**Lung Cancer Detection**



**Team: Inventix**

**Team Members: Sabrish Ramamoorthy and Rajani Maurya**

* **Under Guidance of Professor Chaiyaporn Mutsalkisana**

Contents

[1 Introduction: 3](#_Toc480756454)

[1.1 Problem Statement: 3](#_Toc480756455)

[1.2 Proposed Solution: 4](#_Toc480756456)

[2 Dataset: 4](#_Toc480756457)

[3 Python Libraries: 5](#_Toc480756458)

[4 Approach and Implementation: 6](#_Toc480756459)

[4.1 Data preprocessing and generating 3D image of lung from given DICOM files for each patient 6](#_Toc480756460)

[4.2 Selecting an algorithm to train data 8](#_Toc480756461)

[4.3 Validation on test data and Accuracy 17](#_Toc480756462)

[5 Conclusion: 19](#_Toc480756463)

[6 Future Implementation: 19](#_Toc480756464)

[7 References & Citations: 19](#_Toc480756465)

# 1 Introduction:

## 1.1 Problem Statement:

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. One year ago, the office of the U.S. Vice President spearheaded a bold new initiative, the Cancer Moonshot, to make a decade's worth of progress in cancer prevention, diagnosis, and treatment in just 5 years. National Cancer Institute has provided high resolution lung scans of patients. The project aims to develop algorithm that will determine if the lesions in the lungs are cancerous. This will further help in early detection of cancer and will give more time to save patient’s life.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

The above image A is the X-Ray image of lungs, image B is the CT scan and image C is the MRI scan. As it is seen CT scan is better at determining the presence of lesions.

**Difference between X-ray, CT scan and MRI scan:**

X-rays are a type of radiation, and when they pass through the body, dense objects such as bone block the radiation and appear white on the x-ray film, while less dense tissues appear gray and are difficult to see. X-rays are typically used to diagnose and assess bone degeneration or disease, fractures and dislocations, infections, or tumors.

Organs and tissues within the body contain magnetic properties. MRI, or magnetic resonance imaging, combines a powerful magnet with radio waves (instead of x-rays) and a computer to manipulate these magnetic elements and create highly detailed images of structures in the body. Images are viewed as cross sections or “slices” of the body part being scanned. There is no radiation involved as with x-rays. MRI scans are frequently used to diagnose bone and joint problems.

A computed tomography (CT) scan (also known as CAT scan) is similar to an MRI in the detail and quality of image it produces, yet the CT scan is actually a sophisticated, powerful x-ray that takes 360-degree pictures of internal organs, the spine, and vertebrae. By combining x-rays and a computer, a CT scan, like an MRI, produces cross-sectional views of the body part being scanned. In many cases, a contrast dye is injected into the blood to make the structures more visible. CT scans show the bones of the spine much better than MRI, so they are more useful in diagnosing conditions affecting the vertebrae and other bones of the spine

Reference: http://www.sworthogroup.com/general-orthopedic-faq/25-what-is-the-difference-between-x-rays-mri-and-ct-scan

## 1.2 Proposed Solution:

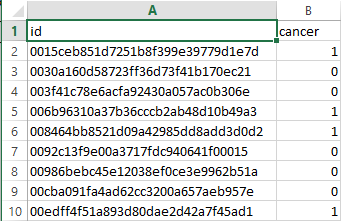
The solution that we propose is to use a neural network that can learn the existence of lesions from the given CT scans. We will train the data and perform validation by splitting the data in proportion of 80% (train data) and 20% (test data). After doing data preprocessing, we will determine which neural network will suit our needs. Then we will work on accuracy by changing number of epochs, number of hidden layers, changing image resolution and changing number of images per person. This will help to achieve our objective of predicting what the likelihood of a patient having lung cancer is.

