Machine Learning Applied to the Detection of Retinal Blood Vessels

Alex Yee
Stanford University
alexayee@stanford.edu

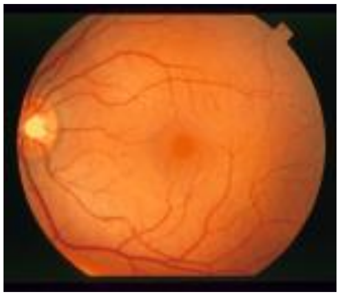
ABSTRACT: The field of ophthalmology (the study of the eye) has increasingly turned to medical imaging to play an important role in diagnosing diseases. Widespread medical conditions can be identified with only pictures of the eye using computer automated processes. Determining the segmentation of the circulatory system in the eye is difficult for doctors as the task of distinguishing blood vessels by simply observing retinal images has proven to be challenging without the aid of technology. Several morphological features of retinal veins and arteries, like diameter, length, branching angle, and tortuosity, have diagnostic relevance and can be used to monitor the progression of diseases [1]. This paper details the process and results of an attempt to improve upon the accuracy of retinal image segmentation to aid doctors in diagnosing diseases of the eye using supervised (support vector machine) and unsupervised (modified k-nearest neighbor) machine learning algorithms.

INTRODUCTION: Many common medical conditions associated with the eye can be efficiently diagnosed by doctors through the observation of retinal images. This process has the potential to improve through the application of machine learning techniques; altered images that highlight the blood vessel patterns increase doctors' ability to correctly diagnose various medical conditions such as diabetes, hypertension, and Arteriosclerosis [2]. Furthermore, it may be possible to entirely automate the detection of eye diseases. One way to alter retinal scans to distinguish the blood vessels from the rest of the image is to establish a classification problem in which each pixel is labeled as representing a blood vessel (positive) or representing any other part of the eye image (negative).

Various features are available for an algorithm of this type: RGB values of each pixel, pixel location, overall curvature, shading, contrast, and many more. Existing machine learning methods approach this problem through the use of Support Vector Machines (SVMs) with feature vectors comprised of some function of the RGB values of pre-processed image pixels among other variables. Currently, the industry standard is approximately 75% accuracy of each positive image pixel [3] compared to hand-drawn blood vessels carefully constructed by experts. This paper documents the construction and application of an SVM that nearly matches this industry standard generalization error using publicly available data. In addition, an unsupervised k-Nearest Neighbor (k-NN) algorithm is developed to streamline the production of segmented images with the ultimate goal being complete automation of the eye disease diagnosis process.

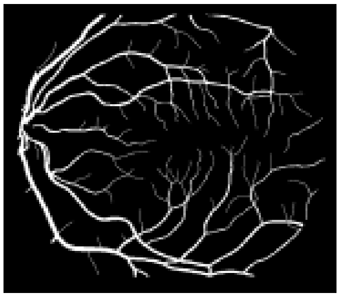
DATA: Retinal images publicly available through Clemson University [4] were used to train and test machine learning algorithms to detect blood vessels. A variety of different blood vessel patterns, image lighting, and eye size were represented by these images. The lack of consistency displayed by these retinal scans reveals the difficulty in distinguishing blood vessels and non-blood vessels. Figure 1 below shows a fairly typical example of a retinal scan—the blood vessels are slightly darker than the rest of the image.

Figure 1. Unprocessed Retinal Scan



In addition, this data set generously provided images hand-drawn by expert ophthalmologists. These images contain the expert's estimation of blood vessel location; training labels were generated for each image using its corresponding expert-drawn image. For the purposes of this study, the expert labels were considered to be the ground-truth locations of blood vessels in the given retinal scans. This data set yielded twenty images and associated labels. Figure 2 below shows the expert-drawn image for the retinal scan shown in figure 1:

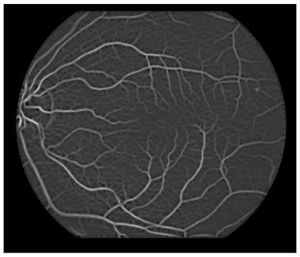
Figure 2. Expert Drawing used for Training Labels



Training labels were generated from figure 2 by assigning the value of -1 to pixels colored black (0) and 1 to pixels colored white (255).

