Quantitative Feature Relevancy using Machine Learning

Diego K. Alcantara
Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ 85721
diego.k.alcantara@gmail.com

Corey J. Miner
Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ 85721
cjminer@email.arizona.edu

Abstract

This paper explores the possible benefits of applying machine learning techniques to both the classification of histological images and the analysis of the features that are most useful in said classification. This technology has applications within medical image analysis which currently relies heavily on highly trained human visual analysis. Human visual analysis is bias heavy and prone to error. Applying computational techniques to medical image analysis allows for more objective and accurate diagnosis. In addition, techniques within machine learning such as mutual information allow for a quantitative basis for the determination of features relevant to a classification label.

1 Introduction

Medical imaging is a critical component of disease diagnosis in modern medical practice. Although advances in medical imaging technologies have enabled healthcare providers to implement more effective point of care strategies, the analysis of medical images remains inefficient and highly subjective. For example, radiologists rely almost exclusively on the visual inspection of clinically acquired images to diagnose abnormal phenomena in patient tissues. As a result, any radiologist's analysis of a medical image is liable to biased influences from the radiologist's own convictions or professional background, leading to inconsistencies in medical diagnoses.

In computer science, the field of image processing utilizes a wealth of mathematical algorithms and processing routines to perform quantitative analyses on visual data in the form of images. Incorporating image processing in medical imaging processes can allow for a quantitative analysis of medical image data, begetting more reliable and consistent diagnoses.

1.1 Feature Selection Methods

When a human analyzes cells in order to make a diagnosis, they are primarily looking at the quantity of circular like features (blobs), both within and around the nucleus of the cell, and the intensity of color within those blobs. Given that, a feature extraction routine was developed to extract said features. The features used for comparison and analysis with mutual information include: the number of blobs within an image, the average area of those blobs, the average red, average green, and average blue color within those images, and average luminance (similar to intensity) of the image.

1.2 Deep Learning Methods

The system uses a convolutional neural network (CNN) to classify the images. A CNN is a fully connected, multilayer learning algorithm with at least one convolutional layer. The convolutional layers allow nodes to locally share weights and are "a many to one" mapping. This reduces overfitting and complexity at each node, allowing systems to conquer more complex problems. CNN's are especially good at identifying images because the nodes share localized weights. In an image, the likelihood of two pixels next to each other being related is relatively high. Locally sharing this information in the form of weights allows the neural net to make a more informed decision about one pixel in an image based on the pixels in the area. With the use of this CNN and the selected features, we are able to classify the type of white blood cells in an image.

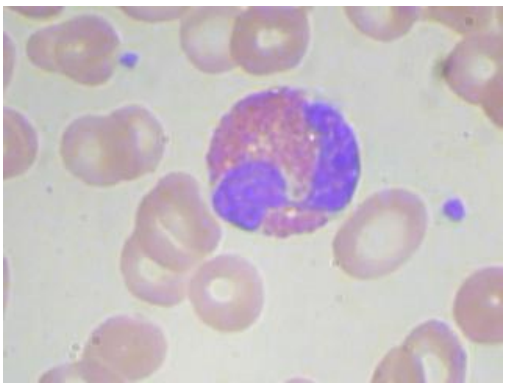
2 Related Work

The article, Classifying White Blood Cells With Deep Learning [2], is highly similar to this research. The article discusses how classifying different medical images currently is primarily done with culture counting and direct human inspection of a sample. They discuss how it may be cheaper, faster, and more accurate to use a Deep Learning algorithm to identify different types of diseases in blood samples rather than using the current methods of identification. In the article, they are specifically looking at classifying a white blood cell as polynuclear or mononuclear. They trained a convolutional neural net on white blood cell images and obtained an accuracy of 98.6% over 20% of their data set for 20 epocs with a batch size of 32. This seems rather high. As one reads further, the data set had only 71 images and very few were mononuclear cells. To boost the size of their training data, they rotated, skewed, and shifted images already in the dataset to create "unique" images. Upon further analysis, it seems the data was picked in a very particular way to enable the high results. They go on to write that this work has not been peer reviewed and, with other datasets, the accuracy is not nearly as good. They state when training with four classifications of white blood cells their accuracy is only 86%.

