Pulmonary Nodule Classification with Convolutional Neural Networks

Sheila Ramaswamy
Stanford University
450 Serra Mall, Stanford, CA 94305
ramasshe@stanford.edu

Karen Truong
Stanford University
450 Serra Mall, Stanford, CA 94305
truongk@stanford.edu

Abstract

With oncologists relying increasingly on low-dose CT scans to detect lung cancer, our project aims to enhance the automated detection of potentially cancerous lung nodules in these scans. While existing algorithms in the medical imaging domain focus on segmentation and diagnosis through traditional image processing techniques for identifying pathological traits, we approach the problem more generally by training and using convolutional neural networks to increase the sensitivity and accuracy of the classification of potential lung nodules. Using roughly 16,000 images of candidate nodules as training data, our best model classifies 89.6% of the actual nodules successfully and significantly reduces false positives when compared to previous attempts at the classification task.

1. Introduction

Worldwide, lung cancer is one of the leading causes of cancer-related deaths [9]. As such, it is important to be able to detect cancer in the lungs as early as possible. Small masses of tissues found in the lungs, called lung/pulmonary nodules (also known as pulmonary lesions), have the potential to become cancerous. Therefore, being able to identify nodules is indispensable to diagnosing lung cancer in its early stages. These nodules, however, are difficult to detect, as they can be as small as 1-2mm.

In the past, the only way to find such nodules was for a trained radiologist to manually analyze lung CT scans, looking for potential nodules. However, this process can be very tedious and time-consuming. In the past decade, computer-aided detection (CAD) systems have developed for both lung nodule segmentation and for the classification of lung nodules as either benign or malignant(cancerous) [3]. Our project, in a sense, focuses more on the task of lung nodule identification - or, more broadly, being able to classify specific lung scan images as either containing a nodule or not.

Current lung-nodule segmentation systems are generally very good at detecting lung nodules; however, in the process, they generate many false positives - so much so that the ratio of false positives to actual positives can be in the hundreds. It makes sense that these systems would produce many false positives: when dealing with a deadly disease such as lung cancer, it is always better to err on the side of caution, labeling anything that looks somewhat like a pulmonary lesion as a lesion. However, having so many false positives can prove counter-productive. Rather than sifting through all the false positives returned by the CAD system, it may be more efficient for a radiologist to revert to the manual methods for detecting the lung nodules.

The objective of our project is to successfully classify potential pulmonary lesions as nodule or non-nodule using neural networks, with extra effort towards minimizing the false negative rate without incurring false positives. As such, our project is a binary classification problem. The input to our algorithm is an image of a lung slice that potentially contains a lung nodule (i.e. the output of one of the existing CAD systems). We then use a CNN to predict whether the image contains a pulmonary lesion. As baselines, we also look at using SVM, kNN, and logistic regression to perform the same task.

2. Related Work

2.1. Common Radiology Practices

Radiologists have previously relied on examining images from chest radiography and PET scans to detect lung cancer [13]. However, advancements in computed tomography (CT) in the 21st century made it a more advantageous tool in both resolution and speed [14]. The manual detection of solid and subsolid pulmonary lesions in thoracic CT scans is quite error-prone, with a particularly high false-negative rate for detecting small nodules due to, for example, their size, density, location, and conspicuousness. To improve the hand annotation of nodules, medical experts expand beyond an axial scan mode and rely on other techniques such as maximum intensity projections and 3D volume renderings [11]. Maximum intensity projection (MIP) is a volume

rendering technique for 3D images that projects voxels with maximum intensity of the parallel rays from a given viewpoint onto the plane [8]. This technique makes it easier to detect denser objects like nodules, since maximum projections will be concentrated in a particular area, whereas other structures like thin blood vessels will have maximum intensities more distributed throughout the lung/image.

2.2. CAD systems

Past CAD systems for lung nodule classification generally depended on deriving a set of input features, based on the contrast, area, circularity, etc. of the nodules, and feeding them into the ANNs. [15]. In the paper Aoyama et al. 2002 [2], they classified nodules as cancerous/non-cancerous through a combination of LDA, heuristics, and a trained ANN. Suzuki et al [16] created a classifier that leveraged multiple MTANNs (massive training artificial neural network), which were trained using input CT images and teaching images. The outputs of the MTANNs were then combined to form a final score. M. Antonelli et al. [1] presented a more novel approach - they described a method for nodule segmentation and classification using a combination of image processing techniques and 3D geometric features. Although the above methods (all pre-2005) were reasonably successful, they depended on some prior knowledge and intervention. With recent increases in computing power, disk space, and data availability, CAD systems for lung nodule classification can now take a more data driven approach.

