Automated Extraction and Classification of Pulmonary Lung Nodules from CT Scans

Mike Huang

Udacity Machine Learning Engineer Capstone Project

1 Definition

1.1 Project Overview

Lung cancer is the second most common cancer in both men and women that afflicts 225,500 people a year in the United States. Nearly 1 out of 4 cancer deaths are from lung cancer, more than colon, breast, and prostate cancers combined [1]. Early detection of the cancer can allow for early treatment which significantly increases the chances of survival [2].

Lung cancer screening is performed with a CT scan that collects hundreds of images to build a full 3D composite of the lung. Next, small growths called pulmonary nodules need to be detected. These nodules show up as small, circular structures on the CT scans.

Figure 1 *Lung nodule at left arrow, cigarettes and lighter at right arrow*

In some cases, the nodules are not obvious and may take a trained eye and considerable amount of time to detect. ***Building a machine learning algorithm that can automatically detect the nodules can save considerable time and money, thus opening the accessibility of prescreening, ultimately saving lives.***

Additionally, most pulmonary nodules are not cancerous as they can also be due to non-cancerous growths, scar tissue, or infections [1]. The task is then to determine the features of a nodule that are associated with malignancy. Current state-of-the-art methods yields a 25% false positive rate in CT lung cancer screenings [3]. ***A convolutional neural network may be used to determine the features associated with cancerous or non-cancerous pulmonary nodules, and may reduce the false positive rate of CT lung cancer screenings.***

1.2 Problem Statement

1. Perform image segmentation to extract nodules from images of lung CT scans
2. Extract features from the nodules to predict whether or not a patient will be diagnosed with lung cancer within one year of the date the scan was taken.

These goals can be summarized by the following flowchart:

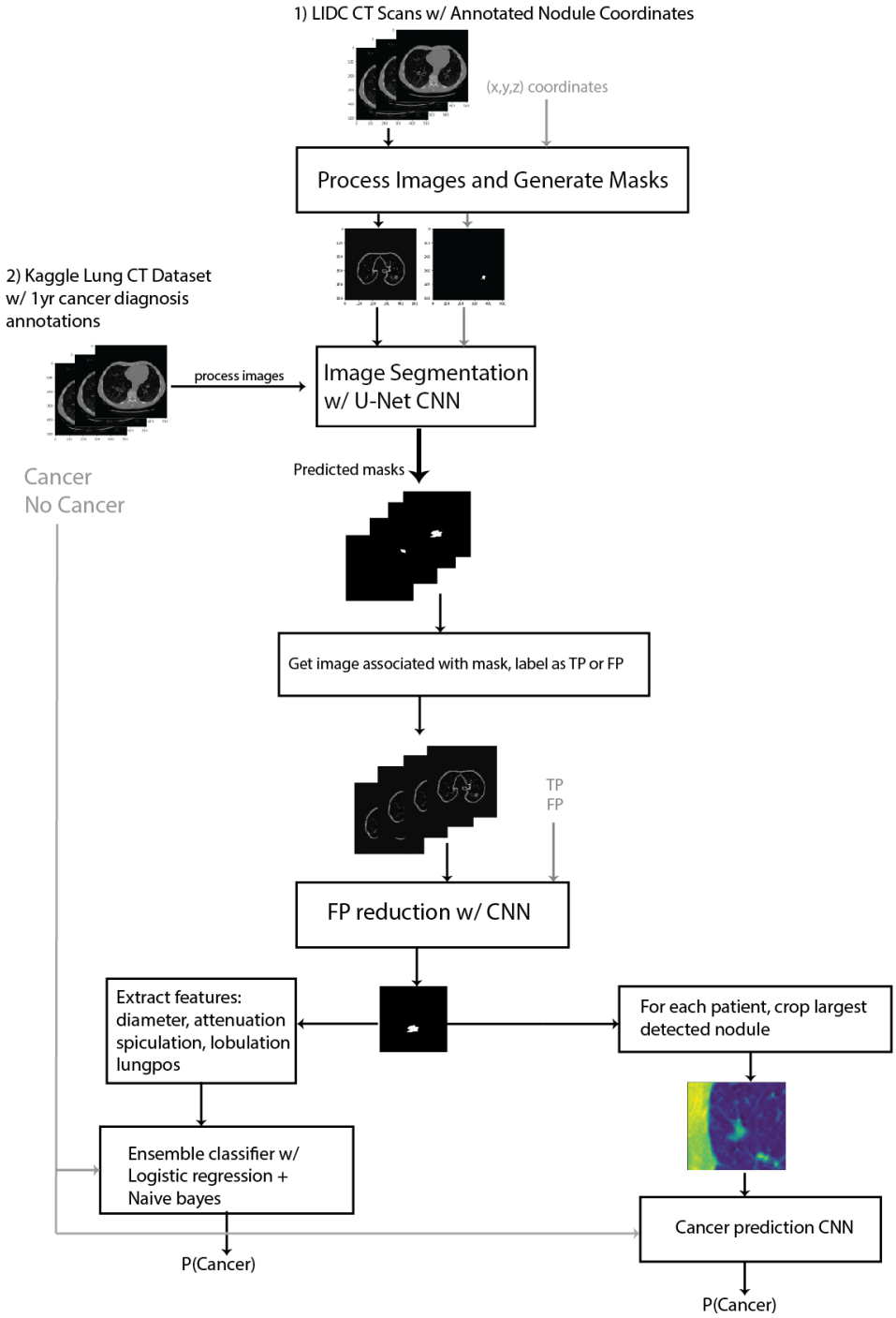


Figure 2 *Obtain a large lung CT dataset that contains annotated nodules that are verified by trained radiologists*
Preprocess dataset into inputs suitable for image segmentation. (See figure. 4)
Train a U-Net convolutional neural network for image segmentation of the nodules. The U-Net CNN will predict the nodule mask.
False positive reduction. A limitation of the U-Net network is that it can only be trained with inputs where the nodule is present. Most slices do not have a nodule. This results in many false positives. Reduce false positive nodule segmentation by u-net network by training a convolutional neural network with labels [true positive, false positive]. Ideally, this CNN will drastically reduce the false positives while only slightly reducing the true positives.
Obtain a second large lung CT dataset that contains labels whether or not it cancerous or non-cancerous
Extract nodules from second lung CT dataset and isolate the largest nodule from each patient.
Label nodules as cancerous or non-cancerous.
Extract features from extracted nodules with either a convolutional neural network or with image processing techniques to train a classifier. The CNN and classifiers will predict the probability of the patient having cancer. Compare their performances.

1.3 Metrics

There are several metrics appropriate for the various stages of this project.

