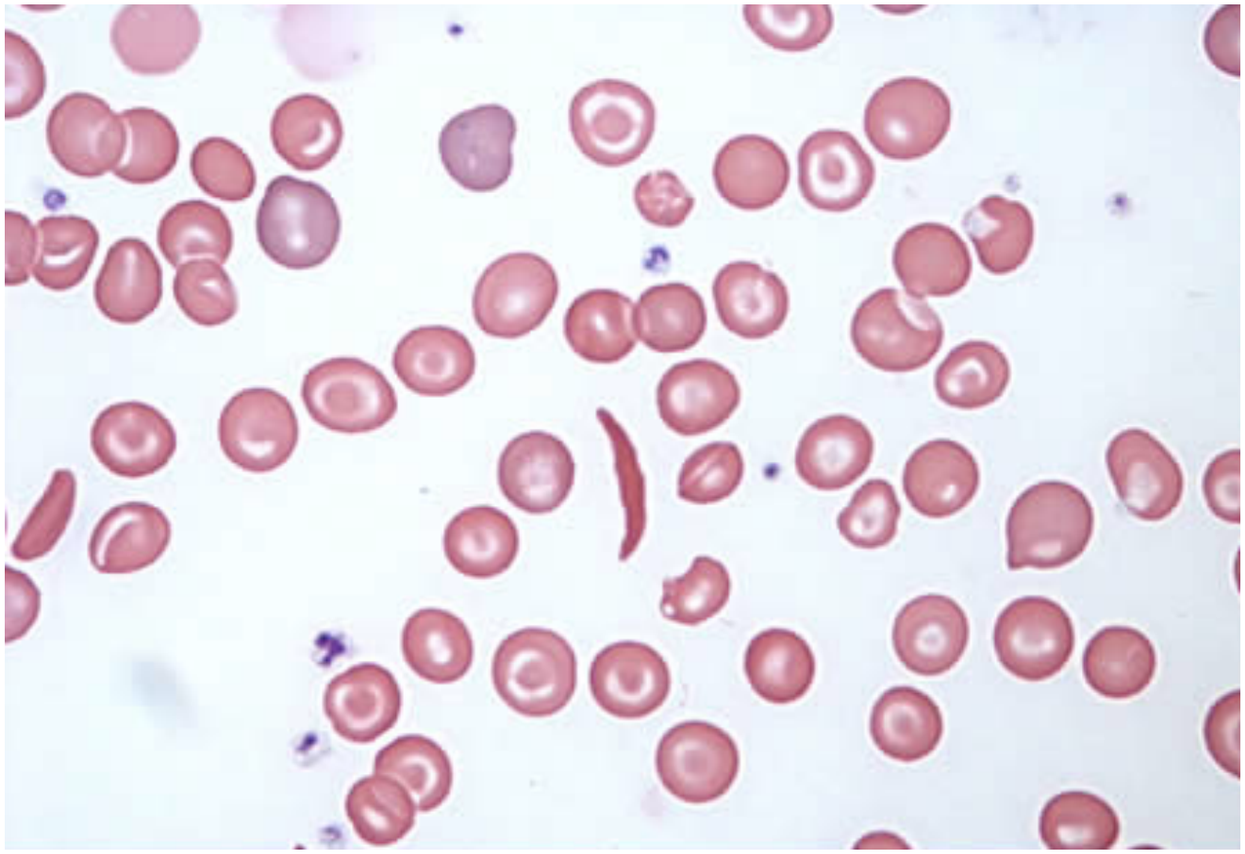
