Lung Cancer Detection and Classification - a Deep Learning Approach

Abhinav
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
aa18@illinois.edu
&
Dipali Ranjan
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
dipalir2@illinois.edu

December 14, 2018

Abstract

Lung Cancer is one of the most common cancers in US with over 225,000 cases and $12 billion in health care costs every year. Current diagnostic methods include biopsies and imaging. Early detection of lung cancer (detection during the earlier stages) significantly improves the chances for survival. In this project we want to explore the task of detecting lung cancer given patients CT scans of lungs. We use LUNA 16 dataset for this problem along with the nodule annotations. We perform a combination of pre processing techniques, lung segmentation and nodule detection on the data. We use features collected to train XgBoost and SVM classifiers and pre-trained Neural Networks such as ResNet. For improved nodule detection, we train U-NET from 3 directions (X, Y, Z) and compare our results with a 3D convolution network of Resnet18.

1 Introduction

Lung Cancer causes 150,000 deaths with over 225,000 cases every year in US. Almost 17% of the lung cancer cases diagnosed survive five years after the diagnosis. Early detection of lung cancer is improves chances of survival but is also difficult to detect due to fewer symptoms. [1] The diagnostic performance and survival rate can be increased by improving computer-aided diagnosis.

Current detection methods include low-dose CT (computed tomography) used by radiologists to look for tissue growths such as nodules that may become cancerous. This detection methodology poses the following problems - early stage nodules are small, and nodules difficult to discern between benign/cancerous with naked eye. This makes diagnosis for radiologists extremely difficult, requiring longer experience especially with low resolution of CT scans leading to higher false positives.

Medical Imaging diagnosis for such cases can be divided into two tasks - Nodule detection and Binary classification of Cancerous/Benign nodules. With the current success of deep learning techniques in image classification techniques, networks have been used to extract high level features from images using transfer learning. The success of Convolutional Neural Networks (CNNs) in object detection tasks can be extended to nodule detection and classification tasks.

Our proposed method for this classification task uses a combination of deep learning and computer vision techniques. In order to classify whether a patient has cancer, we need to transform the given CT scans into an input space that can be interpreted by the deep learning models. We hope to use computer vision techniques to reduce our input space to emphasize on possible nodules and reduce noise from surrounding tissues, bone, air. We train a neural network to improve nodule detection based on additional information given about nodule location. Such a system can should be able to detect large ($\geq$ 10 mm) and small nodules ($\leq$ 3 mm). This would not only help in improving our overall accuracy but such a system could be superior than a human diagnosis.

2 Background

Deep learning based lung nodule diagnostic system generally follow the following steps - 1. Image preprocessing 2. Nodule Candidate generation 3. Malignancy prediction for each nodule 4. Malignancy prediction for overall CAD image. In order to achieve this, it is required to find representative imaging biomarkers at a macroscopic level in chest CT images. [6] Several studies have used statistical methods such as Histogram of Oriented Gradients, [14], Gabor [7] and Local Binary Pattern (LBP) [13]. For more deep learning based feature extraction, large data set is required to train an entire Convolutional Network from scratch (with random initialization). Instead, it is common to pretrain different CNNs models on such as: ConvNet [11], VGG16 [15], MobileNet [9], Xception [5], Inception V3 [16], and then use the either as an initialization or a fixed feature extractor for the task of interest. Some of the existing prediction models such as work by Shen et al [13] who used Multi-scale convolutional neural networks for lung nodule classification, use of texture features for computer-aided diagnosis [13] and other works. Another related work uses Risk stratification of lung nodules using 3d cnn-based multi-task learning by Hussein et al [10] and the use of radiomics to quantify of tumour phenotypes using large image data presented by Hugo et al [4]. We will compare our model with some of the existing models described above.

3 Preprocessing and Modelling Approaches

3.1 Data Description

We use the LIDC-IDRI dataset which consists of diagnostic and lung cancer screening thoracic computed tomography (CT) scans with marked-up annotated lesions [2]. This dataset has 1018 patient cases, where each case includes CT scans and annotations file (XML format). The annotation file consists of radiologist marked nodules classified as one of the three categories - Nodule $\geq$ mm, Nodule $\leq$ 3 mm," and Non-Nodule $\geq$ 3 mm. Table 1 shows the detailed logistics of the data. For each patient we have their patient ID and whether they have cancer or not (0 / 1). Each CT scan can have multiple slices of images which varies for every patient (100 - 400 axial slices). Each slices is of 512 × 512 pixels in DICOM format. We use the *pyDicom* to read these files. We divide this data into training set of size 814, validation set of size 101, and test set of size 103.