2.3. ConvNets in Medical Imaging

Recently, CNNs have become more popular in the general medical image processing community. Roth et al. 2015 [10] trained deep convolutional neural networks to be able to detect sclerotic spine metastases, lymph nodes, and colonic polyps and were able to increase sensitivity by 15-30% for each of the tasks. Furthermore, their results showed ConvNets generalize well to different medical imaging CAD applications.

Although no recent publications have dealt with CNNS and lung nodule classification specifically, there is precedent for using CNNs with lung nodule detection/classification. Specifically, in Lo, Chan et al. 1995 [5], they employ central kernels and peripheral kernels in each layer to distinguish nodules vs non-nodules in chest radiographs, with reasonable success. With today's technology, it will be interesting to see how well deeper CNNs perform with the non-nodule/nodule classification task.

3. Methods

3.1. Baseline Learning Models

As an initial baseline, we utilize simple and linear classifiers provided by the scikit-learn toolkit to be compared to the results of our ensemble of CNNs. The relevant algorithms are described in this section.

3.1.1 k-Nearest Neighbors (kNN)

In k-nearest neighbors, a query point is assigned to the class which has the most representatives within the nearest k neighbors of the point.

3.1.2 Support Vector Machines (SVM)

The SVM attempts to find the max-margin hyperplane separating the dataset based on class label, in a high-dimension feature space. The hyperplane is defined by parameters w and b. The sample x is predicted to be a positive example or negative example according to $y = sgn(K(w, x) + b)$, where K is some kernel (defined below). Finding this optimal margin reduces to solving the convex optimization problem of:

$$argmin_{w,b}\frac{1}{2}||w||^2 + C\Sigma_{i=1}^{m}\epsilon_i \quad (1)$$

where the first term is a penalty according to L2 norm, and the second is subject to constraints

$$y^{(i)}(K(w, x^{(i)}) + b) \geq 1 - \epsilon_i, \epsilon_i \geq 0 \forall i \in \{1, 2, ..m\} \quad (2)$$

where ϵ_is are slack variables to account for non-separability of the data, and kernel $K(x, y)$ is the inner product of vectors x and y. We chose to experiment with the linear kernel:

$$K(x, y)_{linear} = x^T y + c \quad (3)$$

3.1.3 Logistic Regression (LR)

This model has the hypothesis parameterized by θ:

$$h_\theta(x) = sigmoid(\theta^T x) = \frac{1}{1 + exp(-\theta^T x)} \quad (4)$$

The value of $h_\theta(x)$ is interpreted as probability of class 1 for example x. Inference is by $y = 1_{\{h_\theta(x) \geq 0.5\}}$. For training, we try logistic regression with L2 penalty ($C = 1$ being the L2 regularization hyperparameter):

$$\theta = argmin_\theta \frac{1}{2}\Sigma_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + C||\theta||_2^2 \quad (5)$$

3.2. Loss Function

In our convolutional models, we chose to use a Softmax classifier, which produces normalized probabilities (between 0 and 1) for the two classes for a given sample, which we can interpret as a confidence score. The loss function is:

$$Li = f_{y_i} + log\Sigma_j exp(f_j) \quad (6)$$

The total loss is computed as the average of all losses over the samples, in addition to a regularization term to make sure weights are well distributed. The classifier's goal is to minimize the difference between the true label probabilities and the class prediction probabilities, which are calculated by the softmax function $\frac{e^{f_{y_i}}}{\Sigma_j e^{f_j}}$

3.3. Nonlinearities

In our CNNs, we chose to use ReLU nonlinearities. The ReLU (Rectified Linear Unit) essentially thresholds an activation to 0. Although ReLU nonlinearities can sometimes lead to "dead" neurons that never activate, in practice, they have been found to lead to faster convergence. The ReLU formula is as below:

$$F(x) = \max(x, 0) \tag{7}$$

3.4. CNN Architectures

For the binary classification task, we looked at 2 different CNN architectures that proved successful on the ImageNet challenge. In addition, we looked at creating ensembles using the two architectures.

3.4.1 AlexNet

We looked at a modified version of AlexNet, a CNN architecture that uses ReLU nonlinearities. The architecture is as follows: $\{conv-relu-pool-norm\}*2-\{conv-relu\}*3-\{pool\}-\{fc-relu-dropout\}*2-\{fc\}$, followed a softmax loss function. The original model takes inputs of size $(227, 227, 3)$ and outputs scores for 1000 different classes (for the ImageNet challenge). We instead modified it so that the final fully connected layer instead outputs only 2 scores - one for nodules and one for non-nodules.

