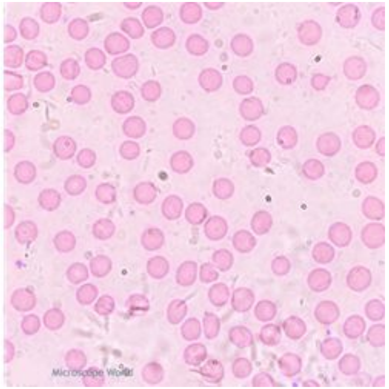
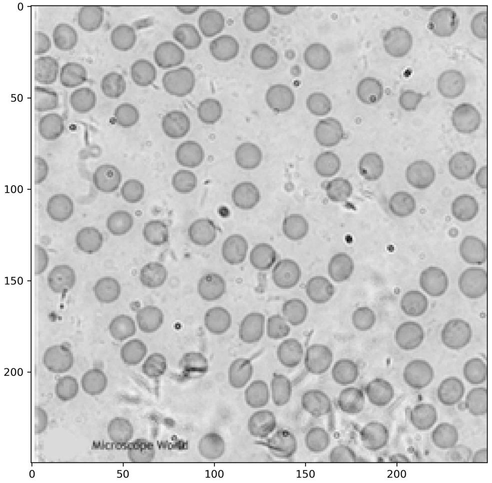
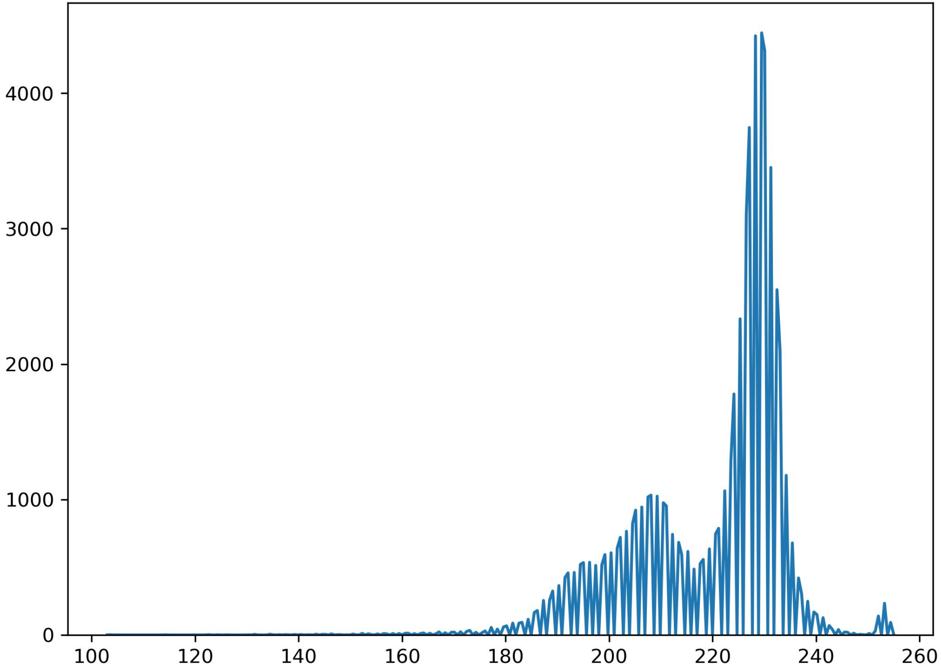
Activity 10 - Blob Analysis