Reference: https://www.kaggle.com/c/data-science-bowl-2017

# 2 Dataset:

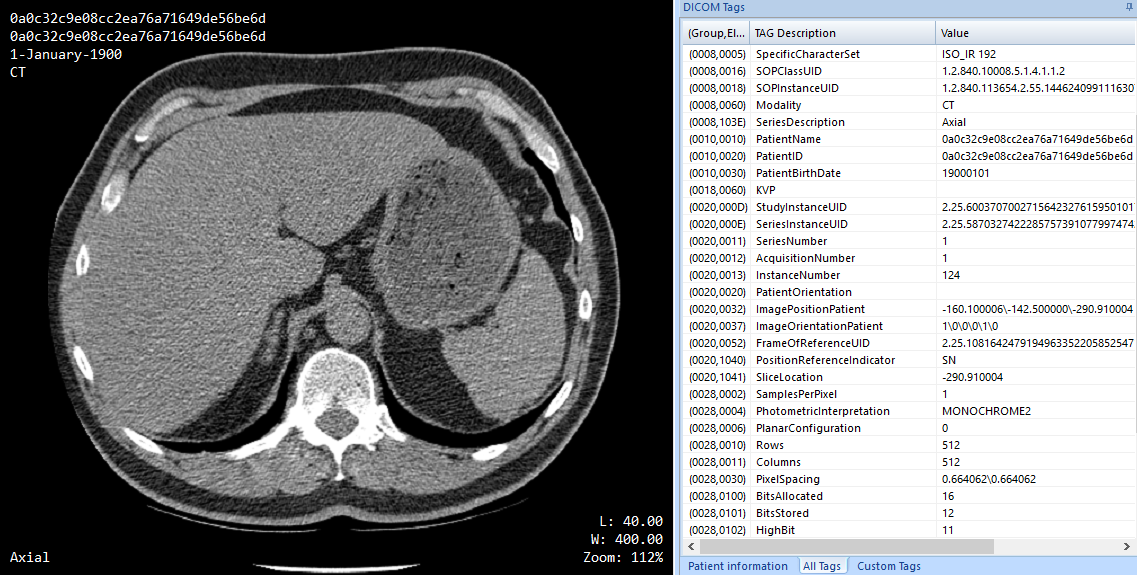
The dataset consists of 1595 patient’s lung 2D CT scans (200 GB). Each patient’s data has around 150 – 190 Digital Imaging and Communications in Medicine (DICOM) images. DICOM is a standard for handling, storing, printing, and [transmitting](https://en.wikipedia.org/wiki/Data_transmission) information in [medical imaging](https://en.wikipedia.org/wiki/Medical_imaging). The dataset has label.csv file which has the actual result as ‘1’ and ‘0’ for training and testing purpose. ‘1’ indicating patient has cancer and ‘0’ indicating patient has no cancerous lesions.

Stage1\_labels.csv



Reference: https://www.kaggle.com/c/data-science-bowl-2017

Sample DICOM image:



Reference: https://www.kaggle.com/c/data-science-bowl-2017

Each DICOM image has metadata attached to it. For example the above image has attribute of ‘Rows’ and ‘Columns’ that indicate the pixels of image. The ‘ImagePositionPatient’ has ‘X’, ‘Y’ and ‘Z’ attribute of the axial view of lungs. This stack of 2D scans with respect to the Z attribute of ‘ImagePositionPatient’ will help to generate 3D view of lungs of that patient.

# 3 Python Libraries:

* Python 3
* Numpy
* Pandas
* Dicom
* Os
* Matplotlib
* Cv2
* Tensorflow

Reference: <https://www.kaggle.com/sentdex/data-science-bowl-2017/first-pass-through-data-w-3d-convnet>

# 4 Approach and Implementation:

Let’s try to understand the approached used to solve the problem,

## 4.1 Data preprocessing and generating 3D image of lung from given DICOM files for each patient

The actual scan, when loaded by dicom, is not just some sort of array of values, instead it's got attributes. We're sorting by the actual image position in the scan. When put these together to get a full 3D rendering of the scan we see image is of size 195 x 512 x 512. That's huge! So, we need to resize the data. Also, if we notice the dimension of all the patient’s dicom, they are not of the same size.

E.g.:

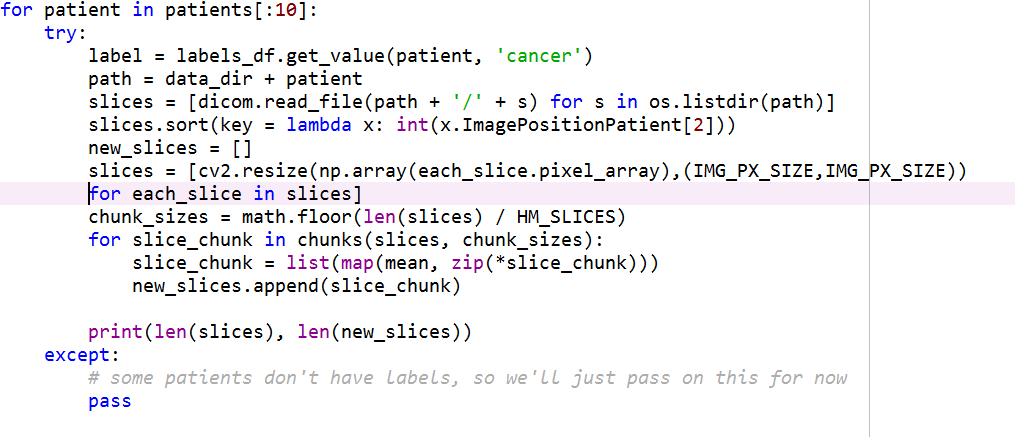
(512, 512) 177

(512, 512) 110

(512, 512) 126

So, we need to down sample the data and make the depth uniform.