Collection Statistics	updated 3/21/2012
Number of Patients	1010
Number of Patients	1308
Number of Series	1018 CT, 290 CR/DX
Number of Images	244,527
Image Size (GB)	124

Table 1: Detailed Data Description [10]

3.2 Data Preprocessing

3.2.1 Lung Thresholding

Using the metdata provided by *dcm* format, we transform the loaded scans into Hounsfield Unit (HU) for easier thresholding. HU units are a measure of radiodensity with Air -1000 HU, Lung tissue -500 HU, bone

700 HU, water and blood 0 HU. We transform the slices of each CT scan by stacking them applying the transformation. Fig 3 shows the distribution of HU units in a CT scan.

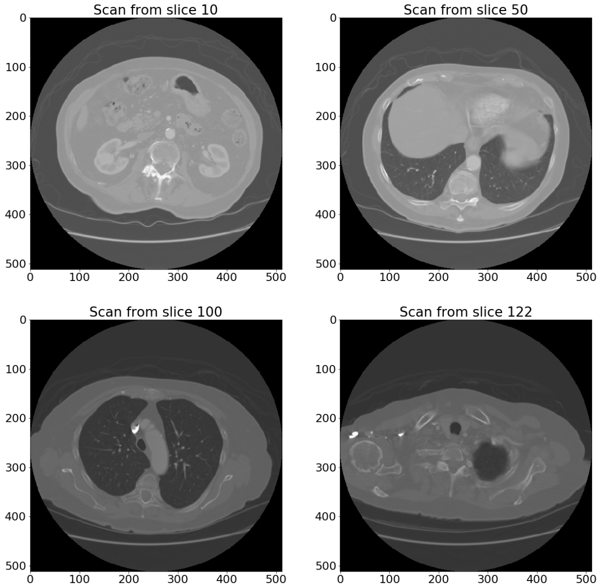

Figure 1: Scan from different locations in a CT scan for a patient

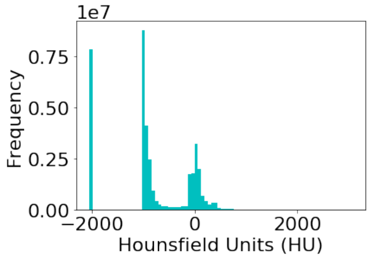

Figure 2

We segment our images at -320 HU as a result we mask out pixels that are close to 1000 or above 320 such that only lung tissue is remaining. To incorporate possible edge cases of nodules in aerial bronchioles, we keep the segments with air. We built a mask using above thresholding and connected components ($skimage.label$). In this way, two pixels are connected when they are neighbors and have the same value. In 2D, they can be neighbors either in a 1- or 2-connected sense. This is followed by a series of dilation morphological operations ($dilation,erosion$) to expand the mask in all direction using a circular kernel.

3.2.2 Resampling

The dataset has varied number of axial slices for every CT Scan for each patient. Spacing represents how much each of the axial slices are far apart from the next slices for every CT Scan. The spacing remains constant for every CT scan of a patient but varies among patients. In order to make the representation of every voxel in input space uniform we perform resampling. A resampling factor is calculated by using pixel spacing and given pixen spacing $[1mm \times 1mm \times 1mm]$.

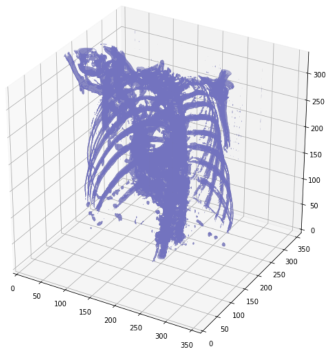

Figure 3: Sample patient 3D image with pixels values ≥400 HU reveals the bone segment.