3.4.2 GoogLeNet

GoogLeNet is a 22-layer CNN, containing "Inception Modules". It uses convolutions, max-pooling layers, ReLU nonlinearities, and the softmax loss function. Each "inception module" consists of multiple convolutions (with different filter sizes) and max-pools that are concatenated together. The original GoogleNet takes inputs of size $(224, 224, 3)$ and outputs scores for the 1000 ImageNet classes. We modified the final FC layers to instead output scores for only 2 classes, as we did with AlexNet.

3.5. Ensembles

Ensembles utilize multiple, generally independently trained, models for the task of prediction. We propose to form an ensemble, consisting of our best two AlexNets and best two GoogleNets. Specifically, we look at two different ensemble strategies:

1. Take the majority class. To break ties, we always choose the positive (nodule) label.

2. If any of the models in the ensemble predict a positive label, we assign the image the positive label.

In both the ensembles schemes, we favor the nodule class so as to minimize false negatives, which is in line with what radiologists prefer if they are to detect these potentially dangerous lesions early on.

3.6. Gradient Descent Optimization Methods

We experimented with different gradient update rules used for the backpropagation through layers of our deep network. The mathematics of a few of those methods are described in this section.

3.6.1 Batch Gradient Descent with Momentum

The default update rule in Caffe, the main toolkit used for our expirements, is to do vanilla batch gradient descent with momentum. While gradient descent updates the weights based on the negative gradient $-\nabla L(W_t)$ of the weights, momentum μ encourages the updates to also change in the direction of the previous update as follows:

$$V_{t+1} = \mu V_t - \alpha \nabla L(W_t) \tag{8}$$

$$W_{t+1} = W_t + V_{t+1} \tag{9}$$

3.6.2 Nesterov Momentum

This strategy works with an accelerated gradient by computing the gradient on $W_t + \mu V_t$ instead of just W_t.

$$V_{t+1} = \mu V_t - \alpha \nabla L(W_t + \mu V_t) \tag{10}$$

$$W_{t+1} = W_t + V_{t+1} \tag{11}$$

3.6.3 Adam

An adaptive per parameter update rule, Adam makes use of momentum as well as maintaining some knowledge of past updates (not just the previous one) in a decaying cache, such that the gradient is updated by its moment and velocity:

$$(m_t)_i = \beta_1 (m_{t-1})_i + (1 - \beta_1)(\nabla L(W_t))_i \tag{12}$$

$$(v_t)_i = \beta_2 (v_{t-1})_i + (1 - \beta_2)(\nabla L(W_t))_i^2 \tag{13}$$

$$(W_{t+1})_i = (W_t)_i - \alpha \frac{\sqrt{1 - (\beta_2)_i^t}}{1 - (\beta_1)_i^t} \frac{(m_t)_i}{\sqrt{(v_t)_i} + \varepsilon} \tag{14}$$

4. Dataset

The Lung Image Database Consortium and Infectious Disease Research Institute (LIDC/IDRI) contains an image collection of diagnostic and lung cancer screening thoracic CT scans[1]. Our project consists of 888 CT scans with marked-up lesions that we use as ground-truth labels for the classification problem. The computed locations in the mark-up file are generated by three existing CAD algorithms [7] [4][12], which are not perfect. Consequently, the annotations file additionally contains false positives that we can incorporate into our training set. The true labeling of the computer-generated candidate nodules was performed by four professional radiologists and reveals that we have a very unbalanced dataset of mostly false positive mark-ups (i.e. non-nodules). The annotations were provided to us by the Lung Nodule Analysis (LUNA) challenge as part of the 2016 IEEE International Symposium on Biomedical Imaging.

4.1. Image Extraction

Each of the 888 CT scans consists of a MetaHeader (.mhd) file and the unprocessed multidimensional scan in a .raw format. A significant amount of time had to be invested in generating the 2-dimensional images of potential nodules by translating the coordinates in the mark-up file to select the correct cross-sectional slices of the lung scan, crop them, and store them in a traditional image file format.

Using the Simple Insight Segmentation and Registration Toolkit (SimpleITK)[6], we read the raw scan in and converted it into a 3-dimensional array. The annotations file provides the candidate locations in world coordinates, which must be converted to non-integer voxel coordinates to correctly identify the region in the array containing the potential lesion. The image is then normalized for grayscale coloring and then cropped around the candidate nodule to generate a 227x227 PNG. Since the provided images are in black and white, we also converted the images to RGB format to make the images meet the input channel dimensions expected for the pre-trained ImageNet models.

