Nasheha Baset

Arianna Smith

Kevin Tran

Date: December 4, 2019

Feature Extraction and Machine Learning to Classify White Blood Cells

Introduction

In this project, we were given images of microscope slides of peripheral blood of normal subjects. The task was to design an algorithm to recognize and classify neutrophils, lymphocytes, and monocytes. For training, 20 original images, the corresponding truth images, and labels were provided for each type of white blood cell (Figure 1). For testing, 10 original images were provided for each type of white blood cell. Various methods for feature extraction were performed and then a machine learning classifier was applied to classify the images. Details are described below.

Description of the work

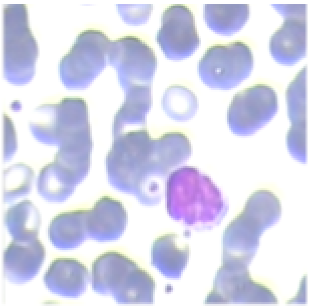

Figure 1. Example image of a lymphocyte

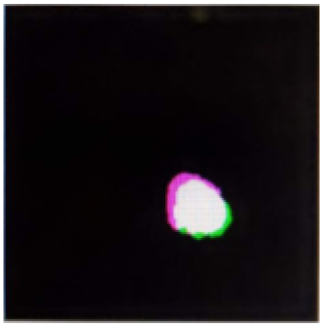

Figure 2. Nucleus segmentation overlayed with nucleus truth

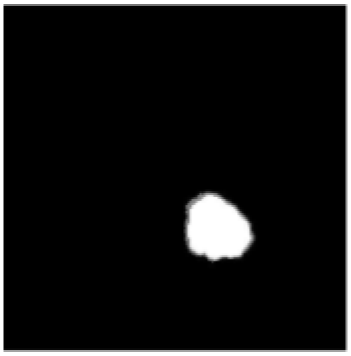

Figure 3. Overlay of cytoplasm and nucleus segmentation.

1. Segmenting the nucleus

   The images were converted from RGB space to LAB space and then normalized. Thresholding was performed using Otsu’s method and then the image was converted to a binary image. Then denoising was performed by removing areas of white that had an area smaller than 250 pixels (Figure 2).

2. Segmenting the cytoplasm-
   Segmentation was performed by creating a square mask using the minimum and maximum coordinate values from the segmented nucleus to act as the active contour initialization. The mask was resized by 5 pixels in order to begin the active contour inside of the object. Saturation values from the image's HSV space were smoothed with anisotropic diffusion followed by histogram equalization. This combination of image processing best improved the cytoplasm segmentation. The image was sharpened with edge detection by the Sobel method. Active contouring was performed with the Chan-Vese method, which worked best with the mask inside of the object, at 100 iterations. Contouring occasionally created multiple objects that were outside of the target, so any extra parts were removed by isolating the largest connected component (Figure 3).

3. Performing feature extraction
   15 features were extracted from the images, which are described below. Values in parentheses are the indices that reference the features.

   Texture (Cytoplasm: (12)Contrast, (13)Correlation, (14)Energy, (15)Homogeneity)
   Four features were produced from texture- contrast, correlation, energy, and homogeneity.
   The cytoplasm area was extracted from the grayscale image and then the size was reduced to rectangular area around the edges of the cytoplasm. The number of grey levels were reduced to 64 and then using a distance of one, four co-occurrence matrices were calculated at different angles and then averaged.

   Perimeter ((1)Nucleus and (2)Cytoplasm)
   We found the minimum and maximum x and y values of the part of the cell being looked at and then created a square mask. The perimeter of the square was the perimeter used.

   Area ((3)Nucleus and (4)Cytoplasm)
   The area was calculated by finding the number of nonzero pixels in the image.

   Roundness ((5)Nucleus and (6)Cytoplasm)
   Roundness was calculated by:

$$roundness = \frac{perimeter^2}{4 * \pi * area}$$

Ratio (Nucleus and Cytoplasm: (7)Single value)

Ratio was calculated by:

$$ratio\ of\ cell = \frac{area\ of\ nucleus}{area\ of\ cytoplasm}$$

Number of parts ((8)Nucleus)

To calculate the number of parts of the nucleus, a binary image was input into a function that calculated the number of objects in the image.

Average color (Cytoplasm: (9)Red, (10)Blue, (11)Green Values)

To calculate the average color, the RGB values for each of the pixels of the segmented cytoplasms were extracted, added together, and divided by the total. When working with Matlab, it is important to note that it was necessary to change the data type from uint8 to uint32, due to the initial values being above the threshold. After processing the information and calculating for the average RGB values, we converted it back to uint8. Conducting this step prevented the loss of data.

4. Applying a Machine Learning Algorithm

In order to choose the best machine learning algorithm, we started with a simple representation being decision trees. Each of the features extracted as mentioned in section 3 were fitted to a binary classification decision tree with the proper decision labels for the cell types. The test truth was determined by our interpretation of the literature provided. Since the model had a 100% accuracy on the test examples (Table 1), it was decided that a more complicated machine learning representation was not necessary.

The image labels were converted from lymphocyte, monocyte, and neutrophil to 0, 1, and 2 respectively. x3 is the area of the nucleus, x5 is the roundness of the nucleus, and x6 is the roundness of the cytoplasm (Figure 1).