de Castro, Crizzia Mielle | 2015-08076

Histogram of Grayscale Image

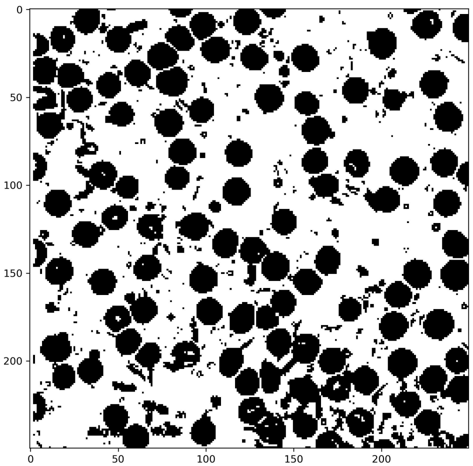


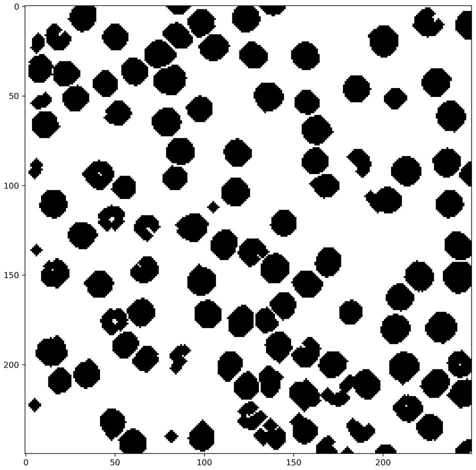




From the histogram of the image, a viable value for thresholding is at around 220. Thus, to obtain the binary image of the original cells, I set all values greater than 220 to be 0, and everything else to 255. Thus, the binary image looks like white blobs in a black background. I also did the opposite: black blobs in a white background. These binary images still need to be cleaned since there is a lot of background noise.

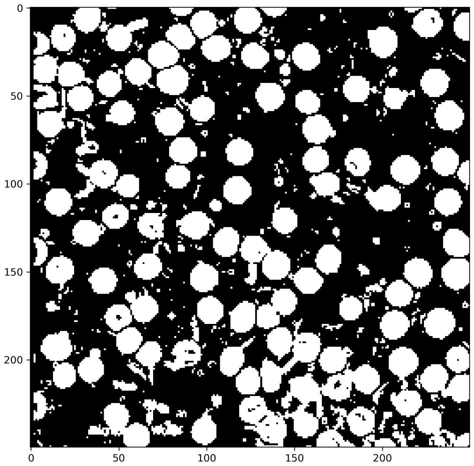
Binary Image Cleaning using Morphological Operations

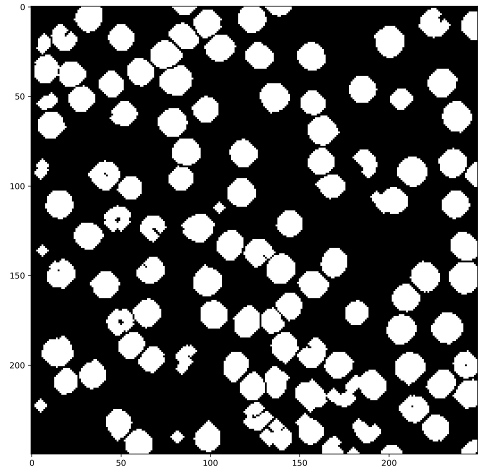




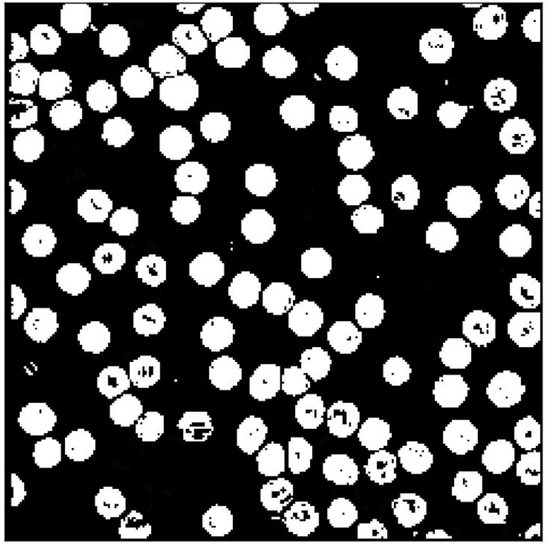
I cleaned the black blobs over a white background using morphological operations. I used OpenCV’s *closing* and a 3x3 cross as my structuring element. Closing is dilation followed by erosion. I did this for three iterations.

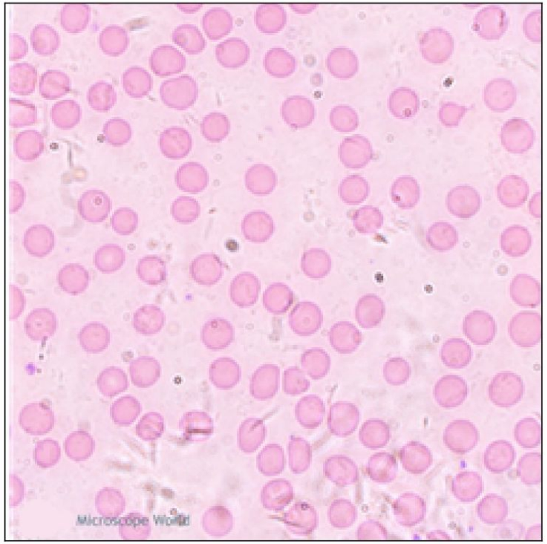
I also cleaned the white blobs over a black background using morphological operations. I used OpenCV’s *opening,* and a 3x3 cross as my structuring element. Opening is erosion followed by dilation. I also did this for three iterations.