Because each 3-D scan has hundreds of false positive annotations associated with it, we chose to extract all the nodule examples but undersample the non-nodule crops. However, we still maintained a class skew of roughly 85% non-nodule and 15% nodule images. Additionally, we performed mirroring and rotation to augment our training set. Our final dataset is composed of 15,662 images of potential nodules, with 2,428 true positives. The organization running the LUNA challenge split the dataset into roughly 80% training (13,888 crops), 10% validation (853 crops), and test set (915 crops). Figures 1 and 2 contain example images from each class.

[1] https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI

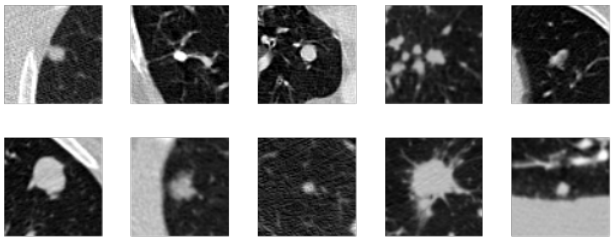

Figure 1. Nodules.

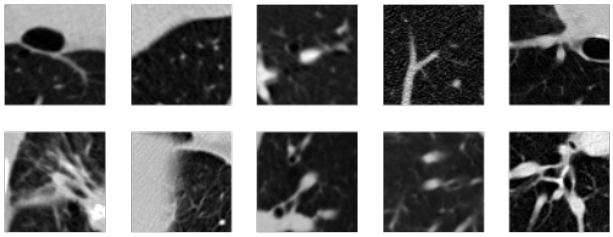

Figure 2. Non-nodules.

5. Experiments

For each of our experiments, the input was the candidate nodule image, and the output was the predicted class label. Note that, as a pre-processing step for all the experiments, we subtracted the mean image (generated from the training set) from each of the images. As described above, rather than using cross-validation, we used a single validation set during training, for each of our models.

5.1. Baseline Models

We used the sci-kit learn library to run each of the baseline models. To adjust for class skew in the kNN model, we kept k small (k=10) and used weights such that the contribution of each neighbor is inversely proportional to its distance from the point. The SVM model used a penalty error of C=0.001, using the notation convention in equation 1. (Note that this is inversely proportional to the regularization constant.) Balanced class weights, which adjusts weights inversely proportional to the class's frequency, and L2 penalty were specified in the parameters for both SVM and the Softmax/LR models.

5.2. ConvNets with Images

The main focus of our project was to use ConvNets on the images. We used the Caffe library to train and run all of our models. Note that for each of the ConvNets, we decided on the final model (i.e. the one we would run the test set on) based on both the overall validation accuracy and the validation sensitivity, so as to minimize FNs.

5.2.1 Transfer Learning

We used pre-trained AlexNet and GoogLeNet models (on ImageNet, in the Caffe Model Zoo) as fixed feature extractors for our classification task. Since lung scans are quite different from the types of images found in ImageNet, we fixed a limited number of layers from the pre-trained network and trained the higher layers from scratch. The idea is that, since earlier convolutional layers tend to pick up more general features, such as edges, they generalize more. As part of our experimentation, we tried to find the optimal number of pre-trained layers to fix in o ur final models. For the "fixed" layers, we still allowed for some learning: rather than set the the learning rate to 0, we instead set it to $1/100$ of the learning rate used for the rest of the network. In addition, for the experiment with i fixed layers, we used the model weights from the previous experiment (with $i + 1$ fixed layers) as a starting point, so as to speed up the learning process.

5.2.2 Hyperparameters

In terms of batch size, we used a relatively small size, due to limited memory resources. For all the AlexNet models, we used a batch size of 128. For GoogLeNet, on the other hand, since it is a deeper network, we used a smaller batch size of 32. Dropout was tried in ranges from 0.25 to 0.75, and learning rate was tuned using a coarse-to-fine sweep from 0.01 to 0.00001.