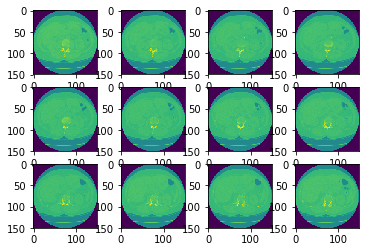
We used openCv library to resize the image.



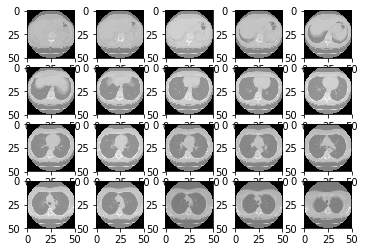
Reference: <https://www.kaggle.com/sentdex/data-science-bowl-2017/first-pass-through-data-w-3d-convnet>

The above code sort all slices of 2D Dicom images based on the z position and then resize the data into a 50x50 image with the help of ‘cv2.resize’ function. After resizing the image, it then combines all the slices of patient’s image into 20 images. We do this step so that all the patients images are of the same size(50x50x20).

After resizing we could see all the image are of size 50x50x20

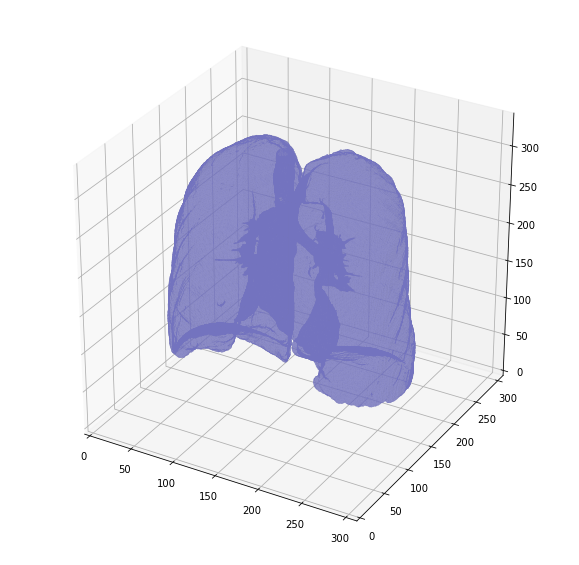
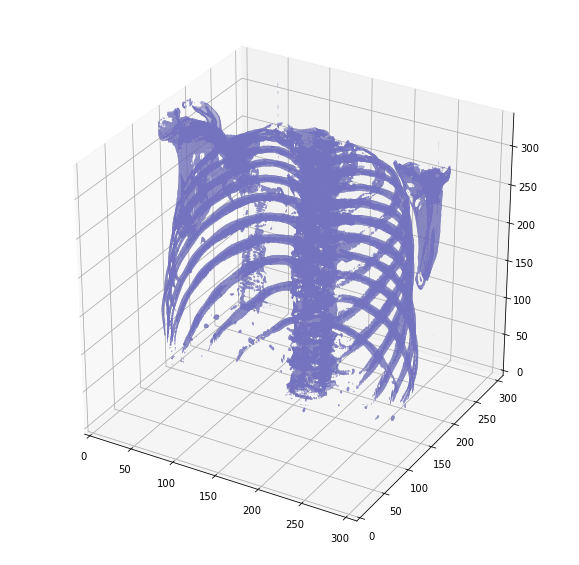


One major issue with the above images are colors and ranges of data. The colors are not same; it has all the shades of green. Since we represent images in bit matrix, the bit value of images is different, it will be very difficult to train our neural network. So, we do a grayscale colormap in the ‘imshow’,



Reference: <https://www.kaggle.com/sentdex/data-science-bowl-2017/first-pass-through-data-w-3d-convnet>

When we construct the 3D images with the help of preprocessed 2D slices it looked like below,



Reference: <https://www.kaggle.com/gzuidhof/data-science-bowl-2017/full-preprocessing-tutorial>

Now we preprocess all the data into one file, training should likely be faster, so we can more easily tweak our neural network and not be processing the data the same way over and over.