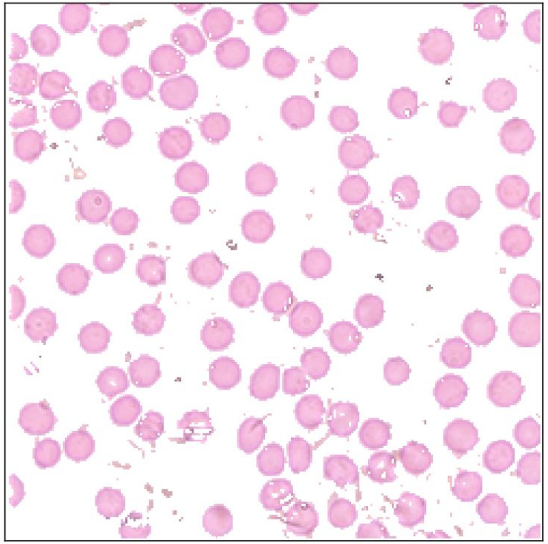




Binary Image Cleaning using Nonparametric Segmentation







I also attempted nonparametric segmentation to clean the original image. I used one of the cells as my patch. This was suggested by one of my classmates, Paolo Mawis. We found out that the best number of bins to use to obtain a clean image is 30.

Comparing the three cleaned images, some of the cells are erased when using morphological operations. This is evident for the “incomplete,” cells at the sides. Some of the blobs in the nonparametric segmentation remain connected even after cleaning. This could lead to confusion later on.

Parameters for OpenCV’s Blob Detector

blobColor = 255 (for white on black)
blobColor = 0 (for black on white)

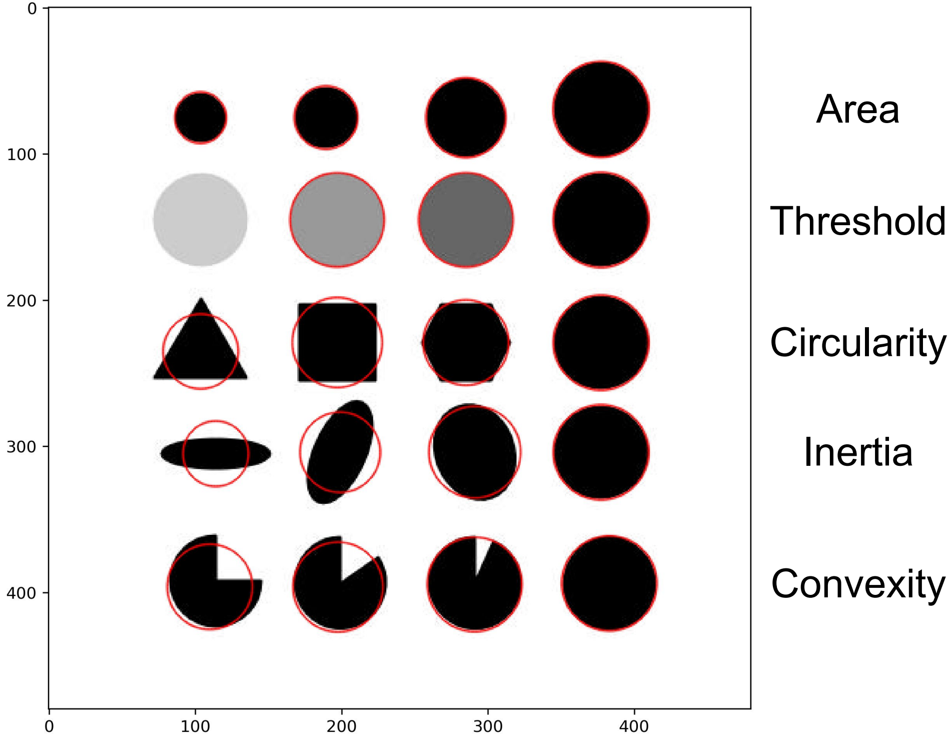
minArea = 50
(to prevent detecting connected cells)
maxArea = 250 (white on black)
maxArea = 300 (black on white)