5.3. Unconventional 3D Network

Another variation of our experiments involved manipulating the data by combining multiple image slices around a candidate nodule, creating an image volume to feed into the AlexNet CNN, effectively giving the input image more depth. Additional images were extracted directly above and below the coordinates given in the annotations to form a 227x227x3 volume as input into our architectures, which do not use any pre-trained weights. The motivation behind this idea is that in practice, radiology oncologists look at volume renderings of the lungs or maximum intensity projections, techniques that have been demonstrated to produce more accurate and precise annotations, as mentioned earlier in section 2.1. We wanted to design an experiment that simulates this type of outlook on the data, or at least provide our model with similar information about the lungs. Furthermore, given that our dataset was in grayscale, which can be represented in one channel, we naturally had additional channels to feed in more information about each nodule beyond a single image crop.

5.4. Evaluation Metrics

Our primary metric is the sensitivity, or true positive rate (TPR), of the models, since in the medical realm a method that fails to identify all health-threatening nodules (the positive class) puts the patient at risk. Another major area of improvement that we hope to achieve is also in reducing false positives, so evaluating the specificity and positive predictive value (PPV) of our model is also important. For reference, the relevant formulas for each metric are displayed below.

$$Sensitivity = \frac{TP}{TP+FN} \quad Specificity = \frac{TN}{TN+FP}$$

$$PPV = \frac{TP}{TP+FP} \qquad NPV = \frac{TN}{TN+FN}$$

Applying the technical approaches described in the Methods section, we expect our neural networks to outperform our baseline classifiers since the neural network should have access to more salient features than our baselines, which rely strictly on individual pixel values and locations. For the purposes of fine-tuning our model, it is easiest to make adjustments as we monitor the loss and validation accuracy over epochs, which are accessible by Caffe logs.

6. Results

The results of our baseline and ConvNet models are summarized below. Note that, since the LUNA challenge is currently ongoing, we have no results to compare our model against. Therefore, we have several baselines for comparison.

6.1. Baseline Results

We ran some preliminary linear classifiers and kNN as baselines. Because the purpose of this project was to focus on convolutional networks, only the test results are reported below in Tables 1, 2, and 3.

		Predicted	
		p	n
Actual	p′	40	85
	n′	19	773

Table 1. kNN confusion.

		Predicted	
		p	n
Actual	p′	53	72
	n′	158	634

Table 2. SVM confusion.

With respect to the positive class, the kNN (k=10) has poor recall, though it does not generate as many false positives as the other two linear models. Because we really do not want false negatives, this model is unacceptable for the classification challenge at hand. The linear SVM is able to correctly label an additional 10% of nodules in the test set, but at the expense of many false positives. Of the three

		Predicted p	Predicted n
Actual	p′	66	59
	n′	264	528

Table 3. LR confusion matrix.

baseline models, the logistic regression/softmax classifier recalled the most nodules (52.8%), but the precision for class 1 (PPV) was only 0.20, indicating that this classifier gave many false positive labels as well.

6.2. ConvNet Results

The results for each of the ConvNets are summarized in Table 4. We also provide sample false negatives in Figure 3 and sample false positives in Figure 4. Since the ensembles combine the results of AlexNet and GoogLeNet, we only show images for AlexNet and GoogLeNet. Results for each of the individual models can be seen below.

	Nodule		Non-nodule	
CNN Model	Sens.	PPV	Spec.	NPV
AlexNet	80.8%	82.1%	97.0%	97.0%
GoogLeNet	79.2%	86.8%	98.1%	97.0%
Ensemble1	84.0%	82.7%	97.2%	97.5%
Ensemble2	88.0%	70.5%	94.2%	98.0%
3D AlexNet1	85.6%	90.7%	98.6%	97.7%
3D AlexNet2	89.6%	86.2%	97.7%	98.3%

Table 4. CNN results on test set.

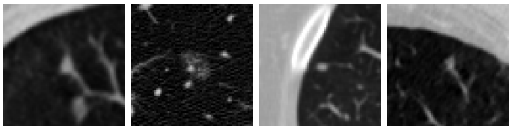

Figure 3. From left to right: false negative for AlexNet only, GoogLeNet only, 3D AlexNet2 only, and all models.

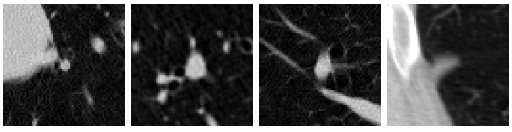

Figure 4. From left to right: false positive for AlexNet only, GoogLeNet only, 3D AlexNet2 only, and all models.

6.2.1 AlexNet

In the final AlexNet model, we ended up fixing the first 3 convolutional layers from the pre-trained model and training from scratch the remaining 2 convolutional layers and all of the fully connected layers. Figure 5 shows loss overtime, with each series representing a model with a different number of fixed layers. Note that training the 4th convolutional layer from scratch did not make a huge impact on the loss function.

