Lung Cancer Nodule Detection using Low Memory Deep Neural Networks

Gauri Pradhan [1] **Sailalitha Gollapudi** [2] **Yuwei Wang** [3]

Abstract

In this report, deep learning architectures will be applied on the dataset from Lung Nodule Analysis 2016 which consists of 3D CT scan of patients. The initial task is to use segmentation techniques to detect nodule images from non-nodule images. Data Preprocessing and Data Augmentation constitute two most important steps in the Image preprocessing. Our project aims to compare performance of various low memory, light- weight deep neural net (DNN) architectures for biomedical image analysis. This will involve using networks including vanilla 2D CNN, U-Net, 2D SqueezeNet, 2D MobileNet for binary classification in order to detect the presence of lung cancer in patient CT scans of lungs with and without early stage lung cancer.

1. Introduction

Lung Cancer (or *lung carcinoma*) is one of the most common maladies around the world.An accurate lung cancer classifier could speed up and reduce costs of lung cancer screening, allowing for more widespread early detection and improved survival.

2. Data Description

Our primary dataset is the Lung Image Database Consortium image collection (LIDC-IDRI)[6] consists of diagnostic and lung cancer screening thoracic computed tomography (CT) scans with marked-up annotated lesions. The annotations and associated candidates files are provided to us in .csv format. The dataset itself has images as *.mhd* files (512 x 512). The characteristics of the data are detailed below:

- For each patient the data consists of CT scan data and a label (0 for no cancer, 1 for cancer).
- The candidates file is a csv file that contains nodule candidate per line. Each line holds the scan name, the x, y, and z position of each candidate in world coordinates, and the corresponding class.
- The annotation file is a csv file that contains one finding per line. Each line holds the SeriesInstanceUID of the scan, the x, y, and z position of each finding in world coordinates; and the corresponding diameter in mm. The annotation file contains 1186 nodules.

3. Data Preprocessing

We are using scipy, SimpleITK tool kit alongside PIL (Python Imaging Library) for image processing tasks including scaling, nodule detection (essential to identify the tumor(s) associated with lung cancer). We have done downsampling, segmentation, normalization, and zero-centering. The dataset has the ground truth coordinates and the radii of the nodules within each scan.

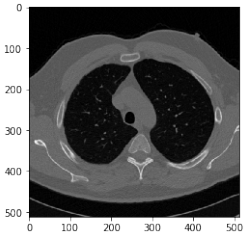
3.1. Downsampling

The dataset being used for analysis happens to contain only **1351** positive samples which constitutes only 0.2451% of the available dataset . This is indicative of heavy imbalance in the dataset. Analyzing the unbalanced dataset yields sub-optimal results since the algorithm will predict no failure for every example. The Type 1 error is unacceptably high. Type 1 error occurs when a failure occurs but the model predicts no failure (also known as a false negative). **Down-sampling** creates a balanced dataset by matching the number of samples in the minority class with a random sample from the majority class. We built a new dataset with **1351** the positive cases and add to it a set of **5404** randomly samples negative cases (4 times the number of positive samples) thus, giving us a dataset which comprises of 20.0% positives and 80% negatives. Still the dataset is unbalanced though the imbalance is not as substantial as before. This imbalance will be further addressed in the section of Data Augmentation.

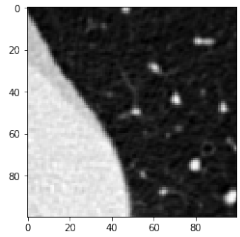
3.2. Thresholding and Segmentation

First, every scan was rescaled so that every voxel represented a volume of 1x1x1 mm. In the tomography images given in the dataset, the segmentation has to be done to handle and detect nodules and masses that were hidden near the edges of the lung tissue. This had to be done carefully as any sub-optimal process can cause these to be removed

from the images leaving no chance for the nodule detector to find. The preprocessing involves masking the regions which are not lungs, also known as the process of *thresholding* . The Hounsfield Units (HU) which are a measure of radiodensity, we will be masking all regions in the scan which are not lung tissue. This unit of measure for lung tissue is -500 HU whereas that of air is -1000 HU, bone is around 700 HU and other tissues and blood have a 0 HU. With CT scans the pixel intensities can be expressed in HU and have semantic meaning. All intensities were clipped on the minimum and maximum interesting hounsfield value. HU, So we mask out pixels that are close to 1000 or above 320 to leave lung tissue as the only segment. After thresholding each CT scan, we create 2D slices of size 512 x 512 only for the images have the nodules present in them (referring to the co-ordinates of nodules given in the candidates.csv file).
proposed plan is to input these images directly into the neural network. However, such large size of images is not suitable hence, it was reduced. The following images showcase the ways images have been reduced in size while segmenting out the region around the nodule. This area specific segmentation is performed using the data annotated in the file candidates.csv which provides the co-ordinates position of nodule in each scan.The final image size that is fed into the model is 100 x 100.