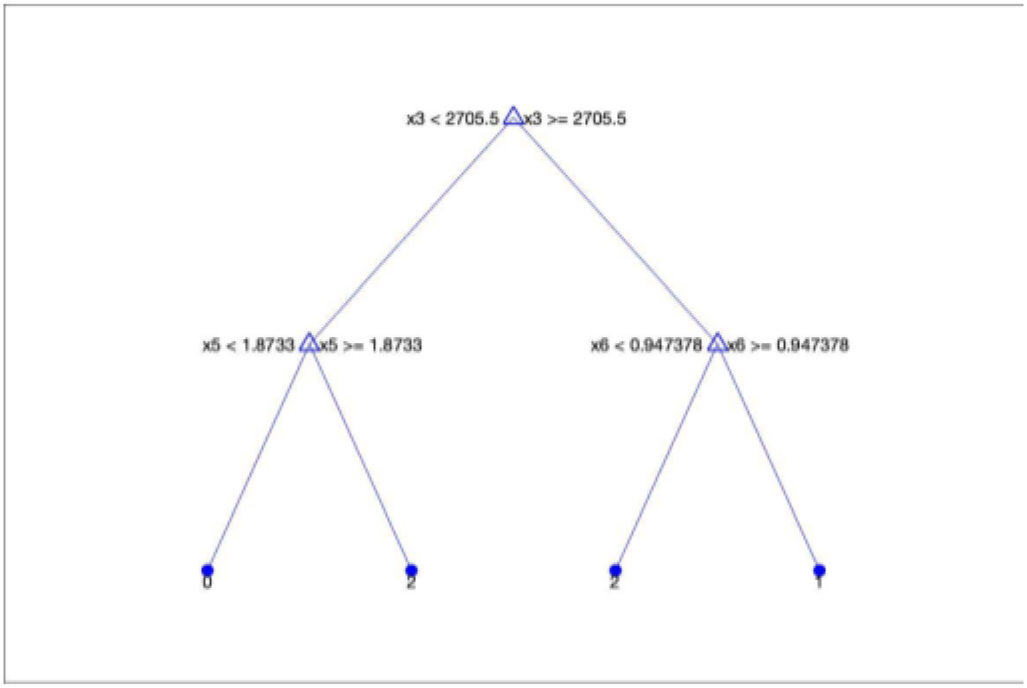

Figure 4. Resulting decision tree with the cell labels as leaves (0: Lymphocyte, 1: Monocyte, 2: Neutrophil). Internal nodes are the features used.

Image Label	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
Test Label	2	2	0	2	1	0	1	1	0	1
Label Meaning	Neut	Neut	Lymp	Neut	Mono	Lymp	Mono	Mono	Lymp	Mono
Truth	Neut	Neut	Lymp	Neut	Mono	Lymp	Mono	Mono	Lymp	Mono

Table 1. Predictions made by the fitted binary decision tree. The first row is the image label given in the test set. The second row is the raw label given by the tree. Third row is the translation of the label values (0: Lymp/Lymphocyte, 1: Mono/Monocyte, 2: Neut/Neutrophil). Fourth row is the truth based on our interpretation of the given literature.

Conclusion

After segmenting the nucleus and cytoplasm using the 20 original images provided of each type of cell, 15 features, that were mentioned above, were extracted and fed into a machine learning classifier. In this case, a decision tree was used as the simplest model with the best results. Out of the 15 features, 3 were used by the binary decision tree, and running this with the 10 test images, it was found to have 100% accuracy in identifying lymphocytes, monocytes, and neutrophils.

Appendix

(see attached for code)

References

[1] S. H. Rezatofighi and H. Soltanian-Zadeh, “Automatic recognition of five types of white blood cells in peripheral blood,” Computerized Medical Imaging and Graphics, vol. 35, no. 4, pp. 333–343, 2011.

[2] Cao, Y 2019, Operations in Intensity Space, lecture notes, Medical Image Analysis Math 6346, University of Texas at Dallas, delivered 19 August 2019.

[3] Cao, Y 2019, Histograms, Linear Filtering, Convolution, lecture notes, Medical Image Analysis Math 6346, University of Texas at Dallas, delivered 21 August 2019.

[4] Cao, Y 2019, Image Smoothing and Anisotropic Diffusion, Medical Image Analysis Math 6346, University of Texas at Dallas, delivered 26 August 2019.

[5] Cao, Y 2019, Segmentation and Thresholding, Medical Image Analysis Math 6346, University of Texas at Dallas, delivered 28 August 2019.

[6] Cao, Y 2019, Thresholding, Segmentation as Clustering, Minimization, k-means, and Ostu’s Method, Medical Image Analysis Math 6346, University of Texas at Dallas, delivered 4 September 2019.

[7] Cao, Y 2019, Fuzzy C-means and Gaussian Mixture Models, Medical Image Analysis Math 6346, University of Texas at Dallas, delivered 9 September 2019.

[8] Cao, Y 2019, Gaussian Mixture Model and the EM Algorithm, Medical Image Analysis Math 6346, University of Texas at Dallas, delivered 11 September 2019.

[9] Cao, Y 2019, Region Based Active Contour and Active Contour Without Edges, Medical Image Analysis Math 6346, University of Texas at Dallas, delivered 23 October 2019.

[10] Cao, Y 2019, Region Based Active Contours and Multilayer Neural Networks, Medical Image Analysis Math 6346, University of Texas at Dallas, delivered 28 October 2019.

[11] Cao, Y 2019, Final Project, Medical Image Analysis Math 6346, University of Texas at Dallas, delivered 9 October 2019.

Contributions

Nasheha- Wrote the average color function, and the script that loads the test data. Drafted an initial report, wrote the average color section, and the conclusion.

Arianna- Wrote segmentation of nucleus function, function to create feature matrix, and all feature functions except average color. On the report, I wrote the introduction and sections about the code I wrote.

Kevin- Wrote scripts to load training data for processing, function for cytoplasm segmentation, and classification tree fitting. The same sections were handled on the report.