AUTOMATIC CLASSIFICATION OF CYTOPATHOLOGICAL LUNG CANCER IMAGES

by

Justin Whitaker

A Senior Honors Project Presented to the

Honors College

East Carolina University

In Partial Fulfillment of the

Requirements for

Graduation with Honors

by

Justin Whitaker

Greenville, NC

May, 2018

Approved by:
Junhua Ding
Department of Computer Science, College of Engineering and Technology.

Automatic Classification of Cytopathological Lung Cancer Images

Justin Whitaker
East Carolina University

***Abstract*— The goal of this project is to develop and use cytopathological lung cancer images to train an automatic classifier to label cancerous cell images as being benign or malignant. This problem is distinct from previously solved problems because our data set is composed of images wherein cells are highly overlapping. I trained a preliminary convolutional neural network which classified individual cell images on an openly available data set, and I am currently applying this model to the overlapping cell image dataset using image segmentation.**

I. Introduction

Lung cancer is one of the most prevalent forms of cancer in recent years. 6.4% of people will be diagnosed with lung cancer in their lifetime, and 527,228 Americans had the disease in 2014 [1]. It is also one of the most deadly forms of cancer. Lung cancer accounts for 13.2% of new cancer cases, and 25.9% of cancer deaths[1]. An estimated 222,500 people were diagnosed with lung cancer in 2017, and an estimated 155,870 people were killed by the disease[1]. Diagnosis requires the time of medical experts to identify malignant cancer in cytopathological or histpathological images of cell or tissue samples taken from the patient. From 2007 to 2013, 18.1% of those afflicted survived 5 years [1]. Making pathology diagnostics more accessible may help detect lung cancer in patients earlier and improve survival rates of those with the disease.

The dataset I used was provided by doctors at the Brody School of Medicine at East Carolina University. The images are clumps of cells obtained from patients with lung cancer. The images have been labelled by doctors as belonging to instances of benign or malignant lung cancer.

The purpose of the paper is to investigate the unique problem of classifying cytopathological images which have highly overlapping cells. The primary problem is to extract meaningful features from the high-resolution image dataset without losing the important local context of the individual cells. I used convolutional neural networks to create a preliminary classifier on an external single-cell dataset, and investigated segmentation methods to apply the model to the highly-overlapping image dataset.

II. Background

A. Cytopathology

The two different types of images used in automatic image-based medical diagnosis are histologies and cytopathologies. Histological images are images of a segment of tissue, whereas cytopathological images are of a clump of individual cells. Histopathology requires tissue samples obtained by an invasive surgical extraction. However, it provides more information relative to cytopathology[2]. Cytopathology is noninvasive and only requires easily-obtained cell samples[2]. Discerning malignancy of cancer from cell images is more difficult because the context of the entire tissue is lost. Developing a classifier for cytopathological images would make automatic diagnosis more accessible than a classifier for histological images.

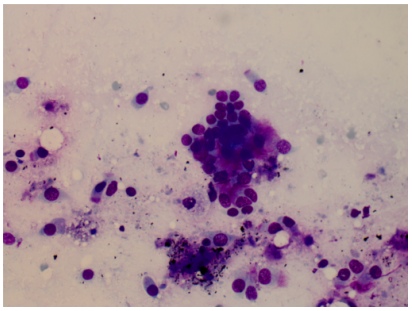

Fig. 1. An example of the highly-overlapping cytopathological images

B. Image Segmentation

Image segmentation occurs after operations to improve the images characteristics (e.g. reduce noise, emphasize edges) and before operations such as object recognition and scene interpretation. There are three categories of image segmentation techniques. These are pixel based, edge based, and region based. Pixel based techniques only use local information, and classify using light intensity. If the foreground and background on an image are different light intensities, pixel based segmentation works fine. The downside of this is that it is sensitive to factors like contrast levels and shadows. Edge based techniques detect edges to segment the image into regions. Edges occur

when the derivative of illumination is high. These methods can fail when edges do not form a complete boundary. Region based techniques are more computationally expensive than the aforementioned methods, and use global image information. They group pixels with similar attributes into regions using a homogeneity criterion such as gray level. Regions can be created either by growing from an initial seed or splitting from the entire image. The hybrid algorithm SMG (Split, Merge, Group) combines both techniques[3]. Image segmentation is an important preprocessing step for machine learning using images.