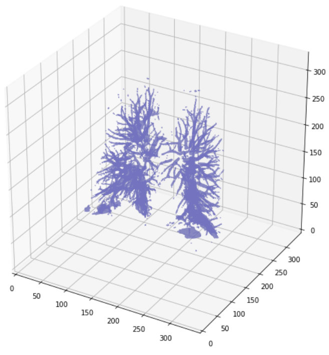

Figure 4: Sample patient bronchioles within lung

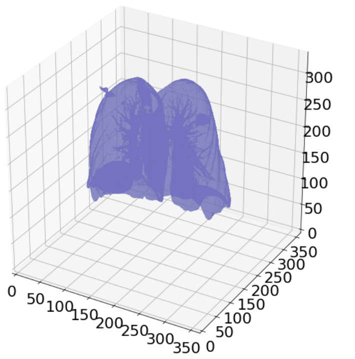

Figure 5: Sample patient final mask in which bronchioles are included

4 Training Architectures and Methodology

We wanted to use Neural Nets to input smaller regions of interest instead of the entire processed 3D images. To accomplish this task, we use 3D Resnet18 and Multi-directional UNet to find top nodule candidates to train our classifier. We train UNet and extract features to train classifier. Resnet50 (Lower layers Fine Tuning) is used to extract low level features and train a classifier.

4.1 3D Resnet18

This method is based on 3D Resnet18 architecture. The architecture is described in figure 6.
Here we extract 3D voxels from the scans preprcocessed above. The size is (64,64,64). We generate two kind of samples, positive and negative. The positive samples are the one where a nodule is actually present. The negative samples are randomly picked from a non-nodule location. The nodule location file is used for this generation. This is now fed into the 3D Resnet18, we employ multiple data augmentation techniques to artificially increase the size of our dataset, this includes random horizontal flip, random rotations, random controlled jitter and random crop. The labels here are specific to the patient having a cancer or not. We use overlapping filters at each step to get the final feature space at the fully connected layer. The extracted features from the fully connected layer will be fed to the classifier at the second stage which will be discussed in a later section.

4.2 Multi-dimensional UNet

Using the processed image from preprocessed image into heavy neural net proved to be a chaalenge in getting good results. Firstly, despite pre processing the size of data didn't reduce significantly. Secondly, the input space to such a network is sparse resulting in poor detection of nodule by classifiers. Inspired from the success of image segmentation in [12], we train a Multi-dimensional Unet architecture to improve our nodule detection. This architecture is shown in figure 9, while the stripped down Unet is shown in figure 8.

The architecture trains 3 separate modified Unets along three directions X,Y,Z of the CT scan images. Fig shows three axial slices. This is done in order to train a 2D network with stacked 2D pre-processed slices. Using 3D data from 3 slices we hope to reduce the sample space and compare how a 2D Net architecture performs on 3D data.

We designed a stripped down version for U-Net in order to reduce the memory usage and the amount of training time for the Unet. As we can see from figure 8 the input to Unet-X, Unet-Y, Unet-Z are 96 × 96 2D CT slices, along with the labels from annotations file. These labels are 96×96 masks where nodule pixels are 1, rest of the pixels are set to 0.