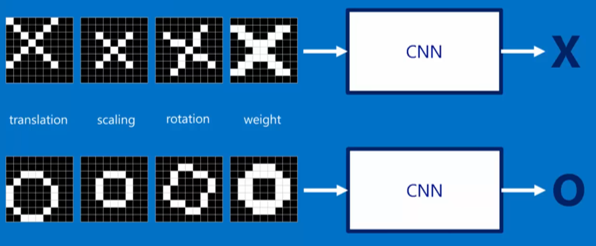
## 4.2 Selecting an algorithm to train data

Now we have got the preprocessed data ready to be fed to our neural network. Since we must train images, we use convolution neural network.

**What is Convolution Neural Network?**

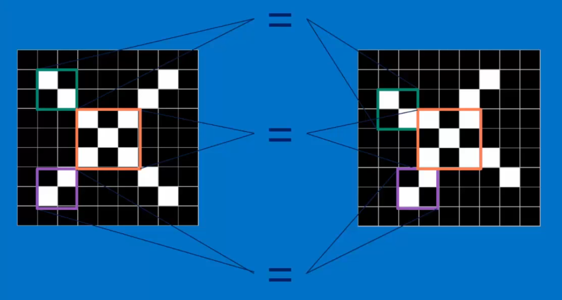
Convolutional neural networks (CNNs) consist of multiple layers of receptive fields. These are small neuron collections which process portions of the input image. The outputs of these collections are then tiled so that their input regions overlap, to obtain a higher-resolution representation of the original image; this is repeated for every such layer. Tiling allows CNNs to tolerate translation of the input image. Convolutional networks may include local or global pooling layers, which combine the outputs of neuron clusters. They also consist of various combinations of convolutional and fully connected layers, with pointwise nonlinearity applied at the end of or after each layer. A convolution operation on small regions of input is introduced to reduce the number of free parameters and improve generalization. One major advantage of networks is the use of shared weight in convolutional layers, which means that the same filter (weights bank) is used for each pixel in the layer; this both reduces memory footprint and improves performance.

* CNN Input



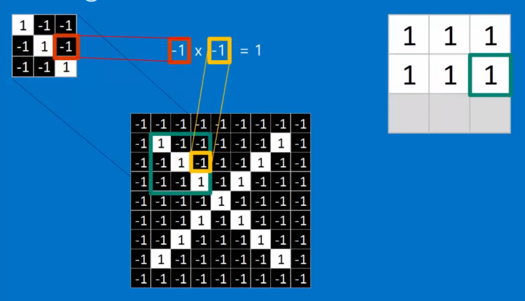
Reference: <https://www.youtube.com/watch?v=FmpDIaiMIeA>

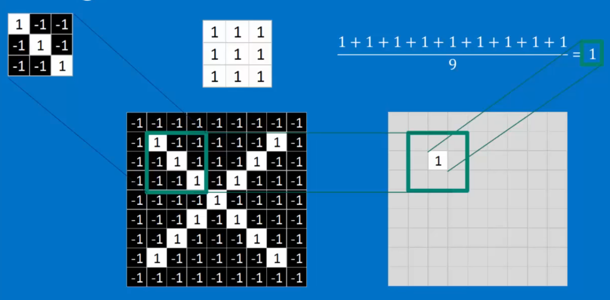
* CNN match pieces of image



Reference: <https://www.youtube.com/watch?v=FmpDIaiMIeA>

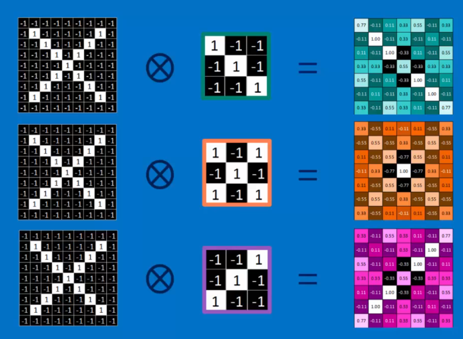
* Filtering





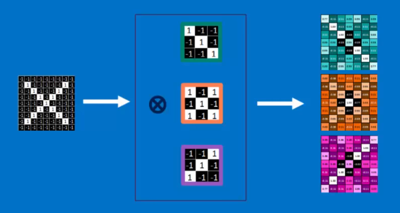
Reference: <https://www.youtube.com/watch?v=FmpDIaiMIeA>