FEATURES: The retinal scan images above were used to generate the following set of three features for each pixel. These features were chosen to maximize the effectiveness of the models while limiting the number of total features for speed of calculation. To limit noise in these calculations, image processing was completed on the raw images. Specifically, images were resized, a Gaussian filter was applied, illumination effects were removed, and RGB values were reduced to grayscale. The data was taken from the pictorial form shown in figures 1 and 2 to RGB values through the `imread` function in MATLAB [5]. These operations yielded retinal image matrices with many noisy features removed. An example is shown in figure 3 below (based off same image as figures 1 and 2).

Figure 3. Retinal Scan After Image Processing



With these simplified images, various features were tested for correlation with blood vessel locations. The most effective three variables listed below proved to yield the most accurate results while limiting the algorithmic complexity.

Grayscale Intensity – RGB values proved to be fairly noisy, so the grayscale values represent the intensity of each pixel. The value is calculated as a function of the RGB values from the raw image for each pixel using the `rgb2gray` function in MATLAB.

Gradient Magnitude – This value is calculated as the norm of a vector containing the gradient of the image in the x and y direction; it is used to detect edges in the image. Equation 1 below details the calculation for gradient magnitude an image (this calculation provides a gradient magnitude value for every pixel in an image).

$$||\nabla F|| = \sqrt{{F_x}^2 + {F_y}^2} \qquad \text{Eq. 1}$$

The partial differentials appearing in equation 1 were calculated for each gray scaled image matrix using the `imgradientxy` function in MATLAB.

Maximum Eigenvalue of Hessian Matrix – Determined to be useful in image detection by prior research [3], this value is calculated for each pixel using second derivatives of the matrix as shown in equation 2 below.

$$\lambda_+ = \frac{1}{2}\left(F_{xx} + F_{yy} + \sqrt{\left(F_{xx} - F_{yy}\right)^2 + \left(2F_{xy}\right)^2}\right) \qquad \text{Eq. 2}$$

The partial differentials appearing in equation 2 were calculated for each gray scaled image matrix using multiple iterations of the `imgradientxy` function in MATLAB.

The procured retinal scan images contained 605 × 700 pixel images, making 423,500 total pixels. Each of these pixels had an associated grayscale intensity, gradient magnitude, and maximum eigenvalue. Therefore, each of the 20 retinal scans was represented by 423,500 separate 3 × 1 feature vectors.

SVM ANALYSIS: To better understand current methods used to detect retinal blood vessels, SVM analysis was performed. The model was trained with the three dimensional feature vectors and data set described in previous sections. Testing was conducted using cross validation since a limited number of samples was available. SVMs fit this application well because they are designed to maximize the margin between the positive and negative examples (in this case, pixels that represent blood vessels and pixels that represent other parts of the eye).

The SVM was implemented on the image above by solving the dual version of the regularization problem. Non-symmetric regularization was added to combat over fitting while maintaining control over the relative numbers of false positives and false negatives. SVM*light*

[6] was used to construct the SVM according to equations 3-5:

$$\frac{1}{2}w^T w + C_1 \sum_{i:y_i=1} \xi_i + C_{-1} \sum_{i:y_i=-1} \xi_i \qquad \text{Eq. 3}$$

$$y_i(w^T x_i + b) \geq 1 - \xi_i,\ \mathrm{i} = 1, 2, \ldots, \mathrm{N} \qquad \text{Eq. 4}$$

$$\xi_i \geq 0,\ \mathrm{i} = 1, 2, \ldots, \mathrm{N} \qquad \text{Eq. 5}$$

The values of C_1 and C_{-1} were chosen such that the ratio of C_1 to C_{-1} was equal to the number of true positive values divided by the number of true negative values as given by an expert drawing. Various cost coefficient inputs to the SVM^{light} solver were tested in an attempt to optimize the algorithm. Since this application contains substantially more negative examples than positive examples, the value of C_1 (0.14) was chosen to be much smaller than C_{-1} (1.63).

Twenty images were available with hand drawn labels; with such a small number of training examples, hold-2-out cross validation was used to train and test the model for 190 combinations tested in total.