C. Neural Networks

a) : Neural networks are complex, supervised models which are able to represent highly nonlinear functions. They are composed of connected layers of simple processing units called perceptrons which generally form a directed acyclic graph. A perceptron applies an output function to a linear combination of its inputs to produce its output. Neural networks undergo learning by optimizing the synaptic weights between these perceptrons to minimize the loss of the function they are representing [4] [5].

b) : Feed forward neural networks can't learn some functions such as exclusive or because they are not linearly separable [4]. The output layer is the only layer of a network whose neuron errors are readily available after testing, which is calculated by subtracting the values they should have by the values they actually produced. Backpropagation uses the errors of the layer following a given layer and the values of the weights connecting the two to determine the errors for that given layer. This can be viewed as each neuron in the previous layer having a share of responsibility for the errors in the next layer. If one had high activation and high weights, its responsibility will be higher than if it had low activation or had low weights. Gradient descent then uses the errors of neurons to optimize their weights. It uses the derivative of the error function to find the slope needed to travel down to optimize the function by having the minimum error [4]. Backpropagation is the mechanism by which neural networks learn.

c) : When combined into a neural network, the simple building blocks of perceptrons can represent any arbitraty function, given a large enough network size [4]. Each layer's node in a standard neural network has an independent connection weight with all nodes in the adjacent layers. These all account for parameters that must be optimized as the network trains. Their ability to model complex functions comes at the cost of having to optimize all of the connections between their neurons, which becomes more costly as the network becomes deeper and wider [6]. In general, the more hidden layers a network has, the more abstract the features detected by perceptrons in the deep layers will be [5]. If a neural network is too large, it will perform poorly on unseen test data [4]. A major benefit of convolutional neural networks is that the weights between layers are not all independent, which will be discussed in more detail later.

d) : The increasing size of datasets, faster processing, new techniques and new methods of artificially increasing dataset sizes have caused the relatively recent resurgence of neural networks in machine learning tasks. Advances in hardware technology today makes training neural networks hundreds of times faster by training on GPU's instead of CPU's [7]. GPU's excel with neural networks because they require the same instruction to be executed on many different pieces of data. Neural networks are useful in medical applications because they can quickly process data and aid clinicians in making diagnoses [8]. They have been applied in many areas of medical diagnosis [8]. In the 1990's, neural networks were used to identify different types of cancer [8]. Now, neural networks have been used with many different types of data in medicine, such as biochemical markers [8]. In the related works section, I discuss more problems similar to the focus of this paper.

1) Convolutional Neural Networks:

a) : A special type of neural network, called convolutional neural networks, utilizes kernel convolutions to reduce the complexity of the model. Kernel convolutions compute the pairwise products of a kernel's weights and an input's values in a frame around the value being output, divided by the number of values in the frame [9]. Regarding images, they use a weighted average of nearby pixels to generate each new output pixel [4]. This can be expressed with the equation

$$r(i) = (s*k)(i,j) = \sum_n \sum_m s(i-n, j-m)k(n,m) \tag{1}$$

where k is the kernel, s is the image being convolved, (n,m) are 2-dimensional indices of the kernel, and (i,j) are 2-dimensional indices of the image [9]. Kernel convolution applies a filter which sweeps across the image.

b) : Weights in convolutional layers take the form of the weights of filters, which are optimized by backpropagation as with weights in a standard neural network. The filter sliding across the input image creates a local receptive field affecting each output pixel. This aids in detecting local patterns [10]. Because the filter is kept constant for each receptive local field, this acts as weight sharing for the image which reduces the complexity of the model and increases the network's efficiency [11]. Shared weights and biases make every filter

respond to the same feature at any location in the image, which reduces the impact of affine transformations on image recognition. Convolutional neural networks are ideal for image processing tasks, and were the focus of my research.

III. Related Work

A. Automatic segmentation of cell nuclei

Wurflinger et. al. worked to automatically segment cell nuclei in cytopathological images. Prior works used threshold-based techniques like the watershed algorithm, which uses differences in brightness between points to identify dividing lines in the cell image. This results in irregular edges and fails to separate touching nuclei when applied to this problem. The authors achieved a 92.6% segmentation success rate basing their efforts on Cubic B-Splines, which was suited for cell nuclei segmentation because it handles closed curves particularly well[2].