* Convolution : trying every possible match



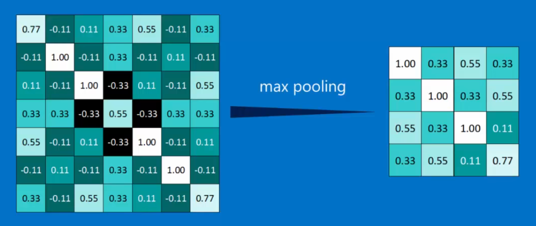
Reference: <https://www.youtube.com/watch?v=FmpDIaiMIeA>

* 1 Image is stack of filtered images



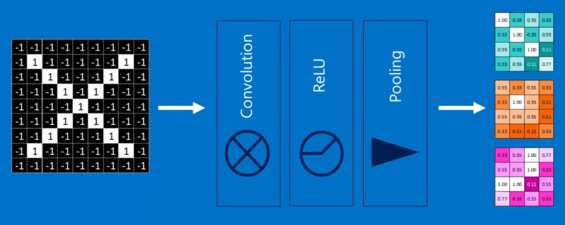
Reference: <https://www.youtube.com/watch?v=FmpDIaiMIeA>

* Pooling



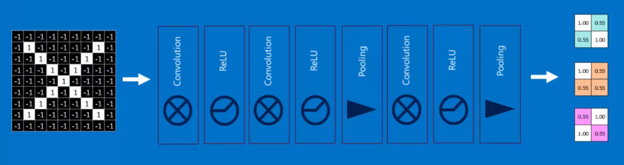
Reference: <https://www.youtube.com/watch?v=FmpDIaiMIeA>

* Layers get stacked



Reference: <https://www.youtube.com/watch?v=FmpDIaiMIeA>

* Layers can be repeated or switched – Deep stacking



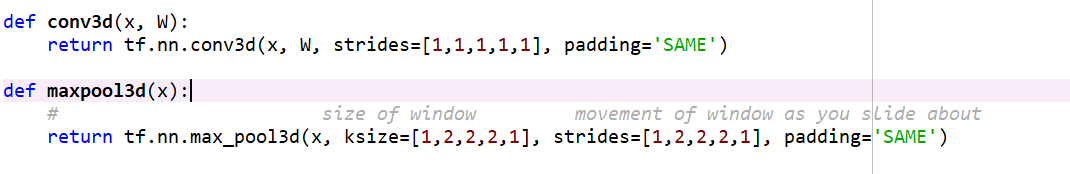
Reference: <https://www.youtube.com/watch?v=FmpDIaiMIeA>

* Fully connected



Reference: <https://www.youtube.com/watch?v=FmpDIaiMIeA>

Since we are dealing with 3D images, we use conv3D library of TensorFlow.

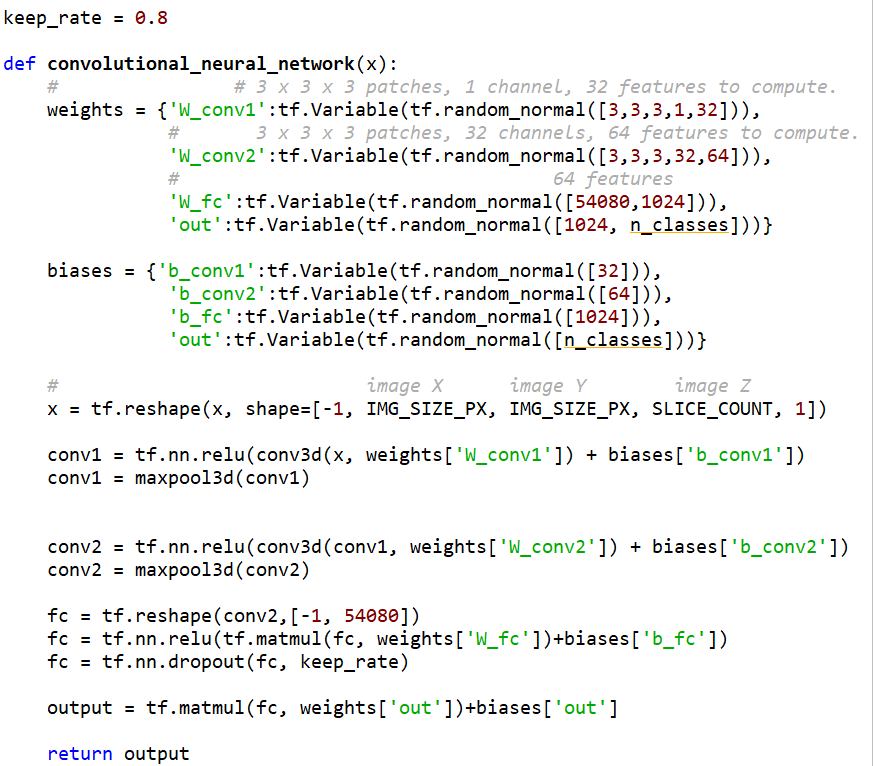


Reference: <https://www.kaggle.com/sentdex/data-science-bowl-2017/first-pass-through-data-w-3d-convnet>

Stride controls how the filter convolves around the input volume. The stride is set at 1 for conv3d. Stride is normally set in a way so that the output volume is an integer and not a fraction. The filter is shifted by 1 units and the output volume shrinks as well.