In terms of hyperparameters, we started off with a learning rate of 0.01 (and a learning rate of 0.0001 for the fixed layer); in addition, for the final training phase, we lowered the base learning rate to 0.001, as the loss had started to plateau. For regularization, we used a $\lambda = 0.0005$.

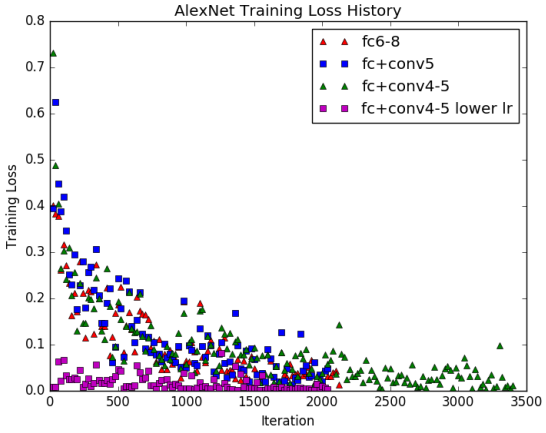

Figure 5. Loss overtime with AlexNet

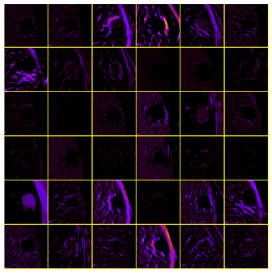

Figure 6. Top 36 activations at Conv1 layer on a test image.

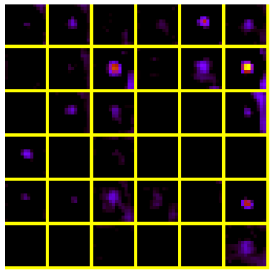

Figure 7. Top 36 activations at Conv5 layer on a test image.

6.2.2 GoogLeNet

As with AlexNet, we performed transfer learning with the GoogleNet. We found that, like with AlexNet, the first few layers worked well as fixed feature extractors. In the final model, we ended up training from scratch four of the inception modules. In terms of hyperparameters, for our best

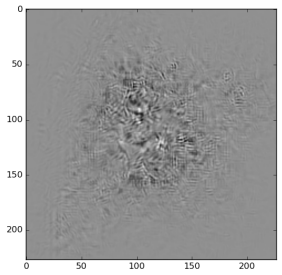

Figure 8. Saliency map

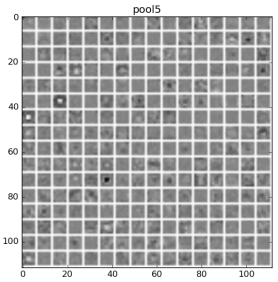

Figure 9. Gradients into pool5

GoogLeNet, like with AlexNet, we originally used a learning rate of 0.01, which was lowered overtime to 0.001. The "fixed" layers were given a learning rate of 0.0001. In terms of regularization, we used $\lambda = 0.0002$.

As seen in Table 4, notice that the sensitivity is lower than than the best AlexNet model, but the specificity is higher. In Tables 5 and 6, we give the confusion matrix for the validation and the test sets. Note that Table 5 represents the model that achieved the lowest number of false negatives, using the GoogLeNet architecture.

		Predicted	
		p	n
Actual	p′	115	34
	n′	13	691

Table 5. GoogLeNet validation confusion matrix.

		Predicted	
		p	n
Actual	p′	99	26
	n′	15	777

Table 6. GoogLeNet test confusion matrix.

6.3. Ensemble Methods

For both the ensembles, we used the two GoogLeNet and two AlexNets that achieved the highest validation accuracy. The results for the ensemble methods can also be found in 4. Notice that in both ensembles, particularly with Ensemble2 (choose positive if any model chooses positive), we see an increase in sensitivity and a substantial decrease in the PPV.

Both the AlexNet models we used came from different snapshots of the weights from the same training session (with all but 3 conv layers trained from scratch). The same was the case with the GoogLeNets.

6.3.1 Unconventional 3D Network

With the unconventional 3D Network, we used the general AlexNet architecture, but trained all the layers from scratch. We first overfitted a model on a small subset of the training data. We then used the resulting weights as a starting point to train the model of the full train set.

In figure 10, we see the validation accuracy over roughly the first 3000 iterations. Note that, with the pretrained weights, we immediately get an overall accuracy of around 94%.