1. **Image segmentation**
   Dice coefficient is suitable loss function for in image segmentation with a U-Net CNN. By minimizing the negative dice coefficient, the model attains a maximal overlap of the predicted mask with the ground truth mask.

$$\text{Dice Coefficient} = \frac{2\text{TP}}{\text{FN}+2\text{TP}+\text{FP}}$$

   TP = True Positive
   FN = False Negative
   FP = False Positive

2. **Nodule extraction**
   Sensitivity is suitable for quantifying the hit-rate of the nodule extraction, or the percentage of true nodules that are detected. Here, TP is a case where the intersection of the predicted mask and the generate mask with the nodule coordinates have a sum of greater than 1 pixels. *Average false positives per scan* is a second useful metric that should be minimized.

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

$$\text{Average \# of false positives per scan} = \frac{FP}{\#Scans}$$

3. **Cancer or no cancer prediction**
   Log Loss is a suitable loss function for binary classification. This quantifies the divergence in predicted probability to the true class labels.

$$LogLoss = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)]$$

   n is the number of patients in the test set
   $\hat{y}_i$ is the predicted probability of the image belonging to a patient with cancer
   y_i is 1 if the diagnosis is cancer, 0 otherwise
   log() is the natural (base e) logarithm

   Area Under Curve (AUC) measures how well the model can discriminate between true positives and false positives by computing the area under an ROC curve.

   Average Precision Score summarizes the precision-recall ROC as the weighted mean of precisions achieved at each threshold. This is a measure of how well the model can detect true positives.

$$AP = \sum_{n}(R_n - R_{n-1})P_n$$

   R_n and P_n is the recall and precision at each threshold for discrimination

2 Analysis

2.1 Data Exploration

Two datasets are used in this paper containing labeled nodules positions for image segmentation and cancer/non-cancer labels for classification

1. **Lung Image Database Consortium Image Collection [4]** (LIDC-IDRI) consists of lung CT scans of 1018 patients (124GB) in dicom format. Four experienced lung radiologists independently reviewed the scans and

annotated the nodules in the dataset. These annotations are contained in the file `list3_2.csv` and contains the nodule's area, diamter, and X,Y,Z coordinates for each patient sample.

Each patient sample contains approximately 100-200 CT scan slices in 512x512 images, forming a 3D composite of the lung. The values of the pixels represents radio density in **Hounsfield Units (HU)**. This ranges from -1000HU for air and up to 700-3000HU for bones. Most lung tissue is in the range of -500 to 0HU. Nodules could range from -500HU to over 200HU. This indicates the composition of the nodules and is a key diagnostic criteria for radiologists.

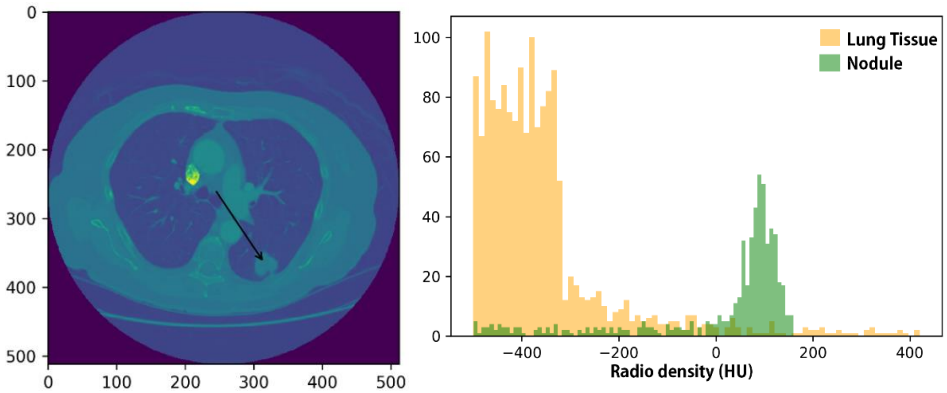


Figure 2 *A lung CT scan with an arrow pointing to a labeled nodule coordinate. Radio density of the lung tissue and lung nodule (cut off below -500HU)*

2. **Kaggle Data Science Bowl 2017 [5]** provides lung CT scans of 1595 patients (146GB) in dicom format and a set of labels denoting if the patient was diagnosed with lung cancer one year after the scans were taken. The classes are unbalanced with 3.6 non cancer samples for every one cancer sample, or the P(cancer)=0.278. The following likelihood plots were generated by extracting features from largest candidate nodule generated from the trained U-Net CNN.