B. Deep CNN to classify skin lesions

Esteva et all applied a deep convolutional neural network to classify 757 types of diseases (e.g. acral lentiginous melanoma, blue nevus..) from 129,450 training images of skin lesions [12] in a taxonomical tree. The top level of this tree were classes benign, malignant, and non-neoplastic, with specificity increasing as you traverse the tree downwards. Their classifier performed on par with human experts, achieving 55% accuracy compared to 53% and 55% for two dermatologists. They increased the effective size of their dataset by using multiple images of each lesion taken by camera from different angles, using random rotation, and random vertical flipping. A special technique they used was transfer learning, in the form of using a pretrained Google Inception V3 CNN with the final layer removed to provide the initial weights.

C. Cancer Diagnosis from CT Scans

a) : Kaggle's Data Science Bowl 2017 was to predict from CT scans whether a patient would be diagnosed with cancer. The CT scans were 3 dimensional tomographic images. There were approximately 1300 images in the dataset. Because the labels were whether the patient was diagnosed with cancer within a year, were so high in dimension, and were so high resolution, the scans themselves were not useful for training by naively using a 3D convolutional neural network. Julian [13] and Hammack [14] worked as a team for this competition. Julian used more useful data from the external LIDC-IDRI dataset of malignancy of nodules manually assessed by doctors. This allowed them to use a sliding window over the scan being tested and to create a matrix of malignant nodule probability, which was finally used to predict if they had cancer. Their process was to normalize the CT scans, identify nodules, predict features including malignancy, and then use the feature predictions for all of the nodules in the scan to predict if the patient would be diagnosed with lung cancer. They combined 17 different 3-dimensional CNN's built with different architectures and subsets of the data. To improve efficiency, they created a model to identify regions of interest that may be abnormal. They used random 3D transformations of the 64mm**3 inputs for testing and training. They regularized their model and improved invariance to common distortions by using lossless and lossy augmentation techniques, including random rotations, transpositions, and zooms [14]. They won second place using convolutional neural networks.

IV. Methodology

a) : I created a preliminary individual cell classifier on an openly available external dataset. I used TensorFlow and TFLearn to implement the convolutional neural network classifier. I also researched methods to apply image segmentation and boxing to the highly-overlapping cell image dataset. If the overlapping cell images can be segmented properly, the individual cell classifier I developed could be applied to the highly-overlapping cell problem.

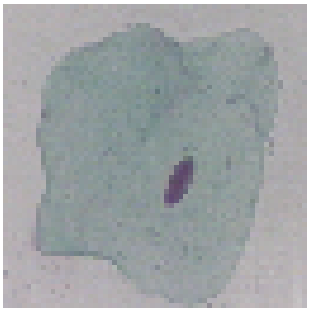

Fig. 2. An example individual benign cell from Martin 2003

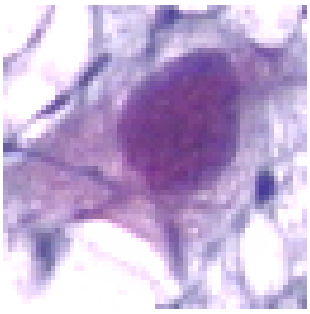

Fig. 3. An example individual malignant cell from Martin 2003

A. Data Acquisition

The dataset of labeled highly-overlapping cell images I used was provided by doctors at the Brody School of Medicine at East Carolina University.

The dataset I used to train a single cell classifier were pap smear images made openly available by Martin, 2003 [15]. This allowed me to create a model to detect malignancy in single cells without segmenting the highly-overlapping images.

B. Tools

I used the machine learning interface TensorFlow and a high-level framework TFLearn to implement my neural network. TensorFlow is both an open source machine learning interface developed by Google and the reference implementation of that interface made by Google as well. TensorFlow is written in C++, and currently has support for C++ and Python as frontend languages. Regardless of which frontend language is used, the same C++ TensorFlow code is used, and there is no difference in efficiency after the TensorFlow session has begun. Although flexible enough to do many machine learning tasks, TensorFlow is especially popular for deep learning applications. Most importantly for this project, it has built in optimization algorithms that compute the gradients of cost functions, which is the essential driver for learning in neural networks [16].