The final two models we chose were the ones with the highest overall validation accuracy (3D AlexNet1) and the one with the highest validation sensitivity (3D AlexNet2). In the case of the latter, we see that even on the test set, it has high sensitivity.

For hyperparameters, we found that using the same learning rate as in the AlexNet with the regular images as input caused the losses to explode. As a result, we used a starting learning rate of 0.005 that decayed overtime to 0.001. In terms of regularization, we found that using 0.001 worked well.

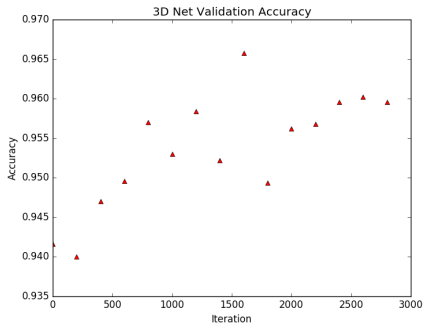

Figure 10. Validation accuracy overtime with Unconventional 3D Network

7. Discussion

Comparing the models, it is clear that our lack of extensive feature extraction for the baseline models makes it a poor classifier. Note that, with a random model (flipping a weighted coin, where the non-nodule class has probability 85%), we can achieve a sensitivity and PPV of 15% and a specificity and NPV of 85%.

In terms of the ConvNets, we see that, when comparing AlexNet to GoogLeNet, AlexNet seems somewhat better at at minimizing false negatives whereas GoogLeNet seems better at minimizing false positives. As a result, as seen in Table 4, we find that the sensitivity is higher for AlexNet and specificity (and the PPV, for that matter) is higher for GoogLeNet. The ensembles, as expected, achieve a higher sensitivity than than the individual models. This is expected as, particularly with Ensemble 2, there is a bias towards the nodule class. However, because of this bias and the class skew, we also see an increase in false positives, resulting in a significant drop in the PPV.

Qualitatively, the saliency map from our best AlexNet in Figure 8 demonstrates a focus on the center of the crops where the nodule is located and takes on a large blob shape.

This observation is further supported by the results of the pooling layer in Figure 9, which shows that the gradients are being updated mostly in the same region. The initial layers of AlexNet, which were fixed, capture very basic features about images that were still relevant to us, such as the edges of objects in the image which "light up", as in Figure 6. Our own convolutional layers trained from scratch, such as conv5 in Figure7, activate based on a structure more sophisticated and relevant to the shape of lesions.

Looking at AlexNet, with regards to transfer learning, we see there is not a significant difference between loss when fixing conv layers 1 to 4 and 1 to 3, as seen in Figure 5. This indicates that the pre-trained AlexNet weights worked well as fixed feature extractors - they were still able to pick out relevant features for identifying nodules.

For GoogLeNet, we see that in Table 5 and 6, there are actually more false negatives than false positives, despite there being more negative (non-nodules) than positive (nodule) examples in the dataset. A similar pattern was found in all of the other models - hence, sensitivity tends to be significantly less than specificity. Given the large class skew, it makes sense that there would be a bias towards the non-nodule class. All of the models try and minimize softmax loss, which weights each class evenly; thus, it is easier to minimize loss by correctly labeling all non-nodules image (note that this would lead to a, minimum, 85% accuracy) rather than focusing on labeling all nodule images correctly.

Finally, observe that, overall, the 3D AlexNet models tend to have consistently high values for sensitivity, PPV, specificity, and NPV. This makes sense, as they are able to incorporate surrounding features and, somewhat, the 3D structure of the nodule into their prediction.

7.1. Overfitting

We naturally expect some overfitting due to the class imbalance of our dataset. To mitigate class skew, we intentionally undersampled negative examples in our dataset. With respect to overfitting in the learning process, we tried increasing dropout, using higher regularization, and setting a smaller batch size, which allows for gradient updates to be a bit noisy and not too stringent to our training data.

7.2. False Negatives/Positives

As seen in Figure 3, we find that all the models tend to mislabel nodules when they do not have the more obvious features of a nodule - the round/spherical shape; note that, at first glance, most of the images look more like blood vessels rather than nodules. However, the GoogLeNet missed an image with the characteristic round, white spots, demonstrative of the tradeoff between false negatives and false positives. In the case of false positives, as seen in Figure 4, all the models tend to misclassify images with the white spheres, characteristic of nodules.