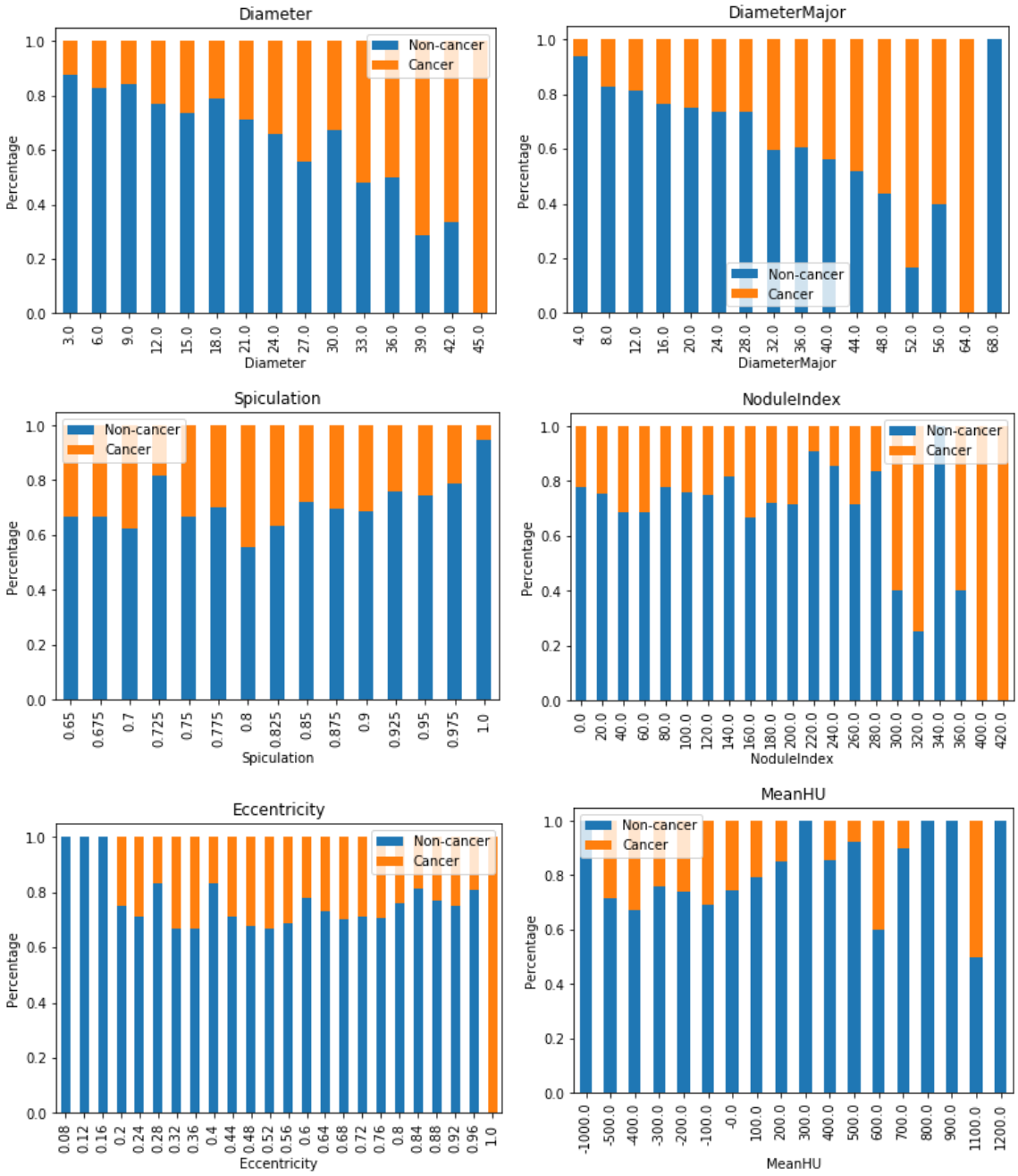


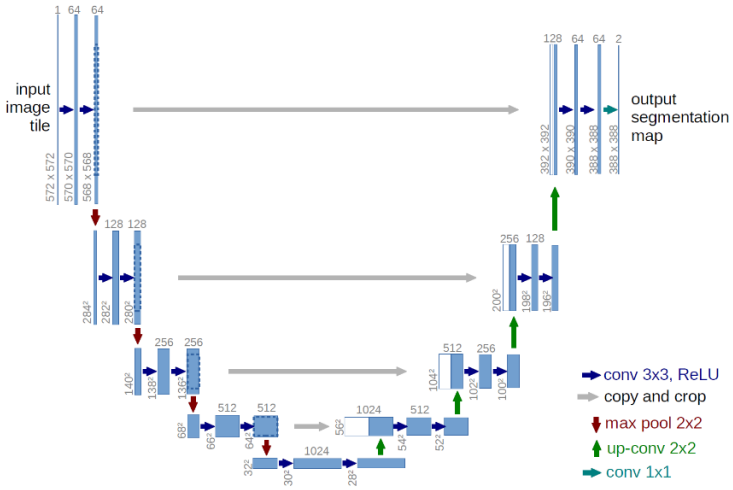
Figure 3 *Visualization of cancer likelihoods in each of the extracted features*

The candidate features were picked based on radiologists methodology for classifying nodule malignancy as outlined in this paper. This includes nodule diameter as the strongest predictor, then attenuation (MeanHU), presence/absence of calcification, and morphological features such as elongation (eccentricity), or spiculation.

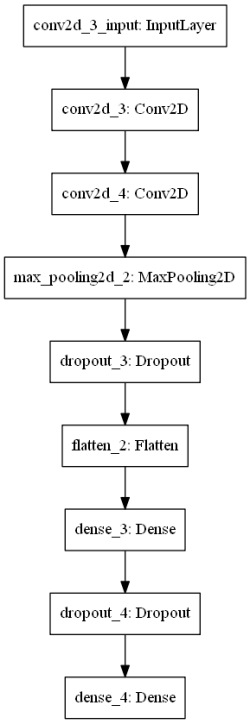
The likelihoods of each feature being cancerous is plotted to provide a visualization on the most informative features. Diameter was indeed the strongest predictor. MeanHU was also predictive as a HU range of 200-400 indicates calcification which is an indicator of benignity. Spiculation was also positively correlated with malignancy (1 indicates a completely round nodule, where a smaller value indicates more spiculation). This is corroborated with this study that found spiculation to have a 5.54 times higher likelihood of being malignant [6].

2.2 Algorithms and Techniques

The U-Net Convolutional Network [6] is the state-of-the-art method for biomedical image segmentation. It takes an input image and an output mask of the region of interest (see fig. 4). It works by first generating a vector of features typical in a convolutional neural network, and then using another up-convolutional neural network to predict the mask given the vector of features.



Convolutional Network for nodule classification



Binary labels such as cancer, non-cancer are assigned to nodules. The CNN can extract the features from the image that are most associated with each label class.

For both the U-net CNN and the CNN model, the parameters include:

1st Conv2D # of filters
2nd Conv2D # of filters
1st Dropout percentage
2nd Dropout percentage
Batch size
Epochs
ADAM optimizer learning rate
Number of layers
BatchNormalization

These parameters will need to be optimized to minimize overfitting in order to maximize the model's ability to generalize to unseen data.

Classifiers for classification of cancer/non-cancer with hand-picked features

This is a binary classification task using morphological and radiological features extracted from the images and masks. The features are continuous and numerical, but can be discretized into categories. The following classifiers were explored:

1. **Logistic regression** is particularly strong in binary classification, making it a top candidate model for this task.
2. **Gaussian Naïve Bayes** is suited for the continuous numerical features. It takes the mean and variance for each feather in each class.
3. **Multinomial Naïve Bayes** requires categorical data. This can be performed with the features transformed into discrete steps. This may be more suited than GaussianNB since some of the feature distributions representing a class is not normally distributed. For example, diameter with no cancer is strongly skewed to the left.
4. **Support Vector Machines** draws a line that maximizes the separation in the points representing the classes in a multidimensional feature space. A kernel trick can be used to fit a more defined boundary.
5. **Random Forest** frequently wins competitions on kaggle for classification tasks. It creates many decision tress with random samples and features and takes a vote on its output. This is good to prevent overfitting.
6. **Gradient Boosting** also frequently wins competitions on kaggle for classification tasks. It's similar to Random Forest but instead of random samples for each tree, it takes the samples with the highest error on the previous tree to train the successive trees.
7. **Ensembles** of models are created by averaging the output of several of the above models.

2.3 Benchmark

Since this is a two part project, there are several benchmarks to compare the performance to.

1. Nodule detection is compared to non-automated methods by radiologists. In this study, four radiologists have nodule-detection sensitivity between 51-83.2%, and a false positive rate ranged from 0.33-1.39 per case [8].
2. Predicting which nodules might be malignant by radiologists in screening studies has a PPV of 1-2% of a nodule designated as "suspicious" [8].
3. Determine how well the features used for the classification can be used to predict the malignancy classification. This can be done by comparing the validation performance with a model trained with randomly generated labels of the same proportion of classes. If the validation performance is similar by training with the real labels compared to the random labels, this indicates the model is not performing better than chance.
4. Compare the performance of a model with hand extracted features (area, radiodensity, eccentricity) with a model with automatically extracted morphological features with the CNN.