TFLearn is a Python library that provides a higher level interface for TensorFlow with an emphasis on ease of use. TensorFlow is useful because it allows programmers to perform computations common in neural networks without programming for a specific computational device. However, TensorFlow requires the programmer to explicitly declare the operations on the flow of data within a network, which results in very verbose programs. Instead of requiring the programmer to describe the computations used in a neural network, TFLearn allows the programmer to work on a higher level of abstraction by defining layers – such as input layers, fully connected layers, and convolutional layers – which are combined into a network. A network written with TFLearn's layer abstractions is more concise than TensorFlow, has less likelihood of programmer error, and is more interpretable. TFLearn is fully compatible with TensorFlow, and can be used directly with TensorFlow operations for fine-tuning. This is why I used TFLearn for experimenting with and implementing my neural networks [17].

C. Preliminary cell classifier

The purpose of my first model was to differentiate cells whose nuclei were large, dark, and irregular, to cells whose nuclei was small and regular. This differentiates cancerous and noncancerous cells. Using the dataset from [15], I merged the original seven classes into two to make classification easier for the model. I grouped normal intermediate and normal superficial into the normal class. I grouped carcinoma in situ, light dysplastic, moderate dysplastic, and severe dysplastic into the abnormal class. Finally, I ignored normal columnar cells because they were structurally different from the other cell types, and I wanted to minimize the difficulty of the preliminary classification problem. After grouping classes together, I had 145 normal and 676 abnormal cell images, totaling 821 images. The images had variable sizes, but were around 200x300 pixels in area.

I used an AlexNet architecture to classify the images into normal and abnormal classes. The code for the network is available at https://github.com/jdwhitaker/pap-smear/blob/master/train_pap_network.py.

I used random rotations between 0 and 359 degrees combined with random flipping to augment the size of my dataset. I used 80% of my data for training and 20% for validation. For optimization, I used mean squared error as the loss function.

D. Cell segmentation

I am currently still working to segment cells in the highly overlapping cell images provided by doctors at the Brody School of Medicine at East Carolina University. I attempted to use Cubic B-Splines segmentation, but it failed because there were not clear edges around cells. I ran the YOLO2 boxing algorithm on the images, but the pretrained network had no success boxing cells [18]. It may still be a viable solution, but it would need to be retrained to box cell images. Currently, I am investigating the applicability of my own segmentation algorithm. The algorithm splits the overlapping cell images into subimages, and then uses K-Means Clustering to classify each subimage as containing cells or being blank space. Clustering is performed using a simple metric, which is the standard deviation of illumination in the image. This detects whether the image has a small degree of change, and is blank, or a large degree of change, and contains cells.

V. Results

In my preliminary cell classifier on the Martin 2003 dataset. the best validation set accuracy was 92.64% using a continuous output and 100% using a thresholded output. After 50 training epochs, my training set accuracy was 88.43%, and my validation set accuracy was 87.73%. The test size was 163. There were 134 true positives, 6 false positives, 22 true negatives, and 1 false negative. The positive error rate was 0.043 and the negative error rate was 0.043, which indicates it was generating a balanced model which did not give undue favor to one label over another. Given the number of abnormal cell images totaled 676 for the entire dataset compared to 145 normal cell images, bias towards false positives is a risk it is prone to.

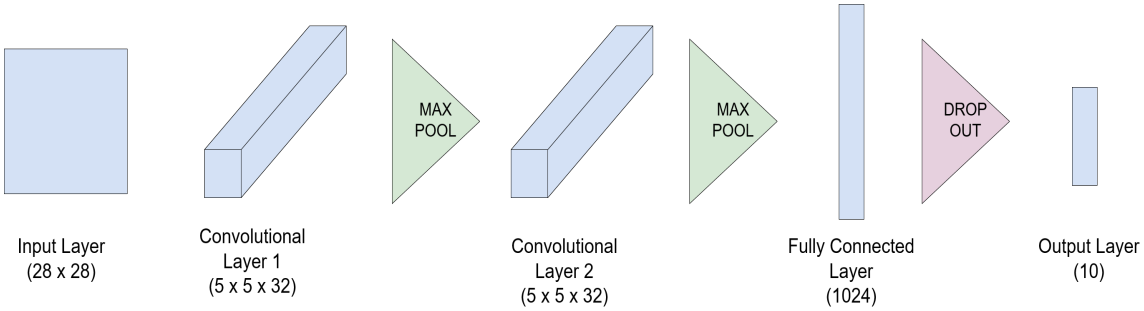

Fig. 4. The AlexNet architecture used in the preliminary cell classifier

a) : After training for 200 epochs, my training set accuracy was 91.76% and my validation set accuracy was 92.64%. After applying a threshold function to the network output to label all outputs below .5 as 0, and all outputs .5 and above as 1, I reached a 100% final classification accuracy rate on my validation set. The test size was 163, with 135 true positives and 28 true negatives, and there were no errors in classification on unseen data.