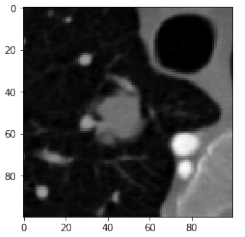


(a) Initial 2D Slice with 512 x 512 size

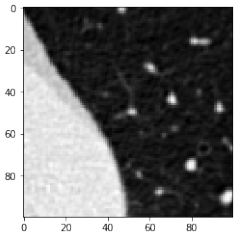


(b) Sub-image of the slice with pixels reduced to 100 x 100

Figure 1. Sub- sampling of CT scan images based on HU for a negative sample of patient #600



(a) Positive

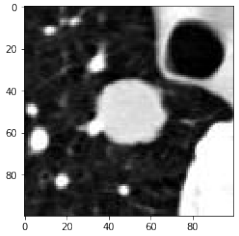


(b) Negative

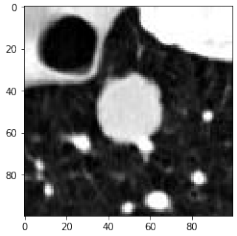
Figure 2. Segments of positive and negative samples

3.3. Data Augmentation

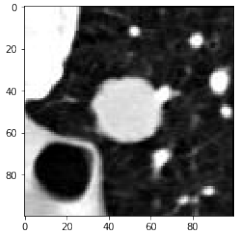
In section 2.1, we dealt with the class imbalance using downsampling. To further resolve the issue, we chose another popular technique of *data augmentation*. We create new images by applying minor transformations onto the images dataset obtained after processing through section 2.2. Minor changes such as flips or translations or rotations alters the image such that our neural network would interpret these as distinct images. The purpose of data augmentation is dual, it compensates for the lack of relevant data as well as help us build a model that is robust towards orientation, location, scale, brightness etc. We account for these situations by training our neural network with additional synthetically modified data. An example of augmented images added to the dataset is illustrated in Figure 3.



(a) Original segment of size 100 x 100



(b) Rotating 90 degrees to create new image



(c) Rotating 180 degrees to create new image

Figure 3. data augmentation of segment image of patient # 3285

3.4. Data Splitting

Once the dataset imbalance is minimized, we can split the data into training and test set based on an 80-20 split. Further the training data is split using an 80-20 split to yield a training and validation set (for cross-validation and hyperparameter tuning of the chosen model). From the pickle files (used to store labels and address of image files) for train, validation and test data, images are extracted and converted into *.jpg* format and storing them in HDF5 files for further use in keras/tensorflow processing.

4. Model Selection

In medical imaging, most widely used architectures include U-Net + ResNet/ U-Net + AlexNet which are heavy on the system's menory. As opposed to these conventional preferences, it is our intention to experiment with the lightweight networks for medical image analysis. We're not using any of the available pre- trained networks as our data differs vastly from the ImageNet used to train these models.

4.1. Baseline CNN model

As described in the Figure (4),we are using a baseline Vanilla CNN using 3 convolutions layers. Each layer of CNN has a stride 1 , since the nodule is defined by the transition from white to dark patch , we believed that stride would be useful to identify this transition. A weighted softmax cross-entropy loss calculated is used, as a label of 0 is far more common than a label of 1 .

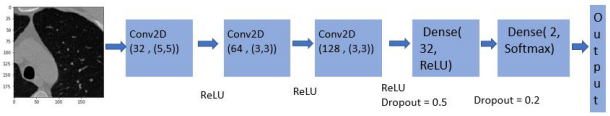
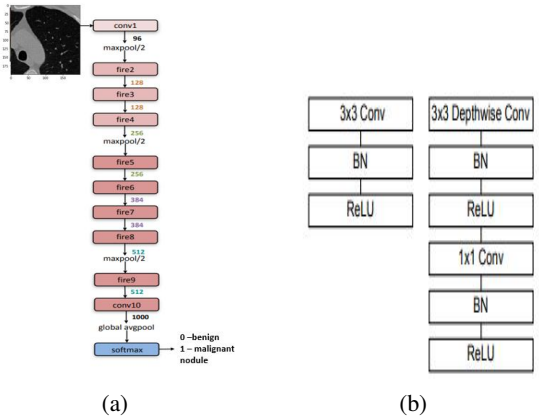


Figure 4. Block diagram depicting architecture of the vanilla 2D CNN

Class weights are hyper- parameters subjected to experimentation. Experimenting the class weights on the Vanilla CNN model , weights= $[0.4, 0.6]$ provided the best results. The architecture of the Baseline Convolutional Neural Network is detailed abve. Since this network has 2 fully connected layers, it takes longer time for the model to train on GPU. Optimizer used - Adam with learning rate 0.001. The best accuracy achieved with this model on validation set was 89.932.