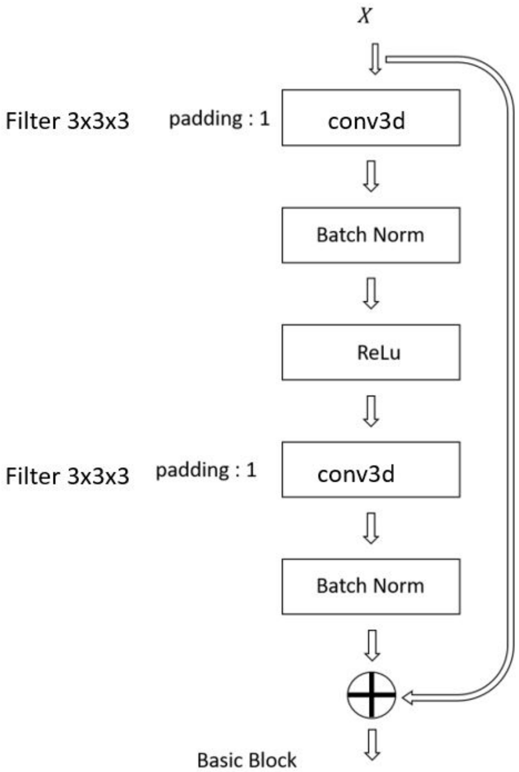

Figure 6: Basic block used for Resnet18

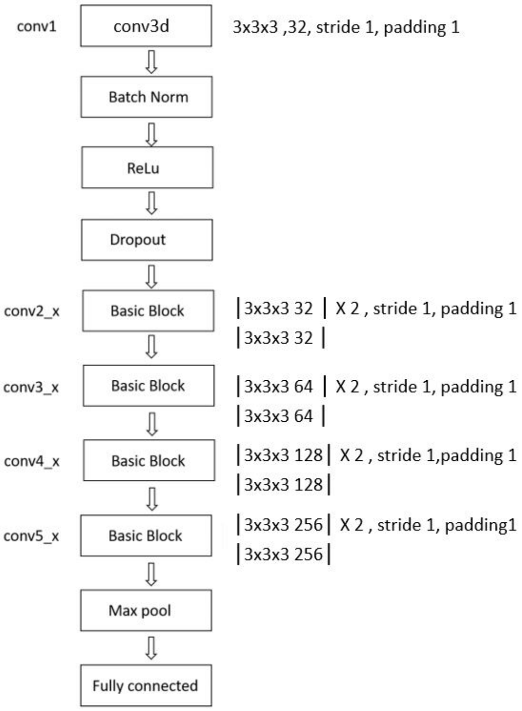
conv1
conv3d
3x3x3 ,32, stride 1, padding 1
Batch Norm
ReLu
Dropout
conv2_x
Basic Block
|3x3x3 32 | X 2 , stride 1, padding 1
|3x3x3 32 |
conv3_x
Basic Block
|3x3x3 64 | X 2 , stride 1, padding 1
|3x3x3 64 |
conv4_x
Basic Block
|3x3x3 128| X 2 , stride 1,padding 1
|3x3x3 128|
conv5_x
Basic Block
|3x3x3 256| X 2 , stride 1, padding1
|3x3x3 256|
Max pool
Fully connected

Figure 7: Block diagram describing 3D Resnet18 along with parameters

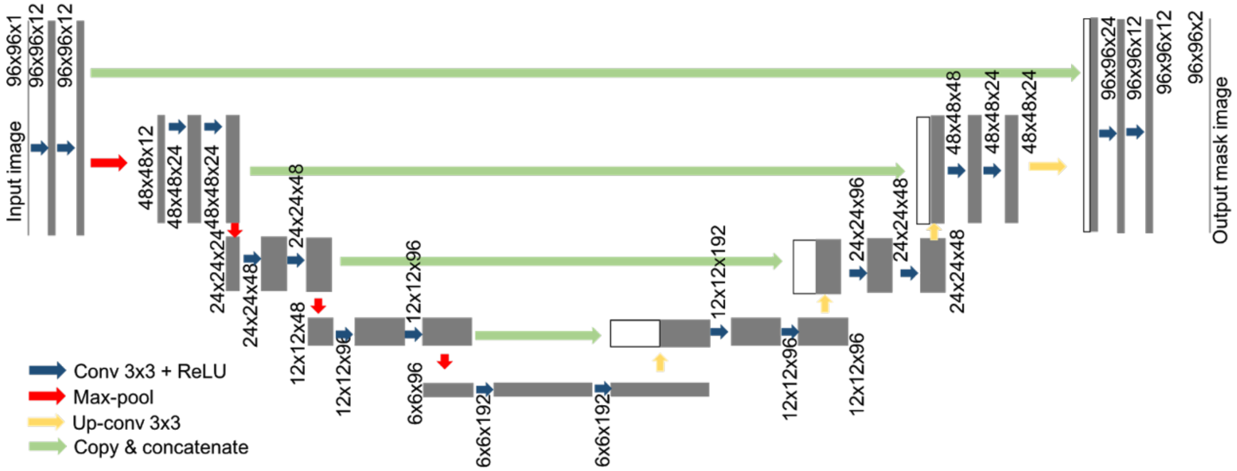

Figure 8: Base toned down Unet used for Unet-X,Y,Z

The output for the each Unet-X,Y,Z is 2D slice with all pixel values between 0 and 1, indicating the probability of a pixel being a nodule. These trained Unets are then applied to the 3 axial slices of CT images along X,Y, Z directions, where we extract features for 10 biggest nodules from the binary segmentation mask. These features include the location, area, volume, HU statistics for the nodules. After feature extraction the output is sent to a classifier (SVM+xgBoost) for lung cancer classification on the test set. Here we perform patient wise detection to get the label.

