A (failed) attempt at Lung Cancer Detection with Multi-Instance Networks

Jay DeStories, Jason Fan, Alex Tong

May 13, 2017

1 Part I: The (failed) Attempt

1.1 Introduction and Problem Statement

The problem of image segmentation and structure annotation within the field of biomedical imaging has become very active in the past years. In 2016 and 2015, the LUng Nodule Annotation (LUNA) challenge and SPIE Lungx challenge, asked researchers to develop models to identify pulmonary nodules in lung CT slices. With the 2016 LUNA Challenge, researchers gained access to annotated CT slices that contained segmentation ground truths for abnormal nodules. In the 2015 SPIE less than 80 CT annotated images of malignant nodules were released to the public.

Finally, with the 2017 Kaggle Data Science Bowl, a larger dataset of 1000+ lung CT images in DICOM format was finally released with cancer/no cancer labels. This has allowed researchers to answer a deeper question about lung CT scans; whether or not there are indicators of malignancy and cancer in a patient's CT scan.

There is, however, one caveat to the Kaggle dataset. Although the presence of malignancy is indicated by global, binary cancer/no-cancer label, the location of malignant nodules and structures are *not* annotated in the training data.

Inspired by recent work in multiple instance learning for whole mammogram classification [1], We investigated two methods for Lung Cancer detection using Multi-Instance Networks.

1.2 Related Work

Over the past decade there has been a significant amount of work towards computer aided diagnosis of lung cancer [2]. Depending on what kind of data researchers have had access to, previous efforts to identify lung cancer can be categorized by two main approaches.

1. Pulmonary Nodule Detection methods.

   These methods use image processing techniques to segment and annotate nodules [3, 4, 5]. These methods mimic radiologists by looking for abnormalities in the form of "solitary white nodule-like blob[s]" in a chest x-rays and CT scans. Lung nodules are potential cancer indicators, and as such are an important part in early lung cancer diagnosis. However, not all pulmonary nodules are malignant and many are benign, there will be false positives in diagnosis if we naively associate nodule presence with cancer.

2. Direct classification methods.

   These methods instead attempt to directly predict the probability of cancer using x-ray and CT images without nodule detection. [6, 7].

One challenge that almost all researchers in biomedical image inference face is small and weakly labeled datasets. Images labeled with cancer/no-cancer binaries are weakly labeled because imaged tissues only display malignancy locally; not *all* of the tissue in an image will have cancer. Zhu et al. in late 2016 trained a Multi-instance network on Mammogram images, successfully demonstrating the ability for the receptive fields of a pretrained network to be used as instances in a biomedical image application. We take inspiration from this paper and apply it to the 3d lung cancer case [1, 8].

1.3 Data

The Multi-Instance Network investigated in this project are trained only on the CT scans of 1397 patients released with the 2017 Kaggle Data Science Bowl. Each CT scan is a set of CT slices in the XY plane taken at different in the Z axis

(the Z axis runs parallel to the spine). The number of slices in the Z axis for each CT scan ranged between M and N slices.

For each CT scan, each CT slice in the XY plane is a 512 by 512 image in which every pixel measures tissue density of a point in the thoracic cavity. Here, density is measured by Hounsfield Units (HU), a linear scale that, unlike pixel values for natural images, has no upper and lower bound.

Tissue	Relevant HU
Air	-1000
Lung	≈ -500
White/Grey Matter	20-45
Muscle	35 - 55
Bone	700 - 3000

Figure 1: Respective densities of tissue

For reference respective HU values for different kinds of tissue are recorded in Figure 1.

1.4 Method

We attempted to tackle this problem using Multiple Instance Learning. Multiple instance learning is a for of semi-supervised or weakly supervised learning. Instead of labels for every instance, we have labels over bags of instances that simply state whether a given class is present. In our case this means that once we have a notion of an instance of lung cancer than this will give us a way to train on more specific parts of the network. Instead of considering the entire lung for each classification, we can focus on more specific portions of the lung and thereby train faster with less overall noise.

When a radiologist examines a lung they examine nodules (small irregular tissue pieces) for potential malignancy. Each nodule can then be classified as malignant or non-malignant. In the case of lungs most nodules are non-malignant.

We would like to build in our prior knowledge about lung cancer (that nodules are useful for detection) into our model. We provide this prior as a form of MIL. Ideally, we would like to classify a lung based on all instances of nodules in a lung. If there were many nodules that are likely malignant then the whole lung should be considered cancerous. The trick is in how to get nodule instances from the lung. We thought about attempting to train a separate segmentation or object detection network to detect the nodules within the lung, but our dataset does not lend itself to that sort of detector.