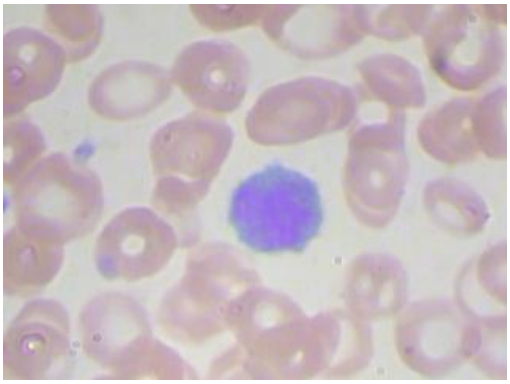
3 Methods

3.1 Image Set

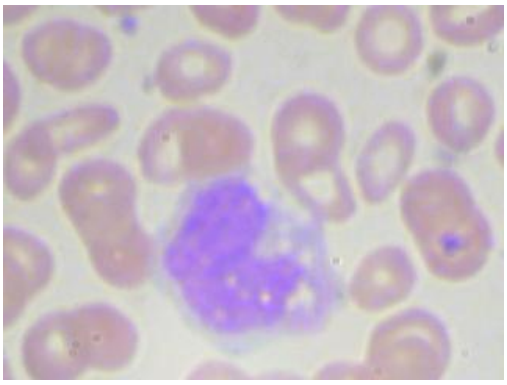
The dataset we are using consists of white blood cells split into four classes (Eosinophil, Lymphocyte, Monocyte, Neutrophil). This data set consists of roughly 625 images from each class for a test set, and roughly 2500 images as a training set. A sample of each class is given below.



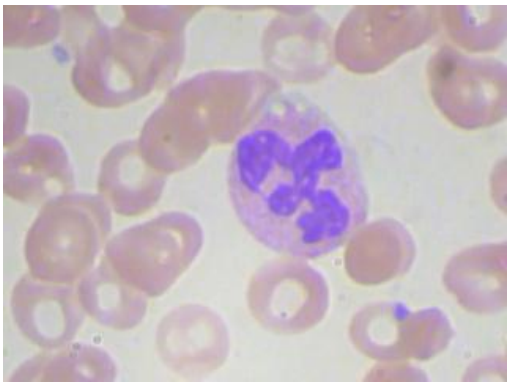
Picture 1: Sample of Eosinophil



Picture 2: Sample of Lymphocyte



Picture 3: Sample of Monocyte



Picture 4: Sample of Neutrophil

3.2 Image Feature Extraction

3.2.1 Color and Luminance Feature Extraction

Extracting the different color channels from an image was a rather elementary process. Within the openCV image library there exists a function that automatically gets the red, green, and blue values from an image. Then, by calling the .mean() method, the average colors across the images could be evaluated.

Obtaining the average luminance value was also relatively easy. Luminance is a weighted correlation of RGB value to the photopic response of the human eye (i.e. how the human eye perceives each color). Luminance value is given by

Luminance = (0.2126*R + 0.7152*G + 0.0722*B)

3.2.2 Blob Detection and Size Calculation

Extracting the circular, blob-like features was a more involved process. There are several ways to detect blobs computationally, First is the Laplacian of Gaussian given by first convolving the image with a Gaussian kernel

$$g(x, y, t) = \frac{1}{2\pi t} e^{-\frac{x^2+y^2}{2t}}$$

At a particular scale t to get a scale space representation given by

$$L(x, y; t) = g(x, y, t) * f(x, y)$$

And applying the Laplacian operator

$$\nabla^2 L = L_{xx} + L_{yy}$$

Next, is the Difference of Gaussian approach which is an approximation of the Laplacian of Gaussian given by

$$\nabla^2_{norm} L(x, y; t) \approx \frac{t}{\Delta t} \left(L(x, y; t + \Delta t) - L(x, y; t) \right)$$

Finally, the determinant of Hessian given by

$$\det H_{norm} L = t^2 (L_{xx} L_{yy} - L^2_{xy})$$

The Scikit Image (SKImage) library contains functions which calculate all three these. Thus, all three algorithms were used in the full image set analysis. The functions returns an array of x,y coordinates of the center of the center of the blob as well as the radius of said blob. This information allowed us to calculate the area of each blob with ease.