4.2. SqueezeNet

It has long been experimented to train heavy networks such as AlexNet , ResNet for nodule detection.With equivalent accuracy, smaller CNN architectures offer at least two advantages: (1) Smaller CNNs require less communication across servers during distributed training. (2) Smaller CNNs are more feasible to deploy on FPGAs and other hardware with limited memory such as mobile phones.[1] Hence, it served as our motivation to use low memory network such a SqueezeNet for classification. We have chosen to ignore the max pooling layer since our image is relatively smaller. The SqueezeNet is similar to AlexNet but involves some changes to the latter's architecture including: (1) Replace 3x3 filters with 1x1 filters . (2) Decrease the number of input channels to 3x3 filters. In order to decrease the parameters we have to decrease the inputs to the 1x1 filters. (3) Design **Fire Modules** which consist of squeeze (1x1 filters) layer and expansion layer (3x3 filters). This is the macro architecture of CNN employed within the framework SqueezeNet is shown in Figure (5).



(a) (b)

Figure 5. Functional unit for (a)Fire module architecture for SqueezeNet (b) Fundamental functional unit of the MobileNet

4.3. MobileNet

Table 1. MobileNet Architecture

Layer	Filter shape	Input Size
Conv	(3, 3, 3 , 32)	(224 , 224 , 3)
Conv dw	(3 , 3 ,32 dw)	(112 , 112 , 32)
Conv	(1, 1 , 32 , 64)	(112 , 112 , 32)
Conv dw	(3 , 3 , 64 dw)	(112, 112, 64)
Conv	(1, 1 , 64 ,128)	(56 ,56, 64)
Conv dw	(3 ,3,128)dw	(56 , 56 , 128)
Conv	(1 , 1 ,128, 1280	(56 , 56 ,128)
Conv dw	(3,3,128) dw	(56,56, 128)
Conv	(1,1,128,256)	(28,28,128)
Conv dw	(3, 3,256) dw	(28 ,28, 256)
Conv	(1, 1 ,256 ,256)	(28 ,28 , 256)
Conv dw	(3,3,256) dw	(28,28,256)
Conv	(1,1,256,512)	(14,14, 256)
Conv dw	(3,3,5120 dw	(14,14,512)
Conv	(1, 1 ,512,512)	(14 , 14 , 512)
Conv dw	(3 ,3,512) dw	(14, 14, 512)
Conv	(1,1,512,1024)	(7, 7,512)
Conv dw	(3,3,1024) dw	(7,7 ,1024)
Conv	(1,1,1024,1024)	(7,7, 1024)
Avg Pool	(Pool, 7 , 7)	(7 ,7 , 1024)
FC / s1	(1024,1000)	(1 , 1 , 1024)
Softmax / s1	Classifier	(1, 1, 1000)

Depth-wise Separable Convolution (dw) is used to reduce the model size and complexity. It is particularly useful for mobile and embedded vision applications.Depth-wise convolution is the channel-wise spatial convolution , followed by Point-wise convolution actually is the 1 x 1 convolution to change the dimension.It is noted that Batch Normalization (BN) and ReLU are applied after each convolution.The depth-wise separable convolution splits this into two layers, a separate layer for filtering and a separate layer for combining. This factorization has the effect of drastically reducing computation and model size.[8]

5. Analysis

In medical tests sensitivity is the extent to which actual positives are not overlooked (so false negatives are few), and specificity is the extent to which actual negatives are classified as such (so false positives are few).

