Convolutional Neural Networks for Detecting Lung Cancer

Ben Heil

4/25/2017

1 Introduction

Lung cancer, with over 200,000 new cases estimated in 2016, is the leading cause of cancer deaths [1], with a survival rate of 17.7 percent [2]. If it is detected early and removed, however, the survival rate increases dramatically. For example, when stage IA tumors are removed surgically a month after diagnosis, the ten year survival rate is around 92% [3]. This indicates that an increase in the efficiency and accuracy of lung cancer detection would decrease the mortality rate of lung cancer patients dramatically.

The standard method of screening for cancer is computerized tomagraphy (CT) scanning. This method uses x-rays to generate a series of cross-sectional images of the target based on the density of different tissues. Typically, these scans are examined visually by a radiologist in order to determine whether they contain potentially cancerous lung nodules. A "second opinion" computer system that would flag images suspected to contain lung cancer nodules would be one way to increase the accuracy of this process. Such systems exist, and are referred to generally as computer aided diagnostics systems (CADs). Currently CADs involve a pipeline that contains preprocessing, segmentation, candidate detection, feature extraction, and classification steps [4]. Though CADs are effective in classifying lung nodules, with some reaching an area under the receiver operating curve of .857 [5], their complexity makes them fragile. If any of the components of the system is disrupted, all of the downstream steps suffer.

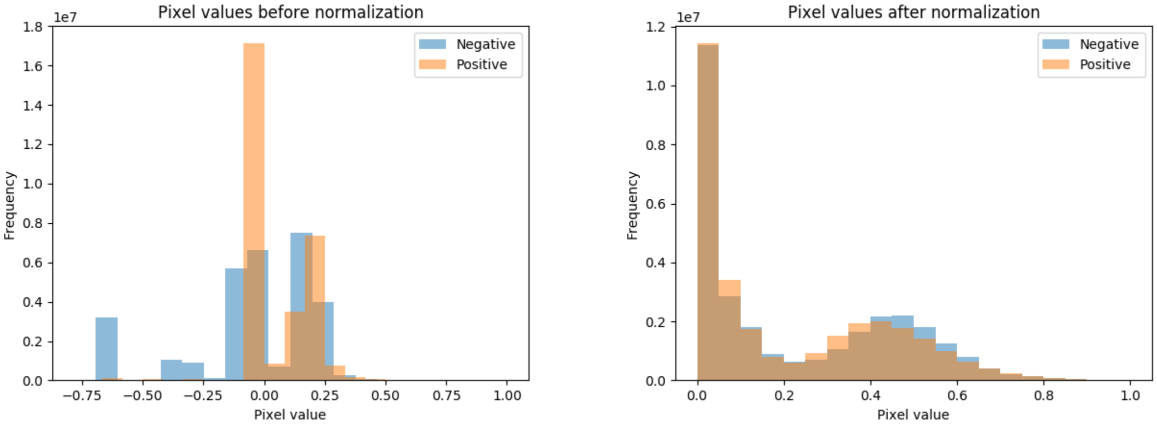
An end to end method would be a more robust alternative to the current system. Convolutional neural networks (CNNs) are one potential solution. Inspired by the architecture of the human visual cortex, CNNs have been used in cancer detection and classification successfully [6] [7]. As part of the Kaggle 2017 Data Science Bowl, I created a convolutional network that determines whether an image from a CT scan contains a potentially cancerous lung nodule.

There are a few difficulties in working with a CT image dataset. The first is the three-dimensionality of the data. Typical convolutional neural networks are designed to operate on individual images with multiple channels (typically there are three of these channels, containing the RGB value of each pixel). CT scans in DICOM format only have one channel, but consist of hundreds of cross-sectional images. Labeled images known to contain tumors are necessary for training the network due to the localized nature of early stage lung cancer. Since the Kaggle scans that are positive for cancer do not have the tumor images marked, it was necessary to find different data sources that had individual images known to contain lung cancer nodules. These images were found in the Cancer Imaging Archive [8].