3.3 Neural Network Architecture and Optimization

Table 1: Convolutional Neural Network Architecture

Batch	Opt	Reg	Reg Const	Shape		Epochs	LR	Acc
32	rmsprop	none	n/a	Dense Layer		25	0.01	83%
				Input Dim: 32	Output Dim: 32			
				Dense Layer				
				Input Dim: 32	Output Dim: 64			
				Dense Layer				
				Input Dim: 64	Output Dim: 4			

For this model the categorical cross-entropy loss function with L2 weight decay regularization was used

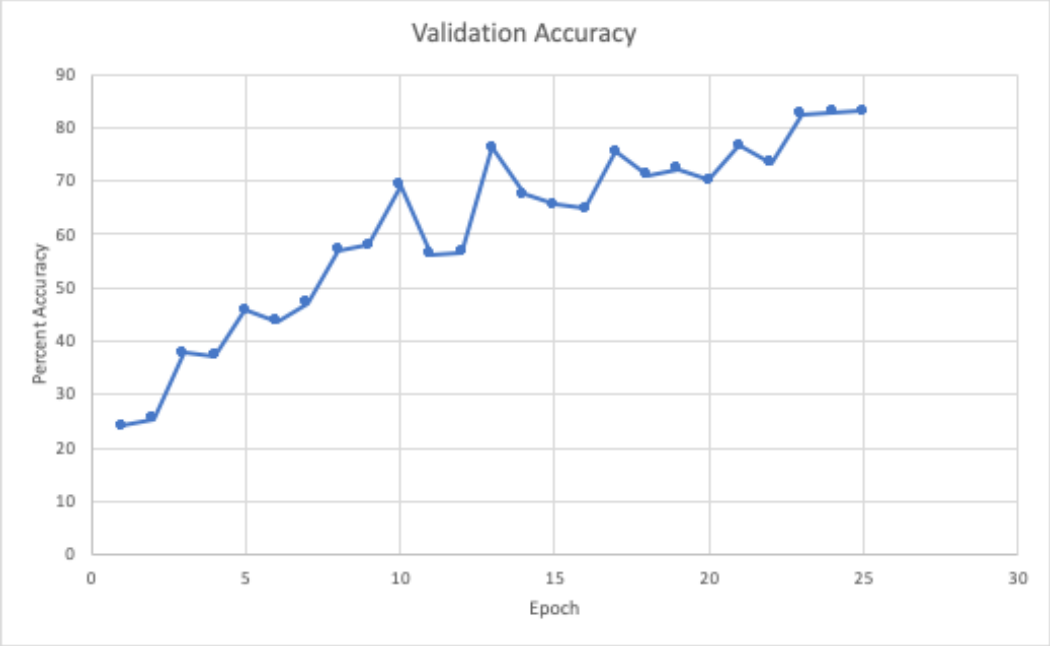
$$E = -\sum_{i=1}^{N_{samples}} crossEntropy(x_i, y_i) + \lambda \sum_{j=1}^{N_{layers}} \sum_{k=1}^{N^j_{units}} \sum_{l=1}^{N^{j+1}_{units}} (w^j_{k,l})^2$$

The optimizer used here was the rmsprop optimizer. This optimizer is a good a choice for this application as there is much redundancy amongst the images given that they are so similar [1].

> "rprop is equivalent to using the gradient but also dividing by the size of the gradient. rmsprop: Keep a moving average of the squared gradient for each weight:
>
> $$MeanSquare(w, t) = 0.9\ MeanSquare(w,\ t-1) + 0.1 \left(\partial E / \partial w (t) \right)^2$$
>
> Dividing the gradient by MeanSquare(w,t) makes the learning work much better. [1]"

4 Results

4.1 CNN Accuracy



Graph 1: Validation Accuracy of CNN used for classifying white blood cells

Graph 1 shows a gradual and rather unsteady accuracy growth of the model overtime. After 25 epochs a validation accuracy of 83% was obtained. The code that this project was based off of only trained the model of 20 epochs before stopping on the four white blood cells. The blog post claims 86% accuracy on classifying the four white blood cell classes, but we were unable to replicate this exact result. This could be due to our choice of training images differing from theirs.