3 Methodology

File Workflow

1. ProcessNoduleDataset.ipynb
    a. *Inputs*: LIDC dataset (DOI folder), list3_2.csv, LIDC-IDRI_MetaData.csv
    b. *Outputs*: noduleimages.npy, nodulemasks.npy
2. TrainUnet.ipynb
    a. *Inputs*: noduleimages.npy, nodulemasks.npy
    b. *Outputs*: unet-weights-improvement.hdf5
3. ClassifyNodulesLIDC.ipynb
    a. *Inputs*: LIDC dataset (DOI folder), list3_2.csv, LIDC-IDRI_MetaData.csv, unet-weights-improvement.hdf5
    b. *Outputs*: truenodule-cnn-weights-improvement.hdf5
4. DetectNodules.ipynb
    a. *Inputs*: unet-weights-improvement.hdf5, truenodule-cnn-weights-improvement.hdf5, Kaggle DSB2017 dataset (stage1 folder)
    b. *Outputs*: DSBNoduleImages*.npy, DSBNoduleMasks*.npy, DSBPatientNoduleIndex*.csv
    *Split into a series of files due to large memory requirements

5. CancerPrediction.ipynb
    a. *Inputs*: DSBNoduleImages*.npy, DSBNoduleMasks*.npy, DSBPatientNoduleIndex*.csv

3.1 Data Preprocessing

Nodule segmentation preprocessing is required to prepare the input data for the U-net CNN. These inputs includes a normalized, masked image, and a nodule region of interest mask for the segmentation label.

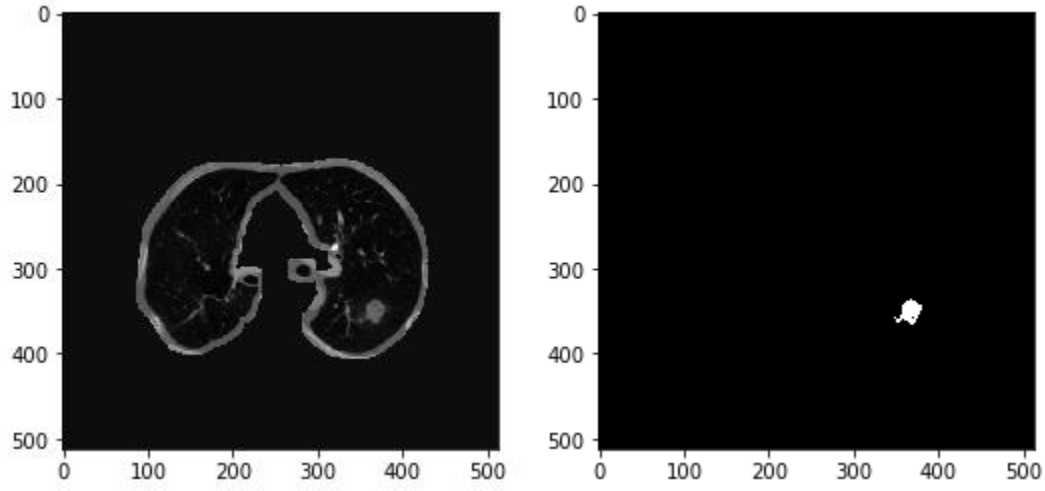


Figure 4 *Processed lung image after normalizing and masking off areas that is not the lung tissue. (Left). Nodule mask is generated to be used as a label for nodule segmentation in a U-Net CNN.*

The preprocessing is conducted in the Jupyter notebook, "ProcessNoduleDataset.ipynb"

1. Load LIDC dataset,
2. Convert dicom to ndarray with the code provided in this preprocessing tutorial. [9]
3. Normalize image by subtracting mean and dividing by the standard deviation
4. Mask lung by confining the radiodensity values in the range of the lung, expand a few pixels into the lung walls to account for nodules connected to the wall. This is done with the supplied code here [10].
5. Get coordinates of nodules from "list3_2.csv" and generate nodule mask with the cellmagicwand package [11] over the coordinate of the nodule extracted from the LIDC dataset. Threshold radio density values under -500HU to remove anything non-lung selected.
6. Remove nodules under 25mm^2 of area since these are unlikely to be malignant
7. Remove all slices that do not contain nodules

Nodule malignancy classification preprocessing prepares the nodules into 64x64 crops for training a convolution neural network

1. For all the nodules extracted from a single patient, isolate the largest nodule.
2. Get centroid coordinate of largest nodule and make a 64x64 crop of the nodule
3. Normalize the dataset by subtracting by the mean and dividing by the standard deviation

