancer, over and over again would be greatly reduced.

In regards to context of use and relevancy to the project, it would be safe to say that the end user, the oncologist, would be the first to be contacted after the programs completeness. When it comes to interactions between user and program, it would be held to a minimum. Remember, we are trying to improve the response the patient receives. Having too many interactions would not resolve the already current response issue. The only time an interaction should occur between user and program is when the program sends the user scans of potential patients that are at high risk. Otherwise, the program would grab the data it requires through the means of internal software the hospital, specialist center, or what have you, currently utilizes to store patient information. Along with the patient's information, the program should also be linked to the CT scanner in order to receive the CT scans for processing.

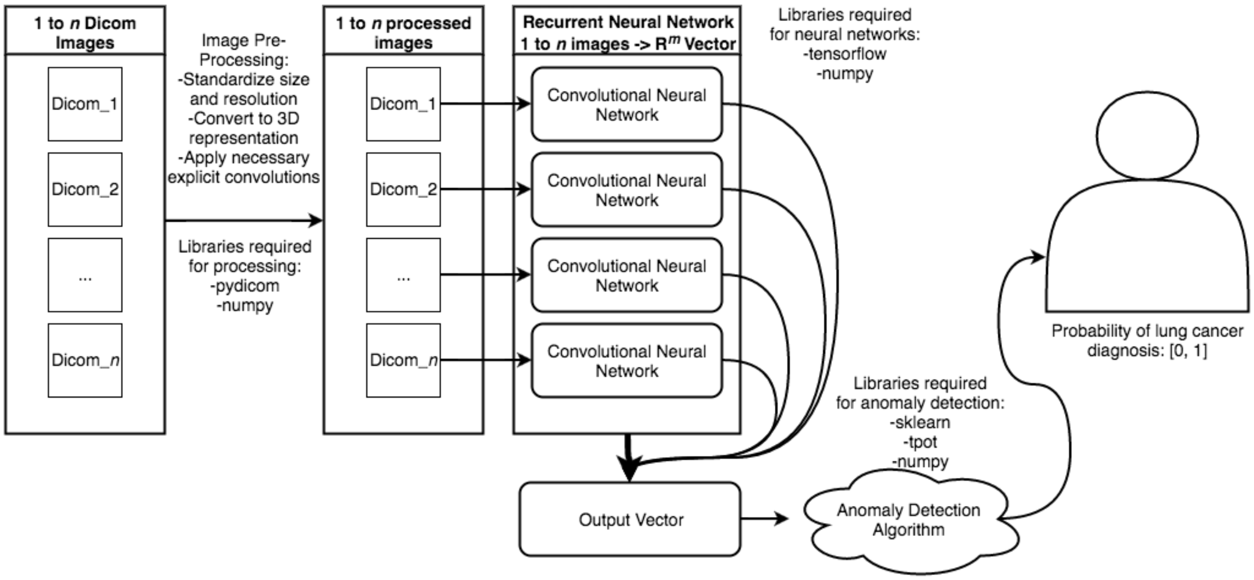
There is an event currently in progress known as the Data Science Bowl 2017. To obtain the data or knowledge base for the project we will be utilizing the current data already stored at https://www.kaggle.com/c/data-science-bowl-2017. The data consists of CT scans that show normal lungs and scans that show abnormal lungs, which will allow for our team to train our program in determining what a normal is versus an abnormal lung.

The general contours of the classification will follow the diagram in ***figure 1***, proceeding in four phases.

1. It will receive an ordered collection of dicom images, the output format of a CT scan. Sample images are of variable resolution, black-white/color, orientation, and size. Further, dicoms are a specialized format allowing for the 2D representation of a 3D scan. In the pre-processing phase, images will be standardized and converted to a 3D representation.
2. Next, in the computer vision phase, each image will pass through a Convolutional Neural Network (CNN). This may have explicitly programmed convolutions or dynamically generated convolutions depending on experimental performance. This CNN will output a high dimensional vector for input to a Recurrent Neural Network (RNN).

3. This RNN will take 1 to *n* CNN outputs, looking for evidence of anomaly over time.
4. Depending again on experimental performance, this RNN may output a scalar probability the patient is diagnosed with lung cancer in the next 12 months or may output a high dimensional vector. This output vector then becomes input to a fourth phase classifier, which will be selected through demonstrated experimental performance.

Figure 1: General System Diagram


1 to *n* Dicom Images
Dicom_1
Dicom_2
...
Dicom_*n*
Image Pre-Processing:
-Standardize size and resolution
-Convert to 3D representation
-Apply necessary explicit convolutions
Libraries required for processing:
-pydicom
-numpy
1 to *n* processed images
Dicom_1
Dicom_2
...
Dicom_*n*
Recurrent Neural Network 1 to *n* images -> R^m Vector
Convolutional Neural Network
Convolutional Neural Network
Convolutional Neural Network
Convolutional Neural Network
Libraries required for neural networks:
-tensorflow
-numpy
Output Vector
Libraries required for anomaly detection:
-sklearn
-tpot
-numpy
Anomaly Detection Algorithm
Probability of lung cancer diagnosis: [0, 1]

For development, this project will require extensive use of available deep learning libraries primarily available in Python 3.5.2+ and used widely in industry. Libraries and their respective phases are marked above in ***Figure 1***. Specifically, they will include:

- Pydicom, for dicom image processing
- Tensorflow, for large scale machine learning on heterogeneous distributed systems
- Numpy, a tensorflow dependency and ultra-efficient Python array implementation
- Sklearn, a high-level machine learning library for rapid prototyping

- Tpot, for using genetic algorithms to optimize a machine learning pipline

GPU-accelerated hardware will be necessary for efficient experimentation, as convolutional neural networks run orders of magnitude slower on CPU-only hardware. Our group has sufficient remote access to such hardware and has begun initial testing for proof of concept, achieving an accuracy of 63% on a very limited CNN implementation after only one hour of training. Working across local and remote development environments will require meticulous use of version control, done through GitHub.

For deployment, delivery conditions will dictate environment requirements. Ideally, the final implementation should run in an environment nearly identical to development. However, it is possible to output all optimized parameters and reconstruct the mathematical model in any programming language, absent all libraries except some form of dicom processing. This is not recommended.