Padding "SAME" tries to pad evenly left and right, but if the number of columns to be added is odd, it will add the extra column to the right, as is the case in this example (the same logic applies vertically: there may be an extra row of zeros at the bottom)

Now we are ready for the Network,



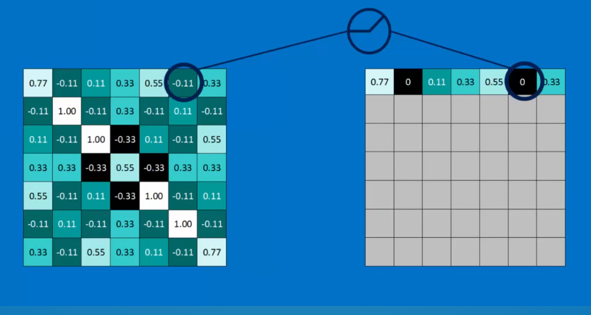
Reference: <https://www.kaggle.com/sentdex/data-science-bowl-2017/first-pass-through-data-w-3d-convnet>

Since the image is represent in RGB and with 32 bit. We used weight as [3,3,3,32]

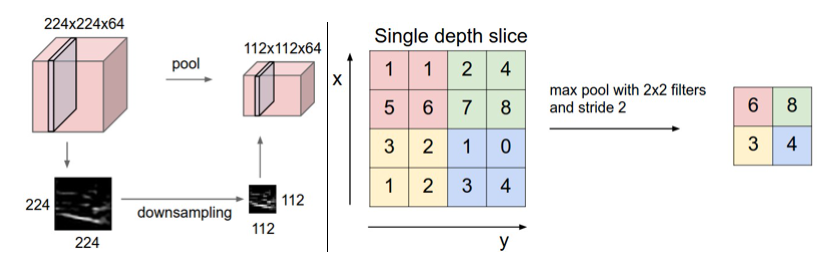
After testing the model with 2,3 and 4 layer, our model performed well with 2 convolution layer

After ‘conv3D’ We added RELU layer. The purpose of this layer is to introduce nonlinearity to a system that basically has just been computing linear operations during the conv layers (just element wise multiplications and summations). The ReLU layer applies the function f(x) = max(0, x) to all of the values in the input volume. In basic terms, this layer just changes all the negative activations to 0. This layer increases the nonlinear properties of the model and the overall network without affecting the receptive fields of the conv layer.

We have added dropout to avoid overfitting of data.