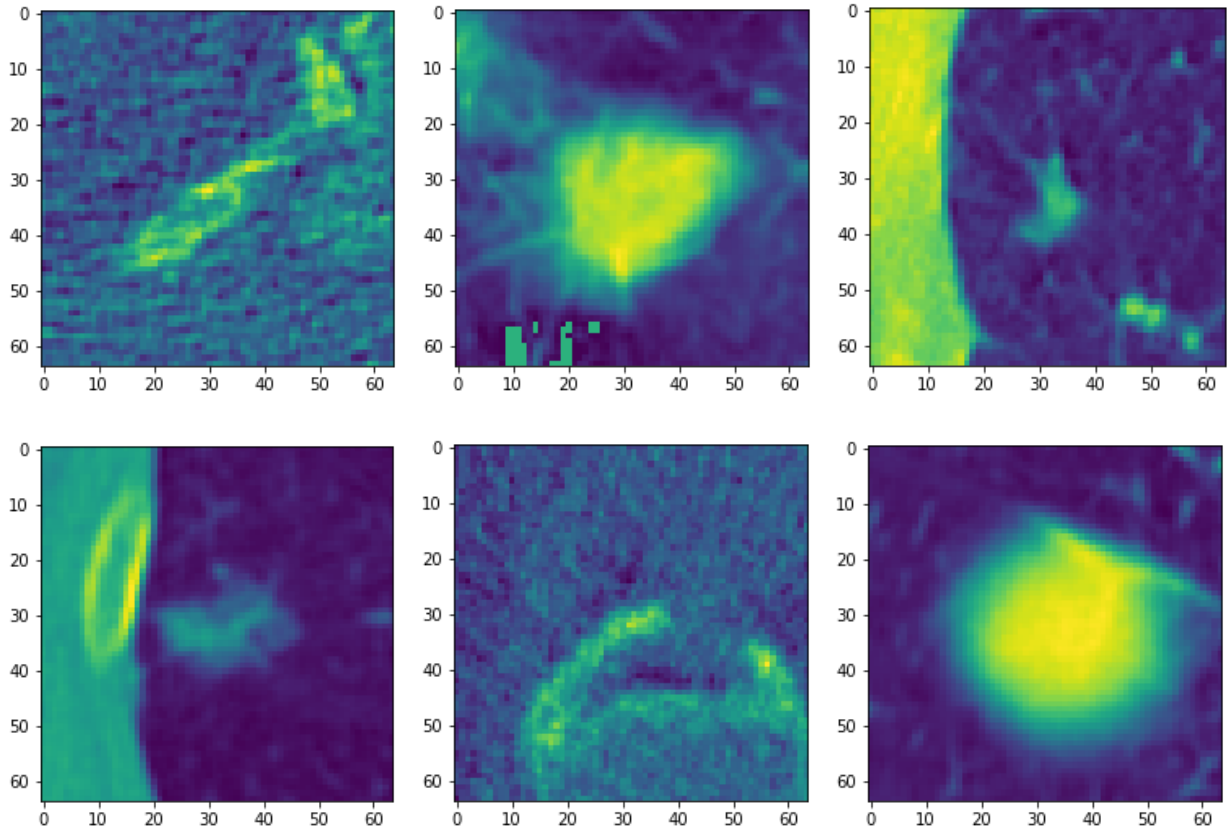


Figure 5 *Examples of 64x64 nodule crops of the largest nodules detected in each patient sample. These are inputted into a convolutional neural network and with malignancy labels supplied in the Kaggle dataset*

3.2 Implementation

Implementation of image segmentation in "2TrainUnet.ipynb"

1. Load preprocessed images and nodule masks generated by "1ProcessNoduleDataset.ipynb"
2. Split to train set and validation split (80:20)
3. Train U-Net with dice coefficient as the loss function with code sourced from [10].
4. Save model weights and plot performance. Repeat step 4 until network converges.

Implementation of true positive, false positive nodule classification in "3ClassifyNodulesLIDC.ipynb"

1. Load LIDC dataset and preprocess for input into u-net model.
2. Input preprocessed images into model to generate candidate nodule masks. This takes ~4hrs per 100 samples with a GTX1070 GPU and 3.8GHz Westmere-EP Xeon. 400 samples were processed.
3. Use "list3_2.csv" to determine which candidate nodules are true positive or false positive, label accordingly.
4. Concatenate all images that generated a false positive nodule. Label images as false positive.
5. Load images containing true positive nodules from preprocessing step and label images as true positive
6. Balance classes by undersampling false positives
7. Convert binary label to categorical with num_classes=2
8. Train_test_split dataset with test size= .3 and shuffling on.
9. Train convolutional neural network to classify true positive or false positive
10. Save model weights and plot performance.

Implementation of nodule detection in "4DetectNodules.ipynb"

1. Load Kaggle dataset and preprocess images with the identical protocol as with the LIDC dataset.
2. Load U-Net model and weights. Input preprocessed images into model to generate candidate nodule masks. This takes ~2hrs per 100 samples with a GTX 1070 GPU and 3.8GHz Westmere-EP Xeon
3. Input preprocessed images associated with nodulemasks into the CNN trained in the previous step to classify nodule as true positive or false positive
4. Save array of all true positive nodule masks and their associated unprocessed lung ct images.
5. For slices where more than one nodule is detected, use `scipy.ndimage.label` to label each nodule.
6. Generate the following features for the largest nodule of each slice: area, *mean HU value, diameter, diameter of major* axis, *speculation, eccentricity*.
7. Repeat steps 1 and 2 iteratively with 200-400 samples at a time due to memory constraints.

Implementation of nodule malignancy classification in "5CancerPredictionClassifiers.ipynb"

1. Subset table of features generated in the previous set with the largest nodule for each patient sample
2. Label as cancer or non-cancer with the labels provided in the Kaggle dataset for each patient
3. Perform exploratory analysis by plotting histograms of features associated with cancer or no cancer. Determine if any features are correlated with the labels.
4. Normalize dataset by subtracting the mean and dividing by the standard deviation for each feature
5. Create cross-validation sets by splitting dataset into 5 folds. Use 4 folds for training and 1 for test.
6. Train classifiers (logisitic regression, naïve bayes, random forest, gradient boosting, SVM). Iterate through 5 test:train splits and average logloss score.
7. Repeat steps 5 and 6 with random labels and compare scores.

Implementation of nodule malignancy classification in "6CancerPredictionCNN.ipynb"

1. Isolate the largest nodule for each patient sample and make a 64x64 crop centered over the nodule.
2. Label as cancer or non-cancer with labels provided in the Kaggle dataset for each patient
3. Create cross-validation sets by splitting dataset into 5 folds. Use 4 folds for training and 1 for test.
4. Train CNN to classify nodule as malignant or not malignant.
5. Generate random labels with the same balance of classes, train CNN with random labels.
6. Compare validation performance between CNN trained with real labels and random labels.

Discussion of complications and challenges

One challenge I ran into was the classes of data I should use to train the U-net. If I train with all of the data, most of the masks would be empty and thus the model might prefer to predict 0 for sample to minimize the loss. Training with only the samples that contain nodules would mean the model will be overly sensitive and there will be a high presence of false positives. Perhaps it might be optimal to train with balanced classes. However, I was able to reduce the false positives drastically with a second stage CNN.

3.3 Refinement

U-Net Convolutional Neural Network

1. Batch size was adjusted so the model would fit into memory. For a GTX1070, this was 6. Decreasing it to 4 slightly increased the rate of processing.
2. The network initially would not converge. Batch normalization was added after each convolutional layer which resulted in convergence at 20 epochs.
3. Model predicts a high rate of false positives. A second CNN is trained to classify false positives and true positives

Hand-picked classifiers for cancer prediction

1. Feature selection was performed by first visualizing the features to determine which ones are informative
2. Next, different combinations of features were tested on various classifiers to determine which ones performed the best

3. Initially, a number of classifiers were explored using default parameters. Out of these classifiers, logistic regression and SVM performed favorably. Further hyper-parameter tuning was conducted with a grid search.
4. The predictions of multiple classifiers were averaged together to create an ensemble model

Convolutional Neural Network for cancer or non-cancer prediction with detected nodules

The initial model sourced from an MNIST tutorial provided a near optimal performance compared to results after refining the model. Refining the model consisted of techniques to reduce model overfitting. This includes performing BatchNormalization after the convolutional layers, tuning the Dropout rate parameter, decreasing the number of convolutional filters, and adjusting the batch size.

One of the key aspects of this dataset that makes it prone to overfitting is the inherent label noise due to the assumption that the largest nodule in patient is the cancerous one if it is cancerous. While nodule size is the strongest predictor for cancer, there are still a proportion of the cases where a smaller nodule may be the cancerous one. This paper uses a large batch size to deal with label noise since the error caused by the noise is more likely to cancel each other out in each step of the gradient descent, leaving the correct labels to drive the direction of the descent [12].