Several different CNN architectures were trained and evaluated on the dataset. The first, a four layer network with two convolutional layers failed to differentiate between images with tumors and images without tumors, so models with more layers were tried. The most successful, which consisted of five convolutional layers, a fully connected layer, and an output layer, had 57.5% accuracy on the held out test set.

This result proves that it is possible to use CNNs to analyze medical images. Given better hardware and training strategies, these techniques could progress to be useful in a clinical setting.

Figure 1: Pixel value histograms before and after normalization



2 Background/Related Work

2.1 Computer Aided Diagnosis

Several approaches to computer aided diagnosis have been developed, including using support vector machines [9], random forest algorithms [10], linear discriminant analysis [5], and hybrid probabilistic sampling [11]. There have fewer attempts, however, at creating an end to end method for detecting lung cancer nodules.

2.2 Deep Learning for Medical Imaging

Deep learning teqniques for classifying cancer from medical images were developed around the same time as the ones involving manual feature generation [12]. Though there were successful models, the lack of processing power available limited the scope of their capabilities. Lo et al. even concluded the introduction to one of their papers with "...it would not be practical to search for a nodule on a radiograph by entering a large image area." [7]. However, the viability of deep learning models has increased along with computers' processing power. Convolutional neural networks have been used particularly frequently [13] in this field, and are a good candidate for further application.

3 Methods

3.1 Data

The data released for the Kaggle competition consisted of roughly 600 gigabytes of CT scan data. Images from the negatively labeled portion of this dataset were selected at random to constitute the negative portion of the training, validation, and test set. The scans were organized one per patient, with each scan consisting of between 100 and 500 512x512 black and white images. The images containing tumors came from two different studies which have their data accessible on the cancer imaging archive [8], [14], [15]. There were 246 images in the training set, 16 in the validation set, and 40 in the held out test set. Each set had an equal number of images with and without tumors.

Because the images were from different datasets, it is possible that the CNN would learn idiosyncrasies from the different scanning machinery in the studies instead of generalizable differences. To prevent this from happening, it was necessary to regularize the pixel value histograms from each image (Fig. 1). This was done by using the adaptive histogram equalization method from scikit-image on each image [16].

3.2 Software and Hardware

Preliminary CNN implementations were done using the Theano package in Python. However, in order to be able to generate and visalize new models more quickly I switched to using Keras with the Theano backend [17]. In order to handle images in the DICOM format, the pydicom library was used. To normalize the pixel frequency histograms, scikit-image was used. The machine used for training the models had an NVIDIA 460 GTX GPU.

3.3 Training

Each model was trained on approximately one thousand randomly selected batches of sixteen images, with each batch containing eight positive and eight negative training images. The update step for these batches consisted of five iterations of stochastic gradient descent with a learning rate (η) of .01 to minimize the in sample mean squared error. Only the best set of weights as determined by the minimum mean squared error on the validation set were saved to prevent overfitting. Models were compared against each other on the basis of their error on the validation set, and only the best model was evaluated with the held-out test set.

3.4 Architecture

Though the number of layers changed in each experiment, the implementation of each layer remained the same. Convolutional layers were 2-dimensional and often, though not always, had max-pooling layers after them. The max-pooling layers downsized the images by keeping only the largest value from the original image within a 4x4 window, then sliding the window over. This process was then repeated until the entire image had its size reduced by a factor in each dimension. Finally, the fully connected layers were the standard design, with their bias terms initialized to zero. Each layer used a hyperbolic tangent (tanh) activation function and batch normalization unless otherwise noted.

4 Results/Discussion

4.1 4 layer network

The first attempt at discriminating between images that contained lung nodules and ones that did not involved a four segment network. The first two segments each consisted of a 2-D convolutional layer with filters of size 3x3 and a max-pooling layer set to reduce the length and width of the input by a factor of four. The output of the second convolutional segment was then flattened into a one-dimensional vector and run through a fully connected layer with 16 nodes, and an output layer with only one node. Batch normalization was not used in this model.

This architecture was not powerful enough to discriminate between images with and without lung nodules. Both the in-sample and out of sample error converged to .25, indicating that the program minimized error by predicting the probability of each class as .5 (Fig. 2). Clearly a more powerful network was needed.

4.2 5 layer network

The next version of the network contained an extra convolutional layer for a total of three convolutional layers, a fully connected layer, and an output layer. In order to compensate for the decreasing size of each convolutional layer caused by max pooling, their filter sizes were 15x15, 7x7, and 3x3. Batch normalization [18] was added after the second, third, and fourth layers in order to help improve the accuracy. Normalization was not added after the first layer because of the training computer's memory restrictions.

The results for the larger network were very encouraging. The stochasticity of the training method is clearly visible in the loss function values and the accuracy, but both are clearly better than the base rate (Fig. 3). This model proved that a CNN was capable of descriminating between images with and without nodules.

Figure 2: Loss and accuracy on a 4 layer CNN

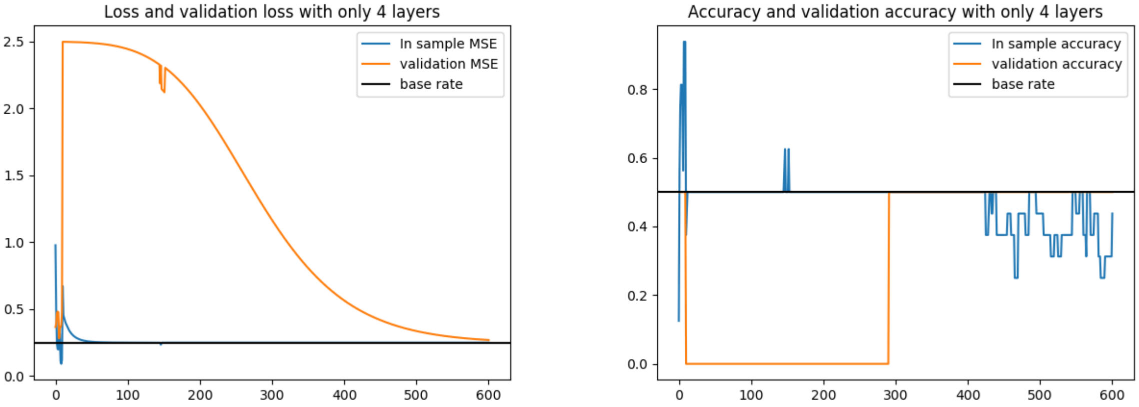


Figure 3: Loss and accuracy on a 5 layer CNN with batch normalization

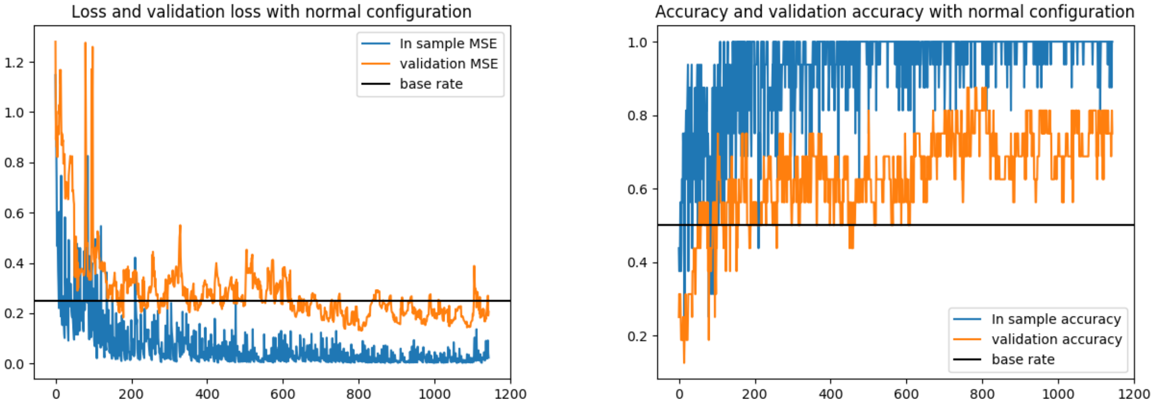


Figure 4: Loss and accuracy on a 5 layer CNN without batch normalization

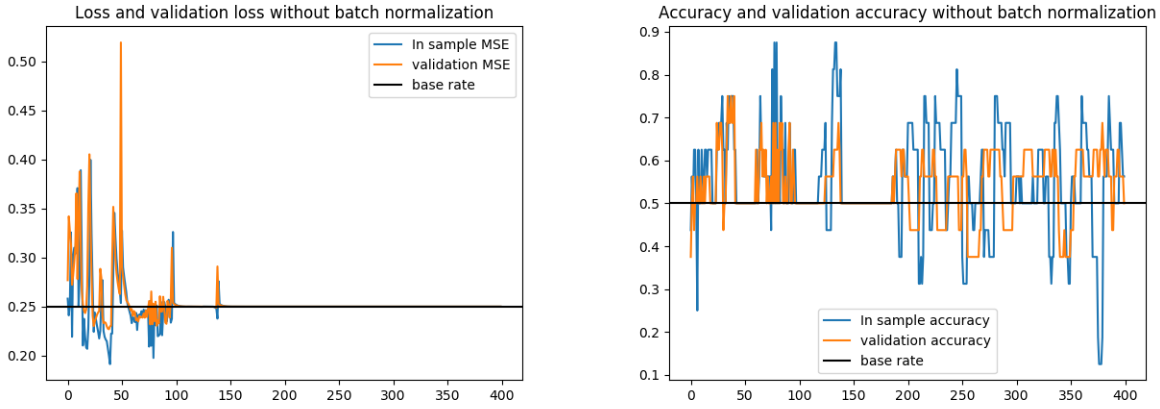
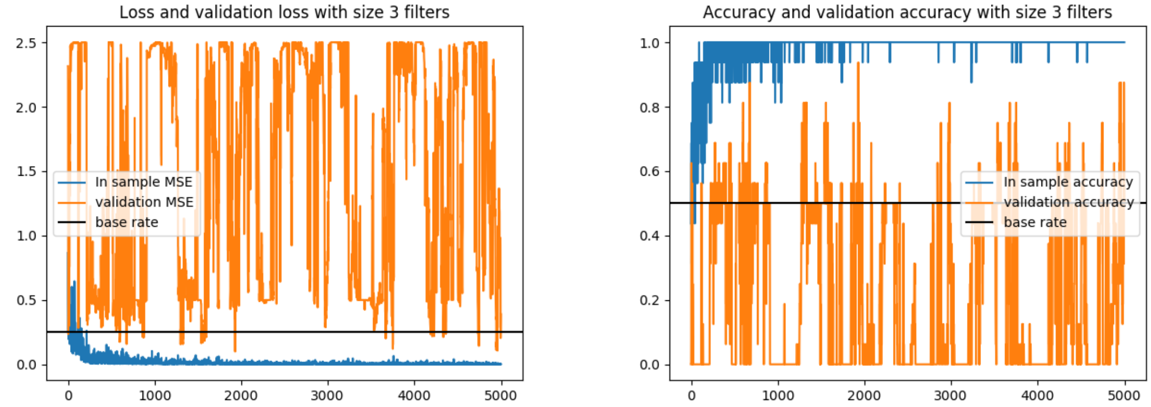


Figure 5: Loss and accuracy on a CNN with filters of size 3x3



4.3 Determining Key Components

To find the most important parts of the network, elements were modified, added, or removed to see the effect they had on the accuracy of the predictions. First the batch normalization layers were removed from the previously described 5 layer network with no other modifications. The model quickly converged to the base rate(Fig. 4), which was important for two reasons. First, it indicated that batch normalization might cause the model to converge after more iterations. It may be the case that this was caused by an increase in complexity resulting from the normalization, but I think it is much more plausible that converging to the base rate took fewer iterations because the model didn't have to learn anything at all. The second, more important result is that batch normalization is crucial for the model to work. When removed, the model ceased to differentiate between classes entirely.

Next, the normalization layers were restored, but the size of the convolutional filters in each layer was reduced to 3x3. One would expect that the program would be much less able to detect nodules because their size on the input image was often an order of magnitude larger than a 3x3 window. This expectation was borne out by the results. The 3x3 convolutional layers seem to fit noise because they can't find signal. The validation loss and accuracy measure vary greatly throughout training, but are consistently worse than the base rate (Fig. 5).

Figure 6: Loss and accuracy with an extra fully connected layer

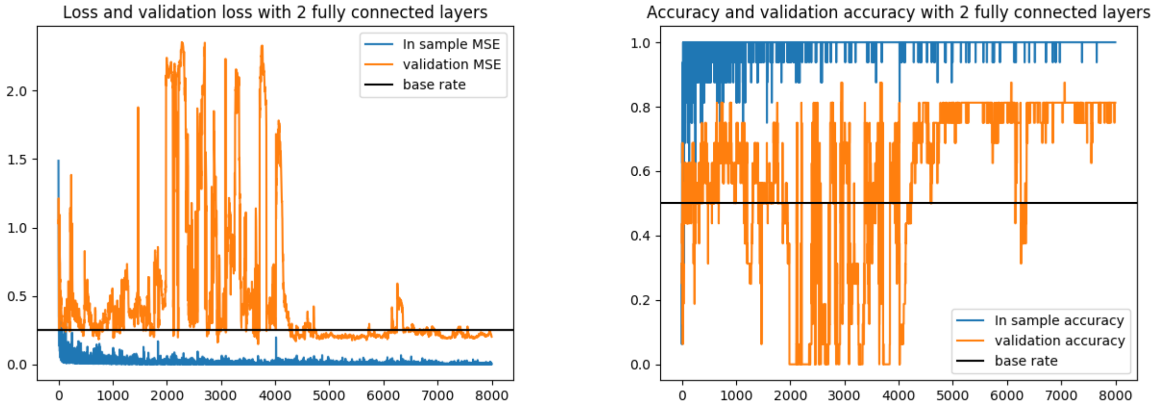
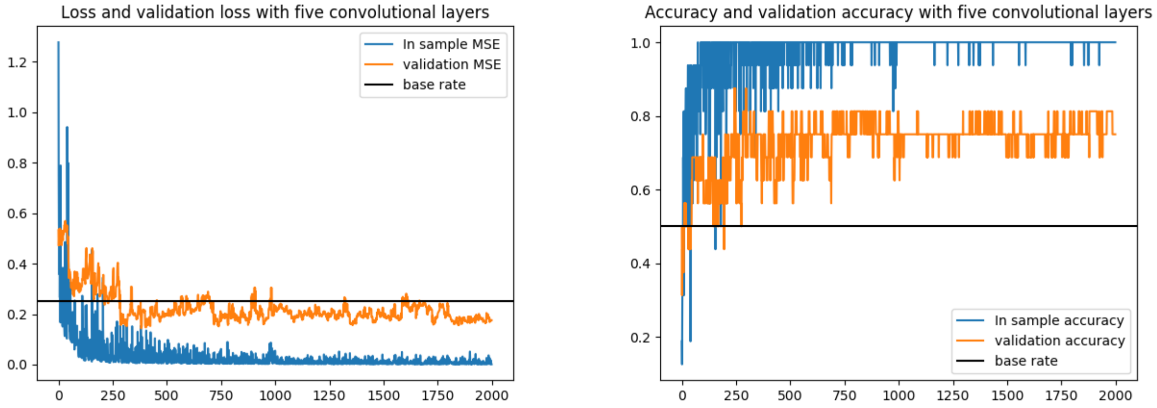


Figure 7: Loss and accuracy with five convolutional layers

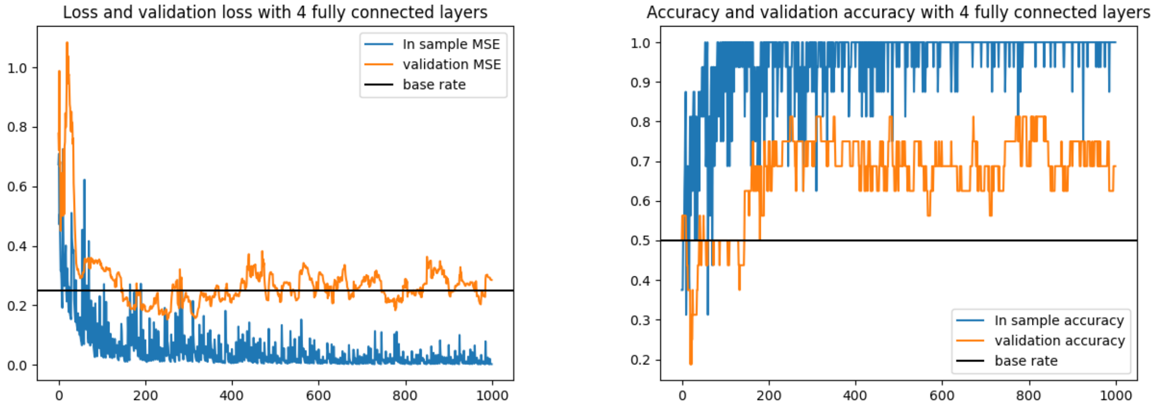


After that I began to increase the number of convolutional and fully connected layers to see if either would increase accuracy. The first experiment involved adding an extra sixteen-node fully connected layer. Though it decreased the variation in the error to some degree, the results were not substantially different from the original architecture (Fig. 6). I expected that adding convolutional layers would have a similarly negligible impact on the error. To the contrary, though it's not markedly different from the other loss graphs adding two extra convolutional layers yielded the best error out of all the different configurations (Fig. 7). Since adding convolutional layers helped, the next experiment tested whether adding both convolutional and fully connected layers at the same time would also improve the model. Surprisingly, adding two new fully connected layers to the successful architecture caused it to work substantially worse (Fig. 8). The large network was not significantly better than the base rate.

4.4 Discussion

The results of each experiment are interesting for different reasons, but the takeaway is this: the results of modifying neural network architecture are not always intuitive. Attempting to increase a model's accuracy by increasing the number of layers may make the model more capable of learning patterns, but it can also lead to overfitting. Not all free parameters behave the same way, however. Adding two extra convolutional

Figure 8: Loss and accuracy with 4 fully connected layers and 5 convolutional layers



layers improved the model's ability to detect lung nodules, while adding a fully connected layer diminished it.

When the best model, the one with five convolutional layers, was run on the held out test set, the result was 57.5% accuracy. This rate is lower than those of current CAD systems, but it is a place to start.

In the future, there are a several ways the model could be improved. The first would be to do a more formal parameter optimization. Instead of changing the number of layers or sizes of filters based on intuition, it would be beneficial to try a range of options systematically and pick the one that has the best accuracy. Another approach would be to leverage the three dimensional nature of CT scans. Three-dimensional convolutional neural networks are commonly used in video processing, where time is treated as a third dimension [19]. These techniques could easily be applied to the three spatial dimensions of a CT scan. I attempted to do exactly that, but was unable because of hardware limitations. Even when randomly sampling only 64 images from each scan and reducing the length and width of each image by a factor of 8, my graphics card did not have a sufficient amount of memory to train the model.

5 Conclusion

Though the accuracy of the final model leaves something to be desired, it proves that it is possible to differentiate between images with and without lung nodules. In the future, this kind of technology could be used to passively screen all chest CT scans to detect nodules in images that are actively screened for signs of other diseases. With more research and better hardware, it is possible that some day convolutional neural networks will be able to save lives.

References

[1] United States Cancer Statistics Working Group, *United States Cancer Statistics: 1999-2013 Incidence and Mortality Web-based Report*, 2016. [Online]. Available: `www.cdc.gov/uscs` (visited on 03/25/2017).

[2] N. Howlader, A. Noone, M. Krapcho, D. Miller, K. Bishop, S. Altekruse, C. Kosary, M. Yu, J. Ruhl, Z. Tatalovich, A. Mariotto, D. Lewis, H. Chen, E. Feuer, and K. Cronin, *SEER Cancer Statistics Review, 1975-2013*, Apr. 2016. [Online]. Available: `https://seer.cancer.gov/csr/1975_2013/` (visited on 03/28/2017).

[3] The International Early Lung Cancer Action Program Investigators, “Survival of Patients with Stage I Lung Cancer Detected on CT Screening,” *New England Journal of Medicine*, 355, no., pp. 1763–1771, Oct. 2006, ISSN: 0028-4793. DOI: `10.1056/NEJMoa060476`. [Online]. Available: `http://dx.doi.org/10.1056/NEJMoa060476` (visited on 03/26/2017).

[4] B. van Ginneken, C. M. Schaefer-Prokop, and M. Prokop, “Computer-aided Diagnosis: How to Move from the Laboratory to the Clinic,” *Radiology*, 261, no., pp. 719–732, Dec. 2011, ISSN: 0033-8419. DOI: `10.1148/radiol.11091710`. [Online]. Available: `http://pubs.rsna.org/doi/full/10.1148/radiol.11091710` (visited on 03/25/2017).

[5] T. W. Way, B. Sahiner, H.-P. Chan, L. Hadjiiski, P. N. Cascade, A. Chughtai, N. Bogot, and E. Kazerooni, “Computer-aided diagnosis of pulmonary nodules on CT scans: Improvement of classification performance with nodule surface features,” *Medical Physics*, 36, no., pp. 3086–3098, Jul. 2009, ISSN: 0094-2405. DOI: `10.1118/1.3140589`. [Online]. Available: `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2832039/` (visited on 03/26/2017).

[6] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, “Dermatologist-level classification of skin cancer with deep neural networks,” en, *Nature*, 542, no., pp. 115–118, Feb. 2017, ISSN: 0028-0836. DOI: `10.1038/nature21056`. [Online]. Available: `http://www.nature.com/nature/journal/v542/n7639/abs/nature21056.html` (visited on 03/25/2017).

[7] S. C. B. Lo, S. L. A. Lou, J.-S. Lin, M. T. Freedman, M. V. Chien, and S. K. Mun, “Artificial convolution neural network techniques and applications for lung nodule detection,” *IEEE Transactions on Medical Imaging*, 14, no., pp. 711–718, Dec. 1995, ISSN: 0278-0062. DOI: `10.1109/42.476112`.

[8] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, L. Tarbox, and F. Prior, “The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository,” en, *Journal of Digital Imaging*, 26, no., pp. 1045–1057, Dec. 2013, ISSN: 0897-1889, 1618-727X. DOI: `10.1007/s10278-013-9622-7`. [Online]. Available: `https://link.springer.com/article/10.1007/s10278-013-9622-7` (visited on 04/30/2017).

[9] P. Campadelli, E. Casiraghi, and D. Artioli, “A Fully Automated Method for Lung Nodule Detection From Postero-Anterior Chest Radiographs,” *IEEE Transactions on Medical Imaging*, 25, no., pp. 1588–1603, Dec. 2006, ISSN: 0278-0062. DOI: `10.1109/TMI.2006.884198`.

[10] S. L. A. Lee, A. Z. Kouzani, and E. J. Hu, “Random forest based lung nodule classification aided by clustering,” *Computerized Medical Imaging and Graphics*, 34, no., pp. 535–542, Oct. 2010, ISSN: 0895-6111. DOI: `10.1016/j.compmedimag.2010.03.006`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S0895611110000418` (visited on 03/28/2017).

[11] P. Cao, J. Yang, W. Li, D. Zhao, and O. Zaiane, “Ensemble-based hybrid probabilistic sampling for imbalanced data learning in lung nodule CAD,” *Computerized Medical Imaging and Graphics*, 38, no., pp. 137–150, Apr. 2014, ISSN: 0895-6111. DOI: `10.1016/j.compmedimag.2013.12.003`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S0895611113002000` (visited on 03/28/2017).

[12] B. Sahiner, H.-P. Chan, N. Petrick, D. Wei, M. A. Helvie, D. D. Adler, and M. M. Goodsitt, “Classification of mass and normal breast tissue: A convolution neural network classifier with spatial domain and texture images,” *IEEE Transactions on Medical Imaging*, 15, no., pp. 598–610, Oct. 1996, ISSN: 0278-0062. DOI: `10.1109/42.538937`.

[13] W. Shen, M. Zhou, F. Yang, D. Yu, D. Dong, C. Yang, Y. Zang, and J. Tian, “Multi-crop Convolutional Neural Networks for lung nodule malignancy suspiciousness classification,” *Pattern Recognition*, 61, no., pp. 663–673, Jan. 2017, ISSN: 0031-3203. DOI: `10.1016/j.patcog.2016.05.029`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S0031320316301133` (visited on 03/25/2017).

[14] I. Armato Samuel G., K. Drukker, F. Li, L. Hadjiiski, G. D. Tourassi, R. M. Engelmann, M. L. Giger, G. Redmond, K. Farahani, J. S. Kirby, and L. P. Clarke, "LUNGx Challenge for computerized lung nodule classification," *Journal of Medical Imaging*, 3, no., pp. 044 506–044 506, 2016, ISSN: 2329-4302. DOI: `10.1117/1.JMI.3.4.044506`. [Online]. Available: `http://dx.doi.org/10.1117/1.JMI.3.4.044506` (visited on 04/27/2017).

[15] O. Grove, A. E. Berglund, M. B. Schabath, H. J. W. L. Aerts, A. Dekker, H. Wang, E. R. Velazquez, P. Lambin, Y. Gu, Y. Balagurunathan, E. Eikman, R. A. Gatenby, S. Eschrich, and R. J. Gillies, "Quantitative Computed Tomographic Descriptors Associate Tumor Shape Complexity and Intratumor Heterogeneity with Prognosis in Lung Adenocarcinoma," *PLOS ONE*, 10, no., e0118261, Mar. 2015, ISSN: 1932-6203. DOI: `10.1371/journal.pone.0118261`. [Online]. Available: `http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0118261` (visited on 04/27/2017).

[16] S. v. d. Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "Scikit-image: Image processing in Python," en, *PeerJ*, 2, no., e453, Jun. 2014, ISSN: 2167-8359. DOI: `10.7717/peerj.453`. [Online]. Available: `https://peerj.com/articles/453` (visited on 04/26/2017).

[17] F. Chollet *et al.*, *Keras*. GitHub, 2015. [Online]. Available: `https://github.com/fchollet/keras`.

[18] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv:1502.03167 [cs]*, no., Feb. 2015, arXiv: 1502.03167. [Online]. Available: `http://arxiv.org/abs/1502.03167` (visited on 04/26/2017).

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale Video Classification with Convolutional Neural Networks," 2014, pp. 1725–1732. [Online]. Available: `http://www.cv-foundation.org/openaccess/content_cvpr_2014/html/Karpathy_Large-scale_Video_Classification_2014_CVPR_paper.html` (visited on 04/30/2017).