We therefore use a form of receptive field MIL applied to mammogram images used by Zhu et a. [1]. Since there are possibly many nodules in a single slice we use the receptive fields of a pretrained network as instances in our network, the idea being that a pretrained classification network such as AlexNet will have receptive fields that focus on interesting objects. Hopefully, this sort of MIL will be able to use the context around each nodule to classify malignancy. This might in an ideal lead to an out performance of human expert radiologists by considering the context around nodules instead of just the nodules themselves.

Notice that using the receptive fields is more noisy than using more specific instances as we have to learn about what makes a malignant or non malignant receptive field, and many receptive fields may be uninteresting, providing more noise to our classification.

1.4.1 Network Architectures

The output of AlexNet's conv5 layer is a 256x6x6 tensor (Channels, Width Height). We perform AlexNet on each slice leaving us with a 256x60x6x6 tensor. Each network varient then has a single learned layer on top of these features.

We tested three network architectures on the lung volumes. A fully connected layer (FC), a Receptive Field MIL network (RF), utilizing each of AlexNets 6x6 receptive field outputs as instances, and a depth MIL network (Z), which took each slice as an instance in our multi-instance learner.

The fully connected network is implemented as a 3D convolution of dimensions 60x6x6. This leaves us with a single value input to the sigmoid function. This is the simplest possible implementation built off of AlexNet.

We wanted to test using the receptive fields of AlexNet as MIL instances. in RF-MIL we therefore have 60x6x6 instances to learn on per image. We take a max over all instances to classify the whole lung volume.

We also wanted to compare the receptive field variant to a more standard MIL approach. in the Z-MIL network we used each of the 60 z layers as an MIL instance. This leaves us with a more reasonable number of instances to learn on.

FC	RF-MIL	Z-MIL
AlexNet_conv5	AlexNet_conv5	AlexNet_conv5
conv_(60, 6, 6)	conv_(1, 1, 1)	conv_(1, 6, 6)
-	pool_(60, 6, 6)	pool_(60, 1, 1)
Sigmoid	Sigmoid	Sigmoid

Figure 2: Network Architectures: conv_(D, H, W)

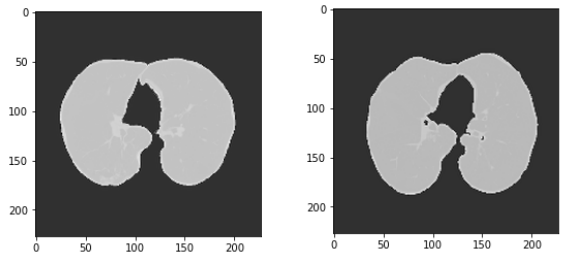

Figure 3: 30th slices of two lung volumes after Otsu's segmentation. left: cancer, right: no cancer

1.4.2 Pre-processing

The original lung images were in DICOM format, a standard format for medical images. These files have a lot of additional meta-data about the image. We wanted to shrink and normalize the lungs as much as possible so that our system could better pick up on the important features. Since we know that radiologists when analyzing lung images for cancer look for nodules, we wanted to preserve nodules and the context surrounding them in our pre-processing. The lungs are originally 512x512 images for each slice with a variable number of slices per lung. We therefore pre-processed with the following steps. (1) For each lung slice we picked the bottom right pixel as background, and found all pixels connected of the same intensity. (2) We segmented the largest non-zero connected region as the lung volume and removed air pockets within the lung as background. (3) We scaled all pixel intensities to between zero and one, clipping at -1500, and 400. We experimented with different ways in doing this (making the mean and variance of intensity values the same as imagenet for instance), but simple scaling seemed to work best. Clipping the at these values meant that we considered a wide variaty of lung densities but ignored air and bone volumes. (4) We trimmed the edges of the lung, normalizing the scale of the lung in our image. Note that this was done in a 3d sense, so not all slices fill the entire volume. (5) We scaled the image to have dimensions 60x227x227 (Depth, Height Width). We fixed the depth to fix the z axis scale as different CT scans had a different level of Z-axis granularity.

These steps left us with a dataset of lungs with a reasonable number of slices that could be input to existing networks trained on imagenet, with also a depth axis that was approximately to scale with the real world, with each pixel representing approximately a 2mm x 2mm x 2mm volume.