4 Results

4.1 Model Evaluation and Validation

Model 1: U-Net Convolutional Neural Network for nodule segmentation

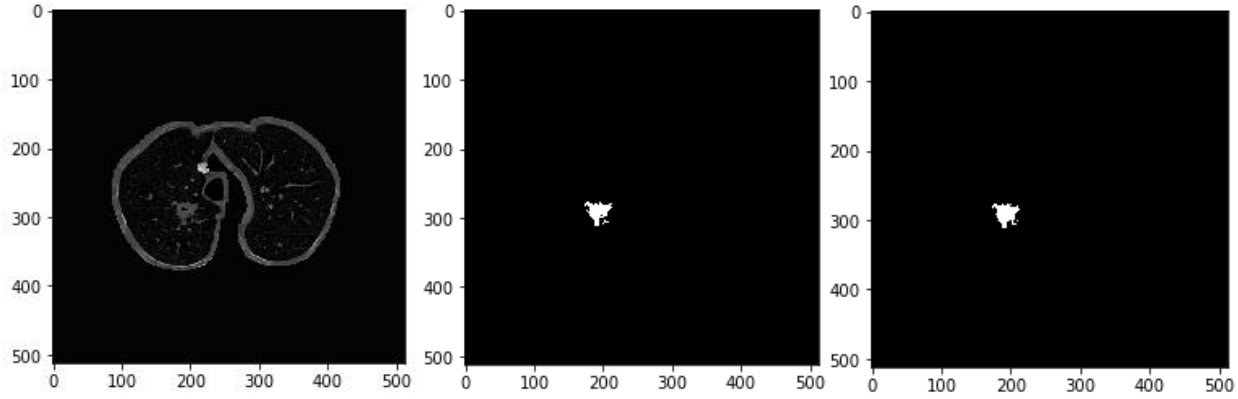


Figure 6 *U-Net image segmentation. Processed CT image (left), ground truth label (center), predicted label (right)*

The data was split into a training and validation set with an 80:20, train:test split. Due to the long training time of 9 hours for 20 epochs, a cross validation was not performed. The U-net model converged in 16 epochs to a dice coefficient of 0.678, indicating a 67.8% overlap between the predicted nodule masks and ground truth nodule masks. However, there was 75% percentage of predicted masks that have at least 1 pixels of overlap with the ground truth masks. Given that the objective is to accurately detect the position of the nodules, the sensitivity and number of false

positives per scan is the appropriate evaluation metric. There were a large number of FPs per TP, at 11.1. This is further reduced in the second model below.

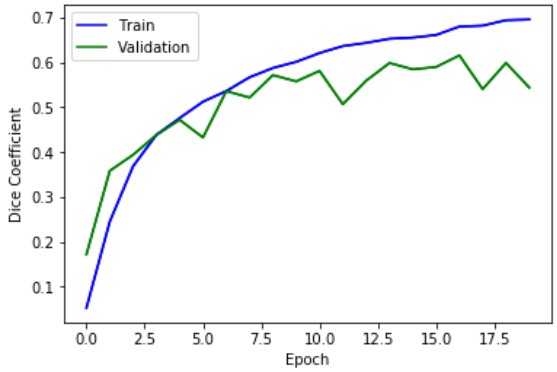


Figure 7 A dice coefficient of 0.678 was reached, indicating a 67.8% of overlap between the predicted nodule masks and ground truth nodule masks.

Model 2: Convolutional Neural Network for reducing false positives of detected nodules

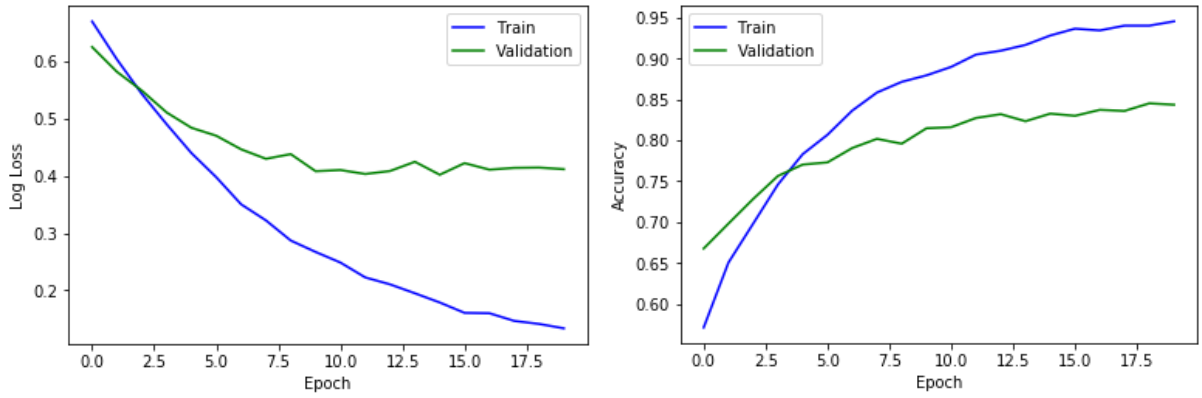


Figure 8 *CNN converges to a validation accuracy of 84.4% at classifying a detected nodule as TP or FP*

	Sensitivity	**Average # of FPs per scan**	**# of FPs per TP**
Before nodule classification	0.75	0.060	11.1
After nodule classification	0.65	0.011	2.32

Model 3: Classification of cancer or non-cancer with handpicked features

The final features selected as predictors included *Diameter, Spiculation, MeanHU, and Eccentricity*. This was determined through A/B testing to find the combination of features that performed the strongest on the best performing model.

The classification of cancer with classifier using handpicked features performed stronger than the CNN at a logloss of approximately 0.55, an AUC of .64, and an average precision of .41. In comparison, these models trained with random labels achieved a logloss of .59, AUC of .50 and an average precision of .29. The probability of cancer in the dataset is .26, so the stratified random labels performed similarly to the proportion of classes while the true labels performed substantially better. Multiple classifiers performed similarly after they were optimized with a grid search. This shows that these models performing similarly in its ability to exploit the information in the input features to make its predictions. Furthermore, transforming the training data into discretized categories by rounding resulted in less than a 0.05% increase in logloss, indicating the robustness of these models.