MODIFIED k-NN ANALYSIS: The k-NN algorithm was modified to be unsupervised by randomly initializing a subset of pixels and their very near neighbors. These clustered points were then used to begin the regular k-NN algorithm (where each pixel is classified according to majority vote of its k nearest neighbors), which was repeated until convergence. Distance was defined as shown equation 6 below:

$$d(x_i, x_j) = \left\| x_i - x_j \right\|_2 \qquad \text{Eq. 6}$$

The *x*'s in this equation represent the feature vectors associated with specific pixels. Essentially equation 6 calculates the distance between two pixels to be the square root of the sum of squares of the difference between each feature of those pixels. This works well in this case because typical feature values are all on the same order of magnitude. The k-NN algorithm ensures that pixels that share the most similar features as measured by this metric will be grouped together.

To optimize this algorithm, three values were determined through trial and error: the number of points to initially cluster, the number of nearest neighbors to consider when initializing, and the total number of nearest neighbors to consider when classifying pixels.

The first number of points to initially cluster and the number of nearest neighbors to cluster along with these initial points were set to minimize the number of iterations to convergence (the total number of iterations was capped to ensure that the algorithm always produced a result in a reasonable amount of time). A random selection of 1,000 pixels were randomly classified along with their 10 nearest neighbors. This corresponds to 10,000 out of 423,500 (~2%) total pixels. Once these values are set, the k-NN algorithm could be implemented normally.

The value of total number of neighbors to consider when classifying pixels greatly affects both the accuracy and runtime of this algorithm. A value of 50 was determined empirically while considering this tradeoff between performance and speed. While this value results in a sub-optimal runtime, it yields accuracy that is slightly superior to existing SVM methods.

Once all of the clusters converged to the two classes of vessel and non-vessel, one post-processing operation was completed to ensure that the image produced matched the dataset training label images. This operation involved setting the cluster with the larger number of member pixels as the negative cluster (value of 0) and the other cluster as the positive cluster (value of 255). This ensured that the blood vessel predictions would be white and the background would be black since all retinal scans contain more non-vessel pixels than vessel pixels.

RESULTS: The SVM was tested using hold-two-out cross validation with 20 total images. The average training error was 1.32% and the average testing error was 5.09%. The modified k-NN algorithm was used to evaluate the same 20 images; Average testing error was 9.28%. Table 1 below shows the confusion matrix for each of these models compared to the expert labels obtained from the data set.

Table 1. Confusion Matrix for SVM Analysis

		SVM Model Prediction		**k-NN Model Prediction**	
		0	1	0	1
Expert Labels	0	89.23%	2.76%	84.57%	7.42%
	1	2.33%	5.68%	1.86%	6.15%

Note that the SVM model is ~95% accurate in its general predictions, but when only considering whether a pixel is representing a blood vessel, improvements can be made as only 71% of positive labels were correctly identified. Similarly, the modified k-NN model was ~90% accurate in general predictions, but correctly classifies over 76% of the blood vessel pixels. Figure 4 below shows the prediction of the SVM for the retinal image shown above in figures 1, 2, and 3. Figure 5 shows the modified k-NN prediction for the same retinal scan. Test labels are reproduced for comparison.

Figure 2. Expert Hand Drawing of Retinal Scan

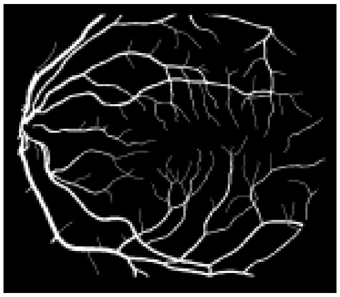


Figure 4. SVM Model Prediction

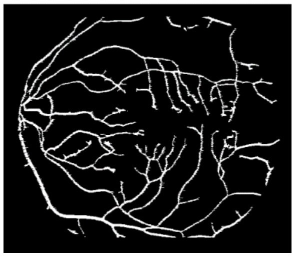
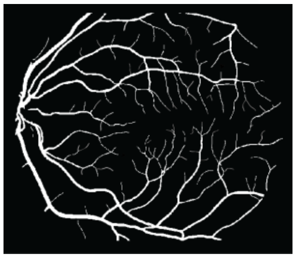


Figure 5. Modified k-NN Model Prediction



DISCUSSION: While the SVM outperformed the k-NN algorithm in terms of average testing error, table 1 reveals that the k-NN algorithm more accurately identified blood vessel pixels. It essentially traded off false positive predictions for true positive predictions. Because about 90% of pixels are not blood vessels, this tradeoff negatively impacted total accuracy more than it improved blood vessel detection accuracy. This aggressive approach is evident when observing figures 4 and 5, as the SVM model predicts the major blood vessels but misses many of the small branches toward the middle of the image. On the other hand, figure 5 reveals that the modified k-NN model correctly predicts many of the small branch blood vessels, but does so while falsely predicting many more branches.