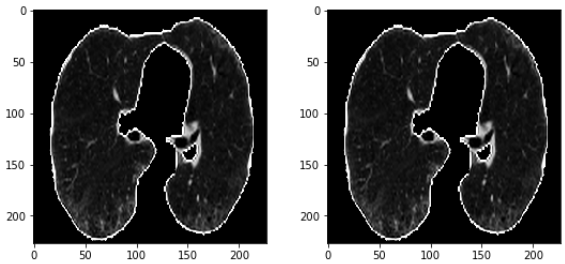

Figure 5: 30th slices of two lung volumes after Otsu's segmentation and pre-processing. left: cancer, right: no cancer

1.4.3 RGB to Grayscale

To extract features from one channel deep slices, our implementations needed to either change the channel depth of the input or alter the first convolutional layer of our pretrained networks.

When we attempted to use the same technique Zhu et. al. used to extract slice-wise features for lung volumes with pretrained convolutional neural networks (CNN), the networks could not fit all sixty DICOM slices of a lung volume into VRAM [1]. The models ran out of VRAM because, in order to use typical CNNs that expect a 3 channel deep RGB image, the implementation used by Zhu et al. effectively tripled the necessary data volume consumed by the first layer because it naively and redundantly copies each grayscale 227 by 227 slice to each of the channels; However, these redundant copies of the grayscale slices in the RGB channels can be eliminated by manipulating the first convolutional layer of our pretrained CNNs. Let us consider the first convolution applied to one RGB pixel $\boldsymbol{x}$, with weight vector $\boldsymbol{w}$, bias term b, and test time channel-wise mean $\boldsymbol{\mu}$. If all three entries

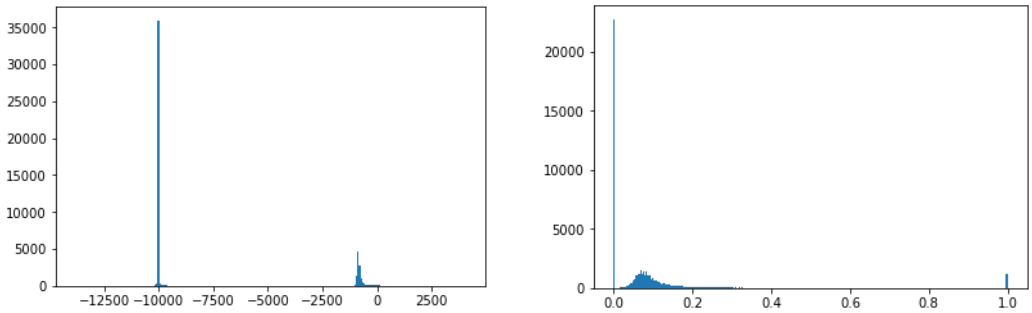

Figure 4: Histograms of pixel intensities. Left: Hounsfield Units (HU) of 30th slice of a lung volume after otsu's segmentation; parts of image on the perimeter outside of the CT scan have value -10000HU. Right: pixel intensities after normalization and clipping of HU values with bounds -1000 and 400

in $\boldsymbol{x}$ have value $\hat{x}$ then,

$$\begin{aligned} & \boldsymbol{w} * (\boldsymbol{x} - \boldsymbol{\mu}) + b \\ &= (\boldsymbol{w} * \boldsymbol{x}) + (b - \boldsymbol{w} * \boldsymbol{\mu}) \\ &= \hat{x}\|\boldsymbol{w}\|_1 + (b - \boldsymbol{w} * \boldsymbol{\mu}). \end{aligned} \tag{1}$$

This means that, with weight vector $\|\boldsymbol{w}\|_1$, bias term $(b - \boldsymbol{w} * \boldsymbol{\mu})$ and no test-time mean, a redundant 3 channel deep convolution on a grayscale image copied 3 times into the RGB space, can be converted to a 1 channel deep convolution! Using this RGB to grayscale conversion, our models reduced the VRAM usage in the first layer by a factor of 3 and could forward pass entire lung volumes to extract slice-wise features.

1.4.4 Multi-Instance Learning

In a malignant lung volume, only a small percentage of the actual tissue is malignant. As such, it may not be appropriate to learn a cancer/no-cancer binary classification for an entire lung volume. To better model the sparsity of malignant tissue in a lung volume, our models instead learn a cancer/no-cancer classification for patches of the lung volume.

1.4.5 Prediction for Multi-Instance Networks

The multiple instance models deem a lung volume cancerous if there exists a patch that is classified to be cancerous.