Table 2. Evaluation Metrics

Model	Accuracy	Specificity	Sensitivity	AUC score
Baseline CNN	89.921	0.877	0.838	0.9209
SqueezeNet	85.212	0.883	0.874	0.9019
MobileNet	86.315	0.765	0.916	0.8919

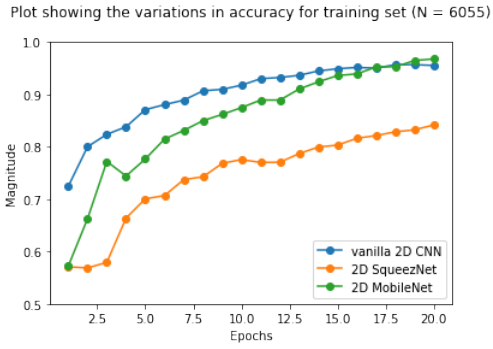


Figure 6. Training Plots over 20 Epochs

Apart from the plots in Figure (6), (7), (8), we also tested each model on our test data which the model hasn't seen before. The results were: 0.8478 (vanilla CNN) $>$ 0.8324 (MobileNet) $>$ 0.8296 (SqueezeNet). It is also necessary in medical applications to be concerned with not only the Accuracy as a metric but also Specificity(Sp) and Sensitivity(Se). These two metrics are of significance since people with malignant cancer should be never be miss-classified as a 'Safe' Patient. Therefore, the best model will be contingent upon the Sp, Se. Observing the area under the ROC curve of a test can be used as a criterion to measure the test's discriminating ability, i.e. how good is the test in a given clinical situation.

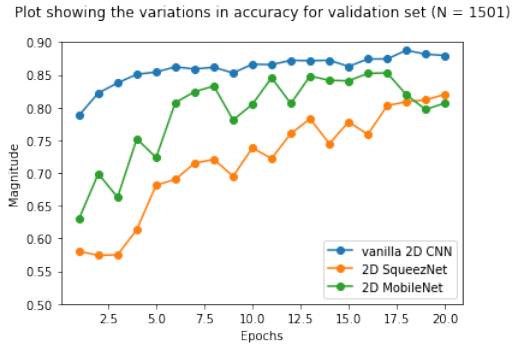


Figure 7. Validation Plots over 20 epochs

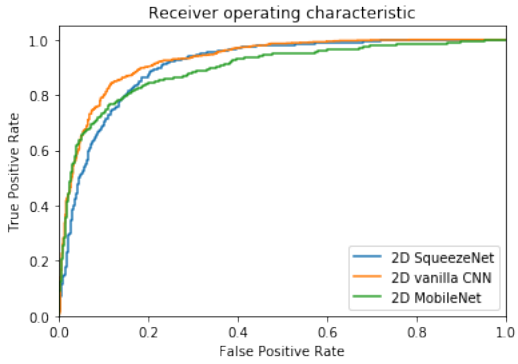


Figure 8. receiver operating characteristics (ROC) curve

6. Conclusion

Although it is for the vanilla CNN achieves higher accuracy on the test data, it is not an ideal choice as it takes longer to train and it's Sp-Se trade- off is comparable to SqueezeNet. Furthermore, a higher accuracy of the MobileNet doesn't make it a favourable choice due to it's poor Sp-Se trade-off. Hence, we decide that for the given data, SqueezeNet is the best model given the faster computational time and good Sp-Se trade- off along with satisfactory accuracy. We believe these results are significant as SqueezeNet and MobileNets aren't widely renowned networks for analysis of biomedical images.

7. Future Work

We aim to extend the project further to adapt these deep learning architectures to predict using 3D volumetric slices. Moreover, the bright pixels usually corresponded with the location of cancerous nodules, so it could be possible to extend our current model to determine the exact location of the cancerous nodules. A part of the future work will also involve experimentation with various segmentation processes (K- means, Watershed) to improvise the 2D models.

References

1. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and $<$ 0.5MB model size Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Keutzer

2. Cruz, Joseph A., and David S. Wishart- Applications of Machine Learning in Cancer Prediction and Prognosis. Cancer Informatics, (January 2006). doi:10.1177/117693510600200030.

3. Exploring 3D Convolutional Neural Networks for Lung Cancer Detection in CT Volumes, Shubhang Desai, Stanford.

4. M. Firmino et al., Computer-aided detection system for lung cancer in computed tomography scans: Review and future prospects, BioMedical Engineering OnLine, vol. 13, p. 41, 2014.

5. Lung Nodule Detection using a Neural Classifier , M. Gomathi and Dr. P. Thangaraj, IACSIT International Journal of Engineering and Technology, Vol.2, No.3, June 2010.

6. LUNA16, Lung nodule analysis 2016. *https:// luna16.grand-challenge.org/.*

7. Julien Di Wit- 'Preprocessing for Medical Imaging with Dicom Data'.

8. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications Andrew G. Howard Menglong Zhu Bo Chen Dmitry Kalenichenko Weijun Wang Tobias Weyand Marco Andreetto Hartwig Adam Google Inc.