Since the ultimate goal of this research is to automate the diagnosis of eye diseases, it is difficult to tell which algorithm is more useful in the long run. The modified k-NN model certainly identifies a higher percentage of the pixels deemed to be blood-vessels by the experts, but it does so with a few major costs. First, as mentioned above, it sacrifices accuracy on the non-blood vessel pixels by predicting branches that don't actually exist. Second, because training data isn't used for this model it takes a while to run for each image. In practices, the extended runtime could severely limit the effectiveness of the modified k-NN algorithm. Third, the unsupervised approach is a bit tougher to adjust as the only parameter that really affects the prediction is the number of nearest neighbors considered for each pixel. As this number is raised, the runtime increases but accuracy increases level off around 50 nearest neighbors.

The SVM algorithm, while failing to meet the current standards of blood vessel detection maintains a relatively small runtime for individual images and provides the requisite flexibility to change the output by adjusting the cost parameters in the model. One downside of this approach is extensive amounts of time are required to train the model on enough images to achieve reasonable accuracy. This threshold number of images can be quite high due to the high variance of lighting between different retinal scans. All things considered, the modified k-NN algorithm seems to work better for this application, but the SVM offers more potential for future improvement.

Each of these algorithms could be improved by adding more features to the feature vectors. However, this has a dramatic downside due to the large number of pixels in each image (423,500). Additionally, there calculation of features that could be useful for this application gets very complicated. Online feature selection could be used to efficiently find extra features [7], but it does not address the efficiency issue that is pertinent to the retinal blood vessel segmentation problem.

CONCLUSIONS: In the big picture, the goal of this research is to work toward a fully automated detection process for various eye diseases. While the modified k-NN algorithm is slightly less accurate overall, it is worth pursuing because it does not require a training set. The downside of the modified k-NN algorithm is that it takes a relatively long time to run on each image which might not be practical in the field. For this reason, it would be beneficial to further modify the presented SVM or apply a different kind of unsupervised algorithm to increase speed and accuracy.

FUTURE WORK: To improve upon the SVM, a neural network scheme for pixel classification might be a good direction to pursue given the algorithm's success in various other image processing fields. In addition, more work could be done to refine the features used to improve accuracy of both models. Full utilization of the methods to segment retinal blood vessels described in this paper will not be complete until they can work in conjunction with some other software that helps to diagnose the diseases of the eye. When this work is more developed, the SVM and modified k-NN algorithms can be modified to yield more applicable results.

REFERENCES

[1] J. J. Kanski, Clinical Ophthalmology: A Systematic Approach. London, U.K.: Butterworth-Heinemann, 1989.

[2] Bhuiyan, A., Nath, B., Chua, J., “An Adaptive Region Growing Segmentation for Blood Vessel Detection from Retinal Images.” Proceedings of the Second International Conference on Computer Vision Theory and Applications, Barcelona, Spain, March 8-11, 2007 - Volume 1

[3] Ricci, E., Perfetti, R., “Retinal Blood Vessel Segmentation Using Line Operators and Support Vector Classification.” *IEEE Transactions on Medical Imaging*. 26(2010).

[4] Hoover, A., M. Goldbaum. "STructured Analysis of the Retina." The STARE Project. Clemson University, n.d. Web. 17 Nov. 2014.

[5] MATLAB and Statistics Toolbox Release 2014a, The MathWorks, Inc., Natick, Massachusetts, United States.

[6] T. Joachims, Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

[7] Glocer, K., Eads, D., Theiler, J. “Online Feature Selection for Pixel Classification.” Proceedings of the 22nd International Conference on Machine Learning (ICML), 2005