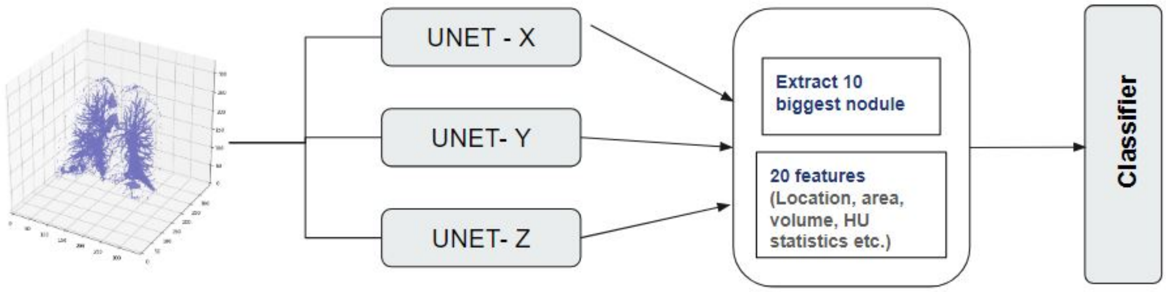
UNET - X
UNET- Y
UNET- Z
Extract 10 biggest nodule
20 features (Location, area, volume, HU statistics etc.)
Classifier

Figure 9: Multidirectional Unet for 3 way training along different axial slices

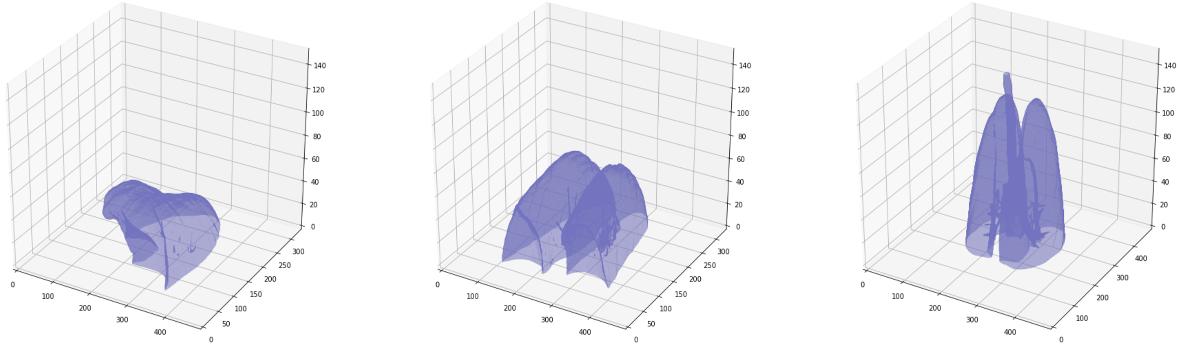

Figure 10: A reconstruction of slices taken in 3 (X,Y,Z) directions used to train U Net (L-R)

4.3 Resnet50 - Lower Layer fine tuning

As we know that different layers of a CNN learn different features. The first few layers learn lower level features while as we progress further the last level layers learn more higher level semantic features. Usually in literature people replace last layers to fine tune a model for a particular dataset for transfer learning. But since lower level feature extraction becomes more important as we migrate to completely unrelated datasets in this case CT images from Imagenet data. So with this intution we fine tuned REsnet50's [8] first few layers while keeping other features same. We also replace the last softmax layer size from 1000 to 2 for binary cancer classification. This is shown in figure 11.

4.4 Classifier

For analyzing the extracted features from 3d Resnet18 and MD-Unet we used 2 classifiers namely - SVM and xgBoost. We use gaussian kernel for SVM. We do a random parameter search optimize our classfiers. In next section we report their performance.

5 Results

Below we report the various hyper paramerter we tuned, the computation hours we used and the final results compared to some benchmarks.