4.2 Mutual Information Results

Table 2: Mutual Information: Scores Feature vs Class

Number of Blobs	Average Area	Average Red	Average Blue	Average Green	Luminance
0.62900	0.57700	0.00932	0.03370	0.02660	0.03930

Table 2 shows the mutual information scores amongst the feature that were tested as they relate to the class label. The scores for the RGB channels and luminance calculation all show much lower mutual information scores than the number of blobs and average area of those blobs. This implies that when differentiating between the four classes of white blood cells, color is much less useful than the number of blobs and the average area of those blobs. This makes sense as when one looks at Pictures 1-4, there is not an immediately obvious differentiation when looking at just color. The number of blobs and the average area of those blobs had much higher scores implying these features are a stronger differentiating characteristic than any of the color channels. Exactly whether this number makes the feature “useful” is differentiation requires further research.

Table 3: Mutual Information Scores: Feature vs Feature (4-Class)						
	Number of Blobs	Average Area	Average Red	Average Blue	Average Green	Average Luminance
Number of Blobs	4.8700	1.2600	0.0874	0.0922	0.0683	0.0640
Average Area	1.2600	6.3100	0.0105	0.0477	0.0262	0.0452
Average Red	0.0900	0.0105	6.3100	1.4500	1.3700	1.6800
Average Blue	0.0902	0.0477	1.4500	6.3100	1.7100	2.6800
Average Green	0.0675	0.0262	1.3700	1.7100	6.3100	1.8800
Average Luminance	0.0649	0.0452	1.6800	2.6800	1.8800	6.3100

Table 3 details the mutual information score between features as they relate to each other. That is, how closely is one feature related to another feature. It should be noted that 1) the scores along the diagonal of the table should be ignored as that is the mutual information score of a feature with itself which would logically be high and therefore doesn't add any useful information and 2) this table is symmetric so reading the top diagonal of the table will give you the same information as reading the bottom diagonal.

This table shows that the color channels (and by association the luminance) are all very related to one another and that average area of a blob is closely related to the number of blobs. In addition, it is observed that the RGB channels and luminance values are not closely related to the number of blobs nor the average area of them. All of these details make sense. All of the images have a similar pink-ish, purple tint to them and therefore they all should have highly correlated RGB values. The correlation between number of blobs and average area of said blobs also makes sense in that the average area calculation will vary with the number of blobs discovered within the image. Furthermore, color has logically very little relation to a quantity of blobs.

What these tables show is a correlation between human observation and computational, quantitative analysis. Applying this technology further could act as a method of redundancy for verifying preconceived ideas of histological cell analysis. It can be extended as a method by which to verify hypothesis about what features may be relevant in a given image.

5 Conclusion

The four image types presented in the dataset used this paper are very similar to one another. This is reflected in the mutual information scores obtained through this experiment. The mutual information scores obtained in Table 2 are seemingly small. This could imply that the number of blobs nor the average area of those blobs are particularly strong differentiators for this type of application. If that is the case, color would be even less of a strong differentiator, although this is obvious both from the data and visual inspection. The degree to which the mutual information score correlates to a strong or weak differentiator requires more research applying these ideas to histology as well as comparison to a more robust set of features. However, this paper demonstrates

a powerful idea of applying quantitative techniques to medical image analysis in order to obtain new insights into cell analysis and diagnosis.

Acknowledgments

We thank Team 18075 of the University of Arizona 2018-2019 Engineering Senior Design class (Diego Alcantara, Pedro Alcaraz, Andrew Burger, Xinyu Li, and Adriana Stohn) for their work on parts of the code base of this project. We focus our thanks on Andrew Burger for his contribution to the CNN and setting up of mutual information tables and Adriana Stohn for her contribution to the feature extraction.

References

[1] Hinton, G. (n.d.). Neural Networks for Machine Learning. Retrieved from http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

[2] Parthasarathy, D., & Parthasarathy, D. (2017, March 29). Classifying White Blood Cells With Deep Learning (Code and data included!). Retrieved from https://blog.athelas.com/classifying-white-blood-cells-with-convolutional-neural-networks-2ca6da239331