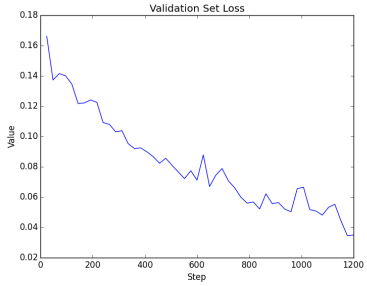

Fig. 5. Validation set loss across training epochs

My preliminary cell segmentation algorithm produced promising results by successfully identifying subimages which contain cells against blank images, but its effectiveness will be unknown until it is used successfully in a cancer malignancy classifier.

VI. Discussion

Convolutional neural networks were effective for creating a preliminary classifier on individual cancerous cell images that were pre-segmented. The highly overlapping cell images could not be fed directly into a classifier, because the data set was too small, and the images were too large in vector space. To overcome this, it will be necessary to segment the images into smaller images without losing important features. Several previously used segmentation techniques proved ineffective for this problem because of the highly-overlapping nature of the cells. Splitting the image into subimages and then using the ones which contain cells as the input to the classifier may solve the problem of overly-high dimensionality images being fed into the network. This hypothesis will be my focus continuing this research. My preliminary model shows that this task can be completed if proper cell segmentation is achieved.

VII. Conclusion

Convolutional neural networks are widely used for classifying biomedical images. Image segmentation is a necessary preprocessing step when dealing with highly-overlapping, high resolution cell images. Creating a successful cytopathological image classifier for lung cancer malignancy would greatly improve rates of early diagnosis and treatment, and would reduce the mortality rate of those affected with the disease.

Acknowledgments

I would like to thank the Honors College at East Carolina University, Dr. Junhua Ding, and NSF's Research Experiences for Undergraduates program for giving me the opportunity to perform this research.

References

[1] N. Howlader, A. Noone, M. Krapcho, D. Miller, K. Bishop, C. Kosary, M. Yu, J. Ruhl, Z. Tatalovich, A. Mariotto, D. Lewis, H. Chen, E. Feuer, and K. Cronin, "Seer cancer statistics review, 1975-2014," *National Cancer Institute*, 2017.

[2] T. Würflinger, J. Stockhausen, D. Meyer-Ebrecht, and A. Böcking, "Robust automatic coregistration, segmentation, and classification of cell nuclei in multimodal cytopathological microscopic images," *Computerized Medical Imaging and Graphics*, vol. 28, no. 1-2, pp. 87–98, 2004.

[3] L. Spirkovska, "A summary of image segmentation techniques," *National Aeronautics and Space Administration*, 07 1993.

[4] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 2010.

[5] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Upper Saddle River, NJ: Pearson, 2009.

[6] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[7] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3642–3649.

[8] F. Amato, A. López, E. M. Peña-Méndez, P. Vaňhara, A. Hampl, and J. Havel, "Artificial neural networks in medical diagnosis," *Journal of Applied Biomedicine*, 2013.

[9] V. Podlozhnyuk, "Image convolution with cuda," *NVIDIA Corporation white paper, June*, vol. 2097, no. 3, 2007.
[10] A. Karpathy, "Convolutional neural networks for visual recognition," goo.gl/tsiqLj, accessed: 2017-6-19.
[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
[12] 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," *Nature*, vol. 542, no. 7639, p. 115, 2017.
[13] J. de Wit, "2nd place solution for the 2017 national datascience bowl," https://juliandewit.github.io/kaggle-ndsb2017/, accessed: 2017-7-13.
[14] D. Hammack, "Forecasting lung cancer diagnoses with deep learning," https://goo.gl/FtHsYz, accessed: 2017-7-14.
[15] E. Martin, J. Jantzen, and B. Bjerregaard, "Pap-smear classification," 2003.
[16] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.
[17] Y. Tang, "Tf. learn: Tensorflow's high-level module for distributed machine learning," 2016.
[18] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2016.