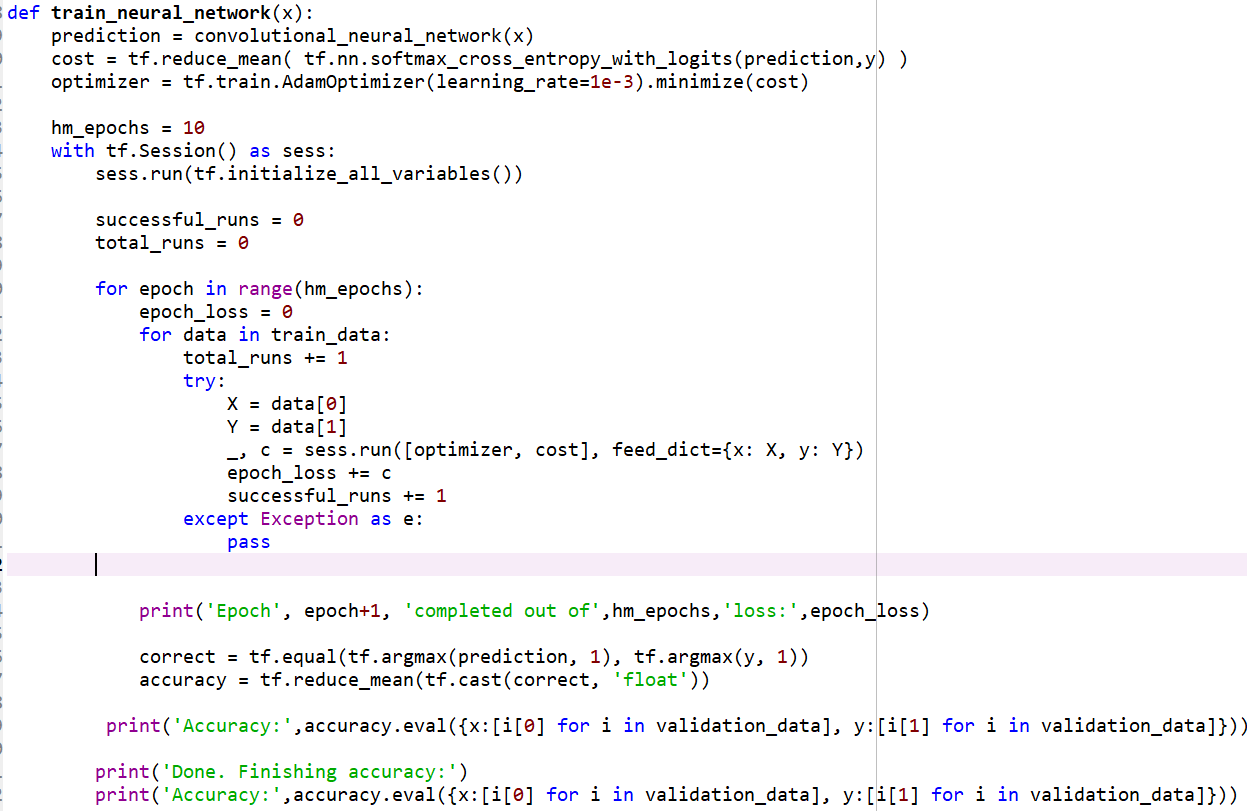
Reference: <https://www.youtube.com/watch?v=FmpDIaiMIeA>



Reference: http://cs231n.github.io/convolutional-networks/

Pooling layer down samples the volume spatially, independently in each depth slice of the input volume. Left: In this example, the input volume of size [224x224x64] is pooled with filter size 2, stride 2 into output volume of size [112x112x64]. Notice that the volume depth is preserved. Right: The most common down sampling operation is max, giving rise to max pooling, here shown with a stride of 2. That is, each max is taken over 4 numbers (little 2x2 square).

Now we train the data,



The above code explains that we used 10 epochs to train our model.

Reference: <https://www.kaggle.com/sentdex/data-science-bowl-2017/first-pass-through-data-w-3d-convnet>

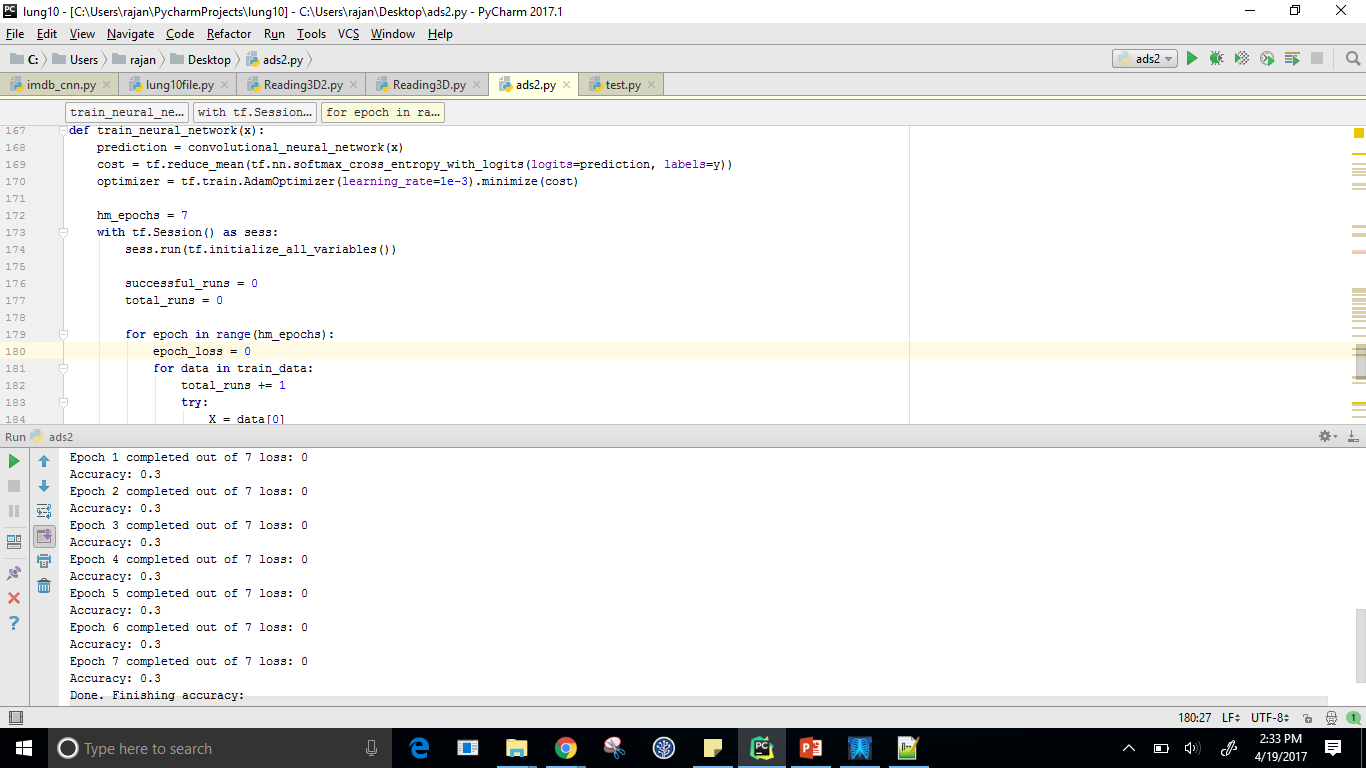
## 4.3 Validation on test data and Accuracy