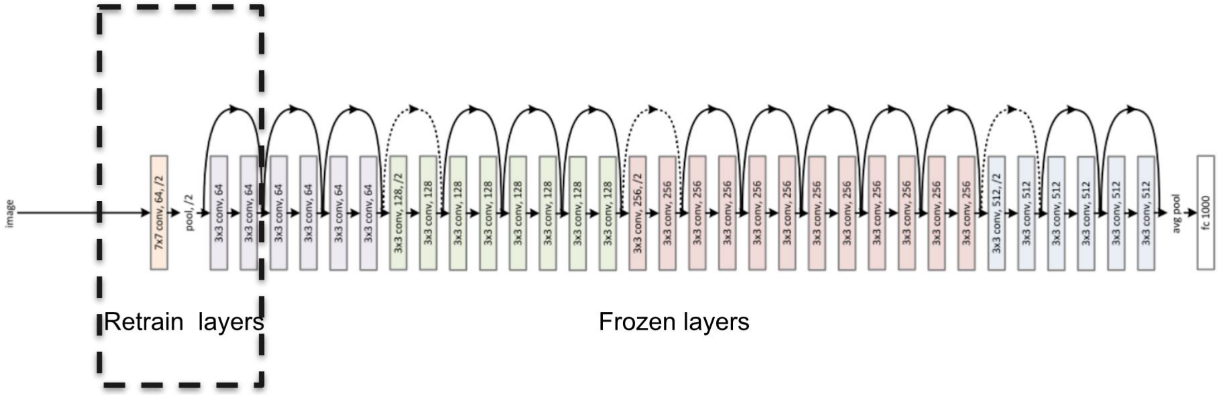

Figure 11: fine tuning resnet50 to learn lower level features

5.1 Hyperparameter Tuning

Below are the hyperparameters for the different architecture tested. We only report the best one for each model. Apart from these hyperparameter sweeps, we used batch normalization, overlapping strides, cross entropy loss, batch size of 200, softmax at the last layer and parallel data loaders to speed up the process. These were common to all the models we trained and tested.

Model	Pooling	Regularization	Dropout	Activation	Optimizer	Learning Rate
3D Resnet18	Max	L2	0.1	LeakyReLU	Adam	0.0001
Unet-X	Average	L1	0.2	Relu	sgd(0.9)	0.0001
Unet-Y	Average	L2	0.2	Relu	sgd(0.9)	0.001
Unet-Z	Average	L2	0.1	tanh	Adam	0.001
Resnet50	Max	L1	0.3	sigmoid	Adam	0.004

Table 2: Best hyperparameters found for each model.

Below we show the training loss for best model on the dataset in figures 12, 13, 14.

For MD-Unet, we added an L2 regularizer and decreased the learning rate during the 100th epoch, resulting in the spike in training loss you see.

5.2 Computational hours used

All training and testing was performed on Campus Cluster and Blue Waters. Below are the exact number of hours used to train and test each model in table 3.

Model	Runs	Epochs	Total Hours
3d Resnet18	20	200	40
MD-Unet	20	800	160
Resnet50	20	60	14
Total			214

Table 3: Computational hours used for hyperparameter tuning and testing of each model. “Runs” is the number of hyperparameter combinations we tried, and “Epochs” is the number of epochs each run was trained for.