On the whole, our model seems to perform well when the nodules are round, bright white appearance. Since this is the characteristic appearance, nodules that skew significantly for this model - or non-nodules that fall too close to this model - inevitably cause errors. It is important to note that, since the ground truth labels were given by radiologist, it is possible that some labels are incorrect due to human error. So, it is still worthwhile to investigate the examples our models erroneously classify as nodule/non-nodule.

7.3. Recovering Previous False Negatives

We had an additional test set of size 60 consisting of nodules that the original CAD algorithms missed. Our best model, the 2nd ensemble, was able to correctly classify 37 of these nodules. Given its bias towards the nodule class, it makes sense that it would give the best results. With the exception of the 3D AlexNets, the remaining models were able to detect between 30 and 31 of the nodules. With the 3D AlexNets, however, they only correctly classified 25 and 26 nodules, respectively. This could indicate that the 3D AlexNets are closer in line to the original CAD systems in how they identify nodules. Regardless, it is promising that our models are able to detect nodules that the original CAD systems missed.

8. Conclusion and Future Work

Given the deadliness of lung cancer, it is important to be able to identify lung nodules - masses of tissue that can become cancerous. While CAD systems exist for this task, they often produce too many false positives to be of use to oncologists. However, we found that CNNs can successfully be used to classify candidate nodules, with relatively high sensitivity and very high specificity. In particular, we found that the unconventional 3D AlexNets were particularly good at identifying nodules, as compared to the other models. We believe this can be attributed to the fact that a higher dimensional outlook on the CT scan provides more salient features about the structure of the candidate nodule being classified. Additionally, we found that ensembles provide a good means for achieving a high sensitivity, but came at the cost of more false positives, unlike with the 3D AlexNets.

This shows promise for incorporating 3D features and structures into the CNNs for nodule classification. Given more time, it would be interesting to incorporate more of the 3D structure (i.e. use more than 3 lung slices) to help classify lung nodules. It would also be useful to see how well our classifier performs when incorporated into a segmentation system. We would then be able to produce an end-to-end system for identifying lung nodules.

References

[1] M. Antonelli and et al. Lung nodule detection in ct scans. *Proceedings of World Academy of Science, Engineering and Technology*, 1.

[2] M. Aoyama and et al. Automated computerized scheme for distinction between benign and malignant solitary pulmonary nodules on chest images. *Med Phys.*, 701(8), 2002.

[3] A. El-Baz and et al. Computer-aided diagnosis systems for lung cancer: Challenges and methodologies. *International Journal of Biomedical Imaging*, 2013, 2013.

[4] C. Jacobs and et al. Automatic detection of subsolid pulmonary nodules in thoracic computed tomography images. *Medical Image Analysis*, 18:374–384, 2014.

[5] S. Lo and et al. Artificial convolution neural network for medical image pattern recognition. *Neural Networks*, 8(7), 1995.

[6] B. C. Lowekamp and et al. The design of simpleitk. *Frontiers in Neuroinformatics*, 7(45), 2013.

[7] K. Murphy and et al. A large scale evaluation of automatic pulmonary nodule detection in chest ct using local image features and k-nearest-neighbour classification. *Medical Image Analysis*, 13:757–770, 2009.

[8] S. Napel and et al. Ct angiography with spiral ct and maximum intensity projection. *Radiology*, 185:607–610.

[9] A. J. R. Siegel, D.Naishadham. Cancer statistics. *CA: a cancer journal for clinicians*, 62(1):10–29.

[10] H. R. Roth and et al. Improving computer-aided detection using convolutional neural networks and random view aggregation. *IEEE Trans. on Medical Imaging*, 2015.

[11] S.Diederich and et al. Detection of pulmonary nodules at spiral ct: comparison of maximum intensity projection sliding slabs and single-image reporting. *European Radiology*, 11, 2001.

[12] A. A. A. Setio and et al. Automatic detection of large pulmonary solid nodules in thoracic ct images. *Medical Physics*, 42(10):5642–5653, 2015.

[13] H. Soda and et al. Limitation of annual screening chest radiography for the diagnosis of lung cancer. a retrospective study. *Cancer*, 72(8):2341–2346.

[14] S. Sone and et al. Characteristics of small lung cancers invisible on conventional chest radiography and detected by population based screening using spiral ct. *The British Journal of Radiology*, 73(866):137–145.

[15] K. Suzuki. Artificial neural networks - methodological advances and biomedical applications. 2011.

[16] K. Suzuki and et al. Computer-aided diagnostic scheme for distinction between benign and malignant nodules in thoracic low-dose ct by use of massive training artificial neural network. *IEEE Trans Med Imaging*, 24(9), 2005.