Since the data size is huge. We trained our model with 80 patient images and tested our model on 20 patient images. After 10 epochs, we got an accuracy of 70%.

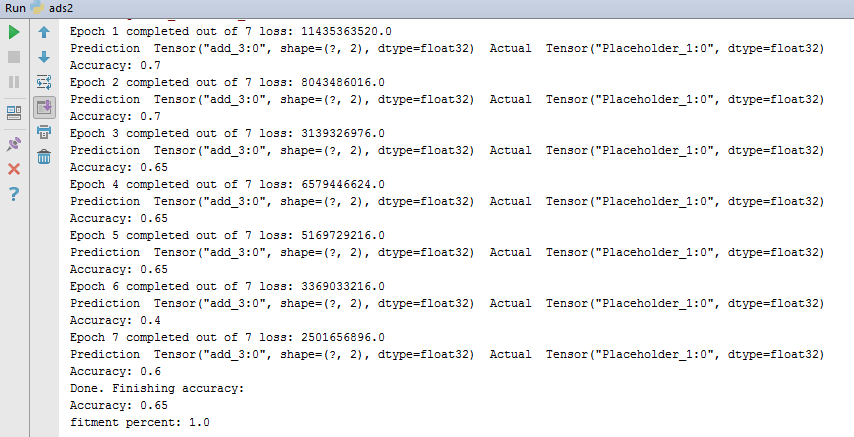
But to predict accurately whether a patient has a lung cancer or not, our model should give an accuracy of more than 90%.

How could we have improved the model is explained under topic ‘Future Implementation’.

Accuracy with 3 hidden layers – 30%:



Accuracy with 2 hidden layers – 65%:



# 5 Conclusion:

The given approach is just the beginning and can be made better. There may be other approaches and thus we conclude that it is possible to detect lung cancer at an earlier stage. However this process requires lots of existing data to better train the algorithm.

# 6 Future Implementation:

* The above implementation should be trained on more dataset to improve accuracy.
* Due to system limitations, we were not able to check few cases like increasing number of epochs, increasing number of hidden layers, increasing image size. These cases should be tested using Microsoft Azure or ResNet.
* Improving algorithm to reduce losses.

# 7 References & Citations:

* <https://en.wikipedia.org/wiki/DICOM>
* <https://www.kaggle.com/c/data-science-bowl-2017>
* <https://www.kaggle.com/sentdex/data-science-bowl-2017/first-pass-through-data-w-3d-convnet>
* <https://www.kaggle.com/gzuidhof/data-science-bowl-2017/full-preprocessing-tutorial>
* <https://www.youtube.com/watch?v=FmpDIaiMIeA>
* [www.quora.com](http://www.quora.com/)
* <http://cs231n.github.io/convolutional-networks/>
* <https://www.youtube.com/watch?v=FmpDIaiMIeA>
* <https://www.youtube.com/watch?v=e2sGq_vI41s>
* <http://www.sworthogroup.com/general-orthopedic-faq/25-what-is-the-difference-between-x-rays-mri-and-ct-scan>
* <http://cs231n.github.io/convolutional-networks/>
* Google Images