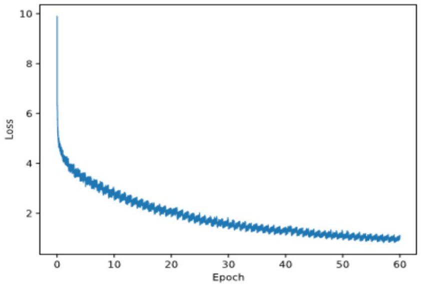

Figure 12: Training loss Resnet50

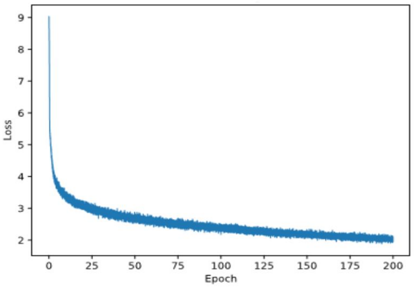

Figure 13: Training loss 3d Resnet18

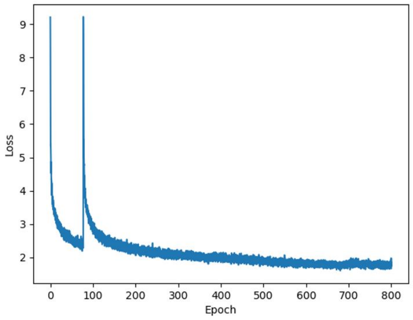

Figure 14: Combined Training loss MD-Unet

5.3 Classification Results

For evaluation of the different classifiers, we calculate the generalized performance over a 2-fold cross-validation using following metrics - accuracy, F-Score, Area Under Curve (AUC). They were computed at each cross validation step. Here accuracy is the number of samples correctly classified over all samples; F-Score is the harmonic mean between True positive rate (the rate of positive samples correctly classified) and the number of positive samples correctly classified over the number of samples classified as positive; and AUC is the area under the plot of true positive rate against the rate of negative samples correctly classified and indicates the diagnostic capacity of a binary classifier. In the end we average these values to get final results.

The table 4 below shows these results along with some state of the art results from literature.

Model	Accuracy	AUC	F-score
Shen et al [14] (SOTA)	86.80%	93%	77%
3D Resnet18 + SVM	71.34%	68.3%	68.67%
3D Resnet18 + xgBoost	70.21%	69.5%	69.67%
MD-Unet + SVM	76.3%	73.29%	70.33%
MD-Unet + xgBoost	75.1%	75.47%	69.5%
Resnet50	68.45%	65.44%	67.08%

Table 4: Classification Results

6 Conclusion and future work

In this project, we prove a methodology for lung cancer classification using different deep learning techniques. The results from different models seem promising. More training is required to reach state of the art results given the size of the dataset.

While generating the train, validation and test splits, we split the data on basis of patient id. SO while training the models we get a variance in the number and size of nodules. Studying how this division affects the performance of our models provide scope for future work.

7 Software used and dependencies.

Latest Pytorch and Keras has been used for development on python3.6 on Linux environment. Our code is accessible via gitlab [3], which has been shared with the code staff.

Acknowledgment

We would like to thank Professor Jian Peng and Nate Russell, for their constant guidance and support throughout this project.

References

[1] Centers for disease control and prevention, “lung cancer statistics. 2016.

[2] Lung image database. 2017.

[3] codebase for project on gitlab, https://gitlab.engr.illinois.edu/aa18/luna16. 2018.

[4] Hugo JWL Aerts, Emmanuel Rios Velazquez, Ralph TH Leijenaar, Chintan Parmar, Patrick Grossmann, Sara Carvalho, Johan Bussink, René Monshouwer, Benjamin Haibe-Kains, Derek Rietveld, et al. Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach. *Nature communications*, 5:4006, 2014.

[5] F Chollet. Xception: deep learning with separable convolutions. *arXiv Prepr. arXiv1610*, 2357, 2016.

[6] M Gomathi and P Thangaraj. Lung nodule detection using a neural classifier. *IACSIT International Journal of Engineering and Technology*, 2(3):291–295, 2010.

[7] Fangfang Han, Huafeng Wang, Guopeng Zhang, Hao Han, Bowen Song, Lihong Li, William Moore, Hongbing Lu, Hong Zhao, and Zhengrong Liang. Texture feature analysis for computer-aided diagnosis on pulmonary nodules. *Journal of digital imaging*, 28(1):99–115, 2015.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[9] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[10] Sarfaraz Hussein, Kunlin Cao, Qi Song, and Ulas Bagci. Risk stratification of lung nodules using 3d cnn-based multi-task learning. In *International conference on information processing in medical imaging*, pages 249–260. Springer, 2017.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[13] Wei Shen, Mu Zhou, Feng Yang, Caiyun Yang, and Jie Tian. Multi-scale convolutional neural networks for lung nodule classification. In *International Conference on Information Processing in Medical Imaging*, pages 588–599. Springer, 2015.

[14] Wei Shen, Mu Zhou, Feng Yang, Dongdong Yu, Di Dong, Caiyun Yang, Yali Zang, and Jie Tian. Multi-crop convolutional neural networks for lung nodule malignancy suspiciousness classification. *Pattern Recognition*, 61:663–673, 2017.

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.