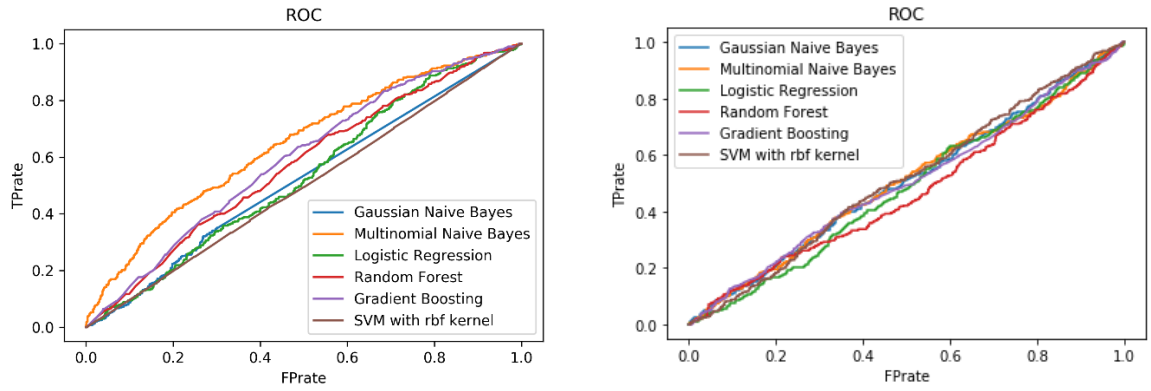


Figure 9 ROC plots True Positive Rate and False Positive Rate for true labels (left), and random labels (right).

Model	Log Loss - True Label	Log Loss - Random Label	AUC – True Label	AUC – Random Label	Average precision – TL	Average precision - RL
Gaussian Naïve Bayes	*0.5850*	*0.6037*	*.6380*	*.5053*	*.4145*	*.2929*
Multinomial Naïve Bayes	***0.5528***	***0.5920***	***.6457***	***.5050***	***.4100***	***.2893***
Logistic Regression	***0.5525***	***0.5939***	***.6448***	***.4823***	***.4158***	***.2655***
Random Forest	*0.5633*	*0.6038*	*.6150*	*.4681*	*.3769*	*.2724*
Gradient Boosting	*0.5572*	*0.5964*	*.6173*	*.5019*	*.3774*	*.2861*
SVM - rbf kernel	*0.5793*	*0.5931*	*.5017*	*.5108*	*.2714*	*.2787*
Ensemble*	***0.5519***		***.6459***		***0.4133***	

**Mean predictions of MultinomialNB and Logistic regression*

Model 4: Convolutional Neural Network for cancer or non-cancer prediction with detected nodules

The CNN model reached a validation loss of 0.5646 and an AUC of 0.6231. This is similar but marginally worse than the best performance of the classifiers with handpicked features. This may be due to diameter being the strongest predictor of cancer. CNNs are designed to be size and scale invariant, but rather focus on the features.

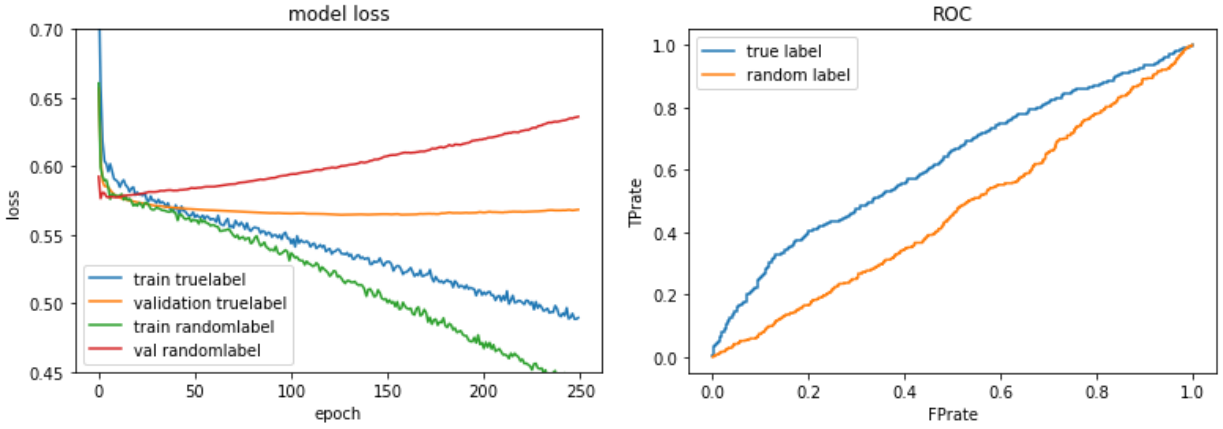


Figure 10 *Model loss for training and validation is compared between true labels and random labels (left) ROC curve is substantially improved for true labels compared to random labels (right)*

	Validation Loss	AUC
True labels	0.5646	0.6231
Random labels	0.5765	0.4651

4.2 Justification

Both the CNN and the classifiers with handpicked features had higher performance than when they were trained with random labels. For the CNN, this indicates it was able to extract informative features that predict cancer in the nodules. With the classifiers, it means the handpicked features were informative for predicting cancer. The performance of both the CNN and classifiers were similar, with the classifiers performing slightly better. This may be due to the fact that CNN is designed to be scale invariant. Given that nodule size is the strongest predictor of cancer, a classifier that is trained with size as a feature would perform better.

Compared to the performance of radiologists, the sensitivity of nodule detection was within the range of radiologists at 65% with the two stage neural networks vs 51-81.3% with radiologists. However, the false positive rate is much higher with the neural networks at 6.78 false positives per case with the neural networks vs 0.33-1.39 false positives per case with radiologists.

Despite the large number of false positives, by solely using the largest nodule detected for cancer prediction, the precision with the classifiers is substantially higher at 41% compared to 1-2% by radiologists. However, this may not be a fair comparison because the classifier performs that well with test data containing the same proportion of classes as the training data. It's unknown how the classifier would perform with test data with similar proportions of classes as in the cohort used in the study with radiologists.

5 Conclusion

5.1 Free Form Visualization

Here are two examples of edge cases that are optimal for the model to classify. The first is a small, round, and calcified (MeanHU in the 200-400 range) nodule. All four of these factors minimize its chance of being classified as malignant. The second is a large, spiculated nodule which has a high probability of being malignant in the model. This corroborates with radiologists guidelines for nodule malignancy classification.