minCircularity = 0.4

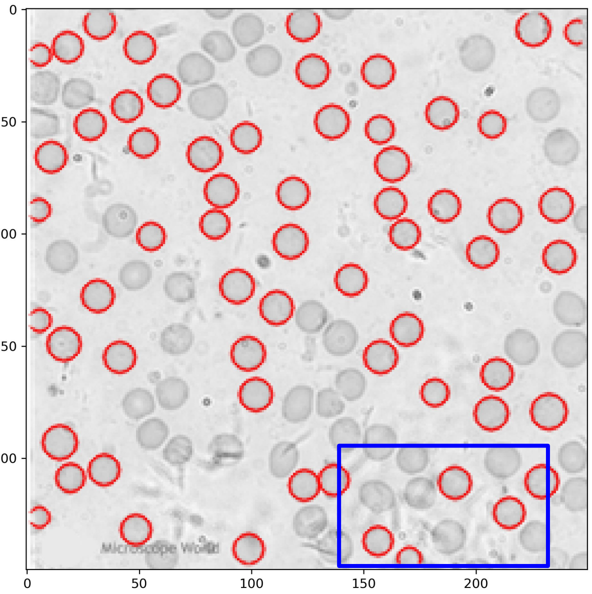
minConvexity = 0.0001

minInertiaRatio = 0.1

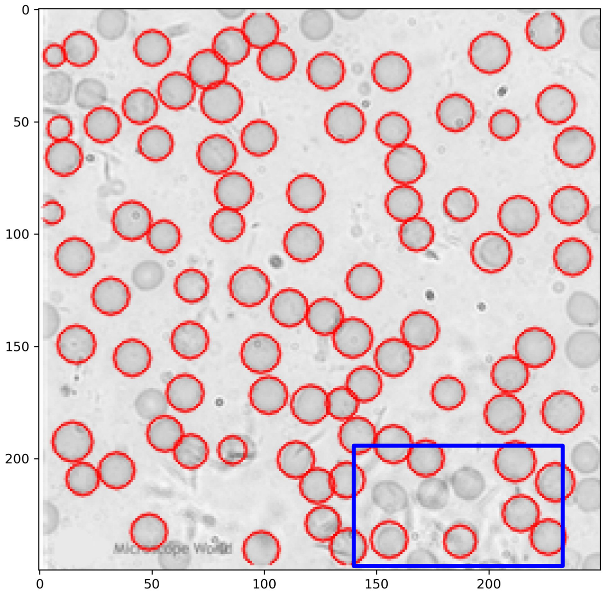
I applied these parameters (except for area) to a trial image. It is labeled on the right how each parameter affects how each blob shape and color is detected. blobColor=0 detects dark blobs, while blobColor=255 detects light blobs. I didn’t use the threshold parameter, since my images are already binary.



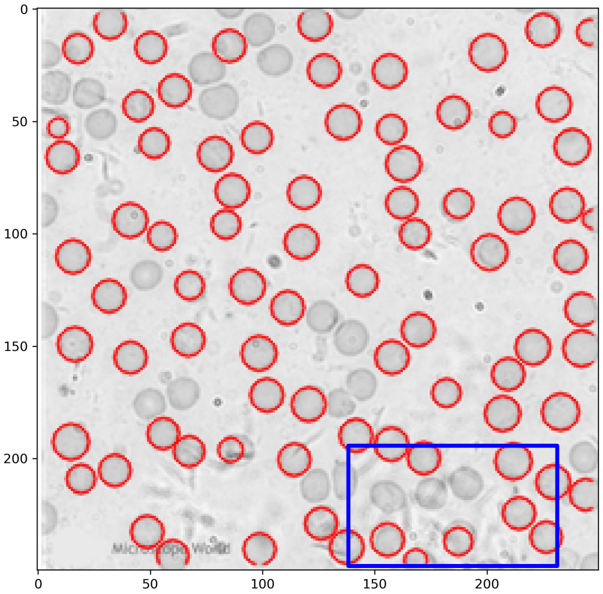
Blob Detection



nonparametric segmentation
64 blobs



black on white
87 blobs



white on black
82 blobs

The binary image with black blobs on a white detected more blobs than the other two. A similar problem among the three is that they have difficulty detecting cells in the noisy area at the bottom right (blue rectangle).

Mean Diameter, Mean Area, Mean Perimeter

	Mean Diameter	Mean Area	Mean Perimeter
Nonparametric segmentation	13.55 ± 1.63	146.32 ± 33.55	42.57 ± 5.12
Black on white	15.88 ± 1.59	199.97 ± 35.96	49.88 ± 4.98
White on black	14.16 ± 1.53	159.41 ± 31.17	44.50 ± 4.81

OpenCV's blob detector outputs the labels for each blob. From these labels, the diameter of each blob can be obtained by using the *size* function (*keyword.size*). The mean diameter, area, and perimeter of each blob can be calculated from this.