Let F be a set of N feature vectors that each encode a patch of the Lung CT volume. In the Z-MIL network, F is a set of 60 feature vectors where each vector is the flattened 256 by 6 by 6 output from AlexNet_conv5. In the RF-MIL network, F is a $60 \times 6 \times 6$ set of 256 channel deep vectors.

Then, $\boldsymbol{r}$ the vector of patchwise probabilities, is defined element wise by

$$r_i = \sigma(\boldsymbol{w}^T \boldsymbol{f}_i + b). \tag{2}$$

Where $\boldsymbol{f}_i$ and r_i is the i-th element of F, and $\boldsymbol{r}$ respectively.

We then use $\boldsymbol{r}$ to predict the probability of cancer $p(y = 1)$ where

$$t = p(y = 1) = \max_i \ r_i. \tag{3}$$

1.4.6 Loss functions

For the FC models, we use regular binary cross entropy loss, which is defined by,

$$L_{max} = -y \log(t) - (1 - y) \log(1 - t). \tag{4}$$

For our Multi-Instance models, Z-MIL and RF-MIL, we use a loss function as suggested by Zhu et. al [1], sparse binary cross entropy loss, that encodes the prior belief that indicators of cancer are sparse.

$$L_{sparse} = -y \log(t) - (1-y) \log(1-t) + \lambda_r \|\boldsymbol{r}\|_1 \tag{5}$$

Here the regularizing parameter $\lambda_r \|\boldsymbol{r}\|_1$ is scaled by hyperparameter λ_r.

Sparse binary cross entropy loss penalizes a model that classify many instances to be cancerous. It encourages the model to only classify few instances to be cancerous which better models our understanding of cancer.

1.4.7 Augmentation

Initial experiments showed that, because of the small data volume, every models easily overfit to either training set; each achieved almost 0 training loss within 100 epochs without significant decrease in validation loss. To prevent overfitting, at training time, each mini-batch is augmented. Each mini-batch is randomly shifted in the Z axis by 0 to 10 voxels, in the X axis by 0 to 50 pixels and also in the Y axis by 0 to 50 pixels. Each mini-batch is then also randomly flipped with probability $\frac{1}{2}$.

This stopped our models from reaching perfect training loss and overfitting.

1.4.8 Training

Initial experiments with the training data, the models were easily confused and predicted class probabilities probabilities equal to the proportion of cancerous lungs to total lungs. Because of this, some models were trained on a subset of the entire training set in which the number of cancer and no cancer samples were equal.

-	Unbalanced	Balanced	Test
# Cancer	362	322	50
# No Cancer	1035	322	121
# Total	1397	644	171

Figure 6: Class distributions for datasets

We used ADAM with default parameters and experimented with the Initial learning rate, the learning rate decay and the number of epochs. Training was done both on a GTX 1080 Ti and a K80 Amazon instance GPU. We found that the 1080 Ti was up to 3x faster than the K80 in training.

Model	Dataset	Initial LR	Decay	Epochs
FC	unbalanced	$1e^{-5}$	0.85	200
Z-MIL	balanced	$1e^{-5}$	0.95	500
Z-MIL	unbalanced	$1e^{-5}$	0.95	200
FC	balanced	$1e^{-5}$	0.85	200
RF-MIL	balanced	$4e^{-3}$	1.00	500

Figure 7: Training hyperparameters for respective models

1.5 Results

We report the best test set loss and average precision for all three networks. The RF-MIL network was not able to train on the unbalanced dataset. While the RF-MIL network had the lowest loss cross-entropy loss of all three networks on the balanced dataset, it also had the lowest average precision. We suspect that this implies not that the RF-MIL network performed the best, but that the cross entropy loss is a poor measure for this data. Overall we see that the FC network had the best performance as measured by AP.

Model	Dataset	Loss	AP
FC	unbalanced	0.621	0.407
Z-MIL	balanced	0.783	0.399
Z-MIL	unbalanced	0.684	0.346
FC	Balanced	0.865	0.302
RF-MIL	Balanced	0.720	0.293

Figure 8: Results for Lung Cancer Detection

2 Part II: Why did the networks fail?

Our networks built on top of pretrained AlexNet failed to learn anything useful from the data. We now examine the possible reasons why this was so.

2.1 A second look at the data

The Lung Image Data Consortium image collection (LIDC), is a similar dataset to the one Kaggle published, however it includes additional nodule level tags such as location, diameter, and malignancy. The top Kaggle submissions did not train on the provided Kaggle data, and trained on LIDC data instead. This is due to a “loophole" in the Kaggle challenge rules that all any publicly available data to be used as additional training data. This additional data allowed them to use nodule level tags to improve their training. This is a clear advantage over the Kaggle dataset where we are not able to eliminate other lung noise as easily.