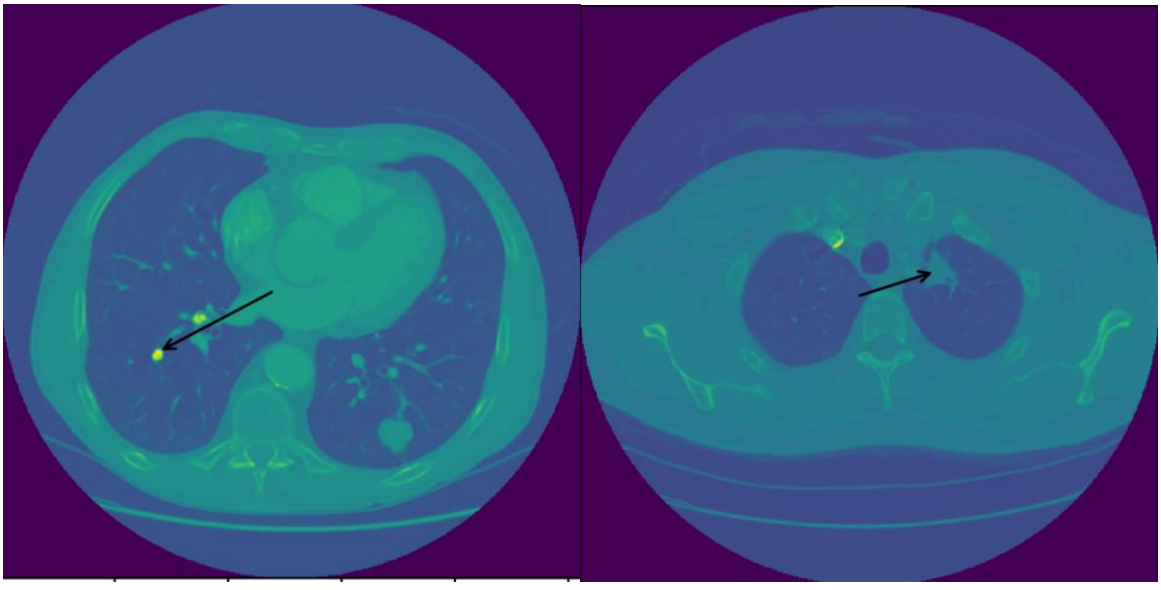


Figure 11 *A small, calcified, round nodule that's benign (left), a large, spiculated nodule that's malignant (right)*

Here are two examples of semi-solid opacity nodules that are very difficult to detect for the model, and for radiologists. They are covered in fat so the radiodensity is similar to the surrounding lung tissue. This makes it particularly difficult for the mask generation where the CellMagicWand tool relies on finding edges. However, this

is particularly a problem considering that these types of nodules are estimated to have 2/3rds to 3/4ths higher likelihoods of malignancy.

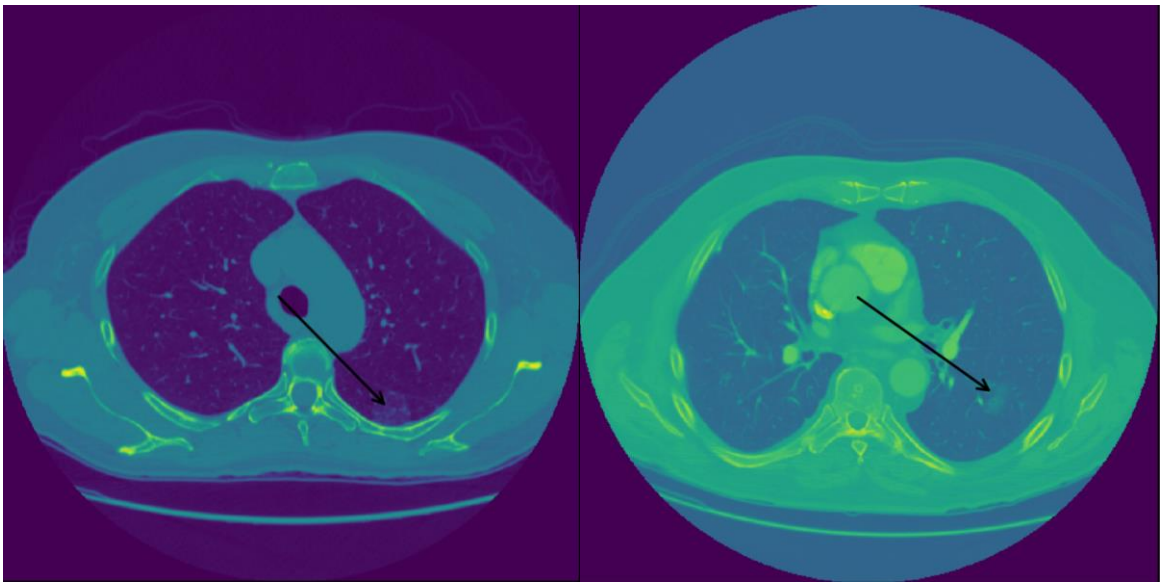


Figure 12 *Semi-solid opacity nodules are reported to have up to 2/3rds to 3/4ths higher likelihoods of malignancy but poses challenges to mask and detect. [13]*

5.2 Reflection

Overall, the end-to-end solution to estimating the likelihood of a diagnosis of cancer within a year of CT scan required multiple steps with multiple models including image processing, nodule mask generation, nodule detection, nodule false positive reduction, feature extraction, and malignancy classification. Many of these steps limited the accuracy of the prediction:

1. The nodule mask generation encountered challenges since many nodules did not have clearly defined edges. The decision to threshold the nodule at -500HU might've removed certain features from nodules.
2. U-Net was only able to detect 65% of the nodules after false positive reduction was applied.
3. The features were only extracted on the largest nodule in each patient.
4. The predicted nodule masks from the U-Net might have errors that may prevent it from generating accurate features.

Despite these issues, it was interesting to see how well the extracted features corresponded to what was reported in the literature given the limitations of only extracting from the largest nodule. The likelihoods of cancer in respect to the diameter was very close to what was predicted in the literature.

5.3 Improvement

Obtain dataset where individual nodules have been labeled by radiologists as malignant or non-malignant. Use this to train a model to more accurately classify malignancy.

While being statistically likely, the largest nodule is not necessarily the malignant nodule if a patient develops cancer. Instead of isolating the largest nodule for each patient, it might be more informative to average the largest *n* nodules.

Next, while the CNN performed worse than the classifiers for cancer prediction, it was close despite only using morphological features instead of the strongest predictor, diameter. Perhaps the feature vector can be extracted from the final layer of the CNN and used alongside diameter as the input features.

Lastly, it might be more accurate to create a Bayesian based model to account for the malignancy likelihoods from the publications that summarizes the most likely features for malignancy instead of training models from extracted features. Then use the model to associate a malignancy probability for each detected nodule by the U-Net.

6 References

1. https://www.cancer.org/cancer/lung-cancer/prevention-and-early-detection/exams-and-tests.html
2. http://www.mayoclinic.org/diseases-conditions/lung-cancer/basics/tests-diagnosis/con-20025531
3. https://biometry.nci.nih.gov/cdas/approved-projects/531/
4. https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI
5. https://www.kaggle.com/c/data-science-bowl-2017
6. http://www.clevelandclinicmeded.com/medicalpubs/diseasemanagement/hematology-oncology/pulmonary-nodules/
7. https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/
8. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2658894/
9. https://www.kaggle.com/gzuidhof/full-preprocessing-tutorial
10. https://www.kaggle.com/c/data-science-bowl-2017/details/tutorial
11. https://github.com/NoahApthorpe/CellMagicWand
12. https://openreview.net/pdf?id=B1p461b0W
13. http://cancergrace.org/lung/2007/11/10/risk-of-ca-among-spns/