The lungs are approximately 400x400x400mm in scale where the average diameter of a nodule in the LIDC dataset is 4.8mm. This means that the signal to noise ratio (if we only consider nodules as signal) is 1,000,000:1. With the given dataset size, even with augmentation, we do not have nearly enough data to learn this signal. LIDC data would have allowed training on a more reasonable signal to noise ratio. When we downsample to 60x227x227 this means that a nodule will

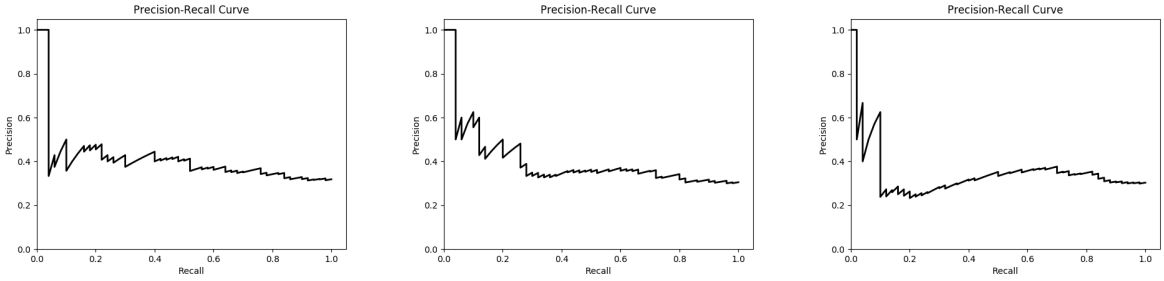

Figure 9: Precision recall curves for the following three models. From left to right: FC trained on unbalanced, Z-MIL trained on balanced, Z-MIL trained on unbalanced.

be approximately 1x3x3 pixels. This is probably too small to get any meaningful data about whether that nodule is malignant or not. We suspect training on the full data might have prevented this problem or at least minimized its effect.

2.2 HOG features + SVM baseline

We attempted to set a baseline with slicewise HOG features with PCA fed into a support vector machine. The results in Figure 10 show that we were not able to do any better than random. We suspect this has to do with the relative size of the nodules and the number of hog features that are uninformative.

We trained on the Balanced dataset which had 644 lungs in the training set (hence the maximum kernal size of 644. Augmentation was not performed during training. Most parameter settings resulted in the SVM predicting all lungs as non-cancerous. This results in a 70.7% accuracy on our test set as the proportion of non-cancerous lungs.

Kernel	Dim.	SVM Loss	Acc.	Prec.	Rec.
RBF*	32	0.60	0.707	0	0
RBF*	644	0.75	0.707	0	0
Lin.	64	0.61	0.713	0.57	0.08
Lin.	644	1.00	0.631	0.24	0.12
Sigmoid.*	644	0.778	0.707	0	0

Figure 10: Results from HOG-SVM experiments

2.3 Testing RF-MIL

We wanted to investigate our RF-MIL network on different easier to work with data. So we compared it's performance to the FC network on a constructed dataset built off of the ants and bees pytorch example.

2.3.1 Architectures

FC	RF-MIL
AlexNet Conv5	AlexNet Conv5
Conv2D (6, 6)	Conv2D (1, 1)
-	Pool (6,6)
Sigmoid	Sigmoid

Figure 12: Network Architectures for Toy Example

2.3.2 Toy Data

We generated best case, toy datasets that model low signal strength and also low noise inputs by embedding grayscale images of Ants and Bees onto a black background (shown in Figure 11). We wanted to ask the question how small a signal can the MIL network detect? Therefore we altered the sizes and random placement of the ant/bee images in a black background and measured the results.

2.3.3 Training

Model	Initial LR	Decay	Epochs
FC	$1e^{-3}$	0.95	50
MIL	$2e^{-3}$	0.85	50

Figure 13: Training hyperparameters for ants and bees models

We again used ADAM for backpropogation. We tried a couple of different hyperparemter settings for the ants and bees model shown in Figure 13.

2.3.4 Results

The FC network outperformed our RF-MIL network. We hypothesis that the RF-MIL network

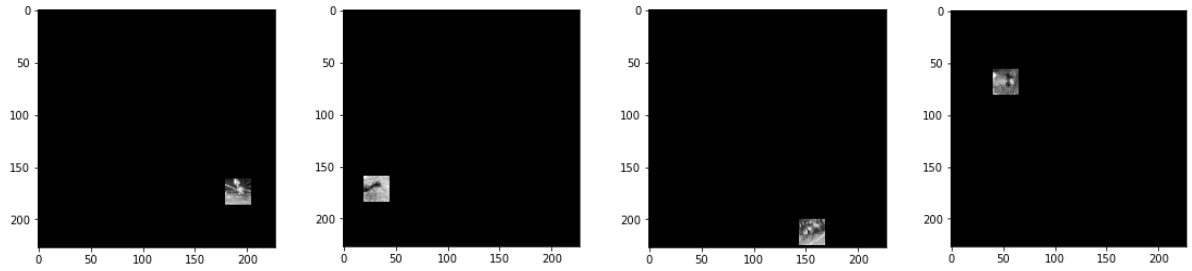

Figure 11: Example of randomly embeded 25 by 25 images of ants and bees. From left to right, the classes are: ant, ant, bee, bee.

would work best in the case where the image was full of bees and a single randomly placed ant image and asked the question is there an ant in this image? We suspect the fully connected network would classify the image as bees only as it would not have the 36 separate instance evaluations that the RF-MIL network has.

Model	Random XY	Size	Loss	AP
MIL	N/A	227	0.451	0.896
FC	N/A	227	0.585	0.898
MIL	False	113	0.610	0.753
FC	False	113	0.480	0.873
MIL	True	113	0.569	0.804
FC	True	113	0.466	0.871
MIL	False	50	0.684	0.660
FC	False	50	0.558	0.809
MIL	True	50	0.673	0.703
FC	True	50	0.573	0.782
MIL	False	25	0.687	0.631
FC	False	25	0.611	0.742
MIL	True	25	0.652	0.709
FC	True	25	0.605	0.767
MIL	False	5	0.692*	0.582
FC	False	5	0.688	0.576
MIL	True	5	0.686	0.588
FC	True	5	0.681	0.581

Figure 14: Results from Ants vs. Bees Classification Test. Size - the height and width of the embedded image in the 227 by 227 input. Random XY - whether or not the embedded image was embedded in a random location; if false, embedded image was embedded in the top left hand corner.

3 Part III: Conclusions and lessons learned

In conclusion we learned that Multiple Instance Learning is not as easy as it seems. The data bottleneck placed by performing a max over all instances limits the rate at which the network is able to learn. Since only one instance is updated by backpropogation per time step then it might potentially take *# of receptive fields* times as long to train the network. We would like to try other MIL methods for final classification (not a simple max) such as a weighted ranking of instances. This would allow the gradient to backpropogate to all instances during each training timestep potentially leading to a huge increase in training speed.

More generally, we were misled by the amount of data. Since each lung provides so much data, we thought we could learn something from the hundreds of gigabytes of data in total. We forgot to consider the importance of the annotations on that data. With the addition of nodule level annotations the effective size of the data explodes. Next time we approach a deep learning problem we will weigh not only the training set size but the annotation quality as well.

References

[1] Wentao Zhu, Qi Lou, Yeeleng Scott Vang, and Xiaohui Xie. Deep multi-instance networks with sparse label assignment for whole mammogram classification. *CoRR*, abs/1612.05968, 2016.

[2] M. G. Penedo, M. J. Carreira, A. Mosquera, and D. Cabello. Computer-aided diagnosis: a neural-network-based approach to lung nod-

ule detection. *IEEE Transactions on Medical Imaging*, 17(6):872–880, Dec 1998.

[3] X. Li, L. Shen, and S. Luo. A solitary feature-based lung nodule detection approach for chest x-ray radiographs. *IEEE Journal of Biomedical and Health Informatics*, PP(99):1–1, 2017.

[4] Rushil Anirudh, Jayaraman J. Thiagarajan, Timo Bremer, and Hyojin Kim. Lung nodule detection using 3d convolutional neural networks trained on weakly labeled data, 2016.

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

[6] Jinsa Kuruvilla and K. Gunavathi. Lung cancer classification using neural networks for {CT} images. *Computer Methods and Programs in Biomedicine*, 113(1):202 – 209, 2014.

[7] Ge Wang He Yang, Hengyong Yu. Deep learning for the classification of lung nodules.

[8] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In *Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10*, NIPS '97, pages 570–576, Cambridge, MA, USA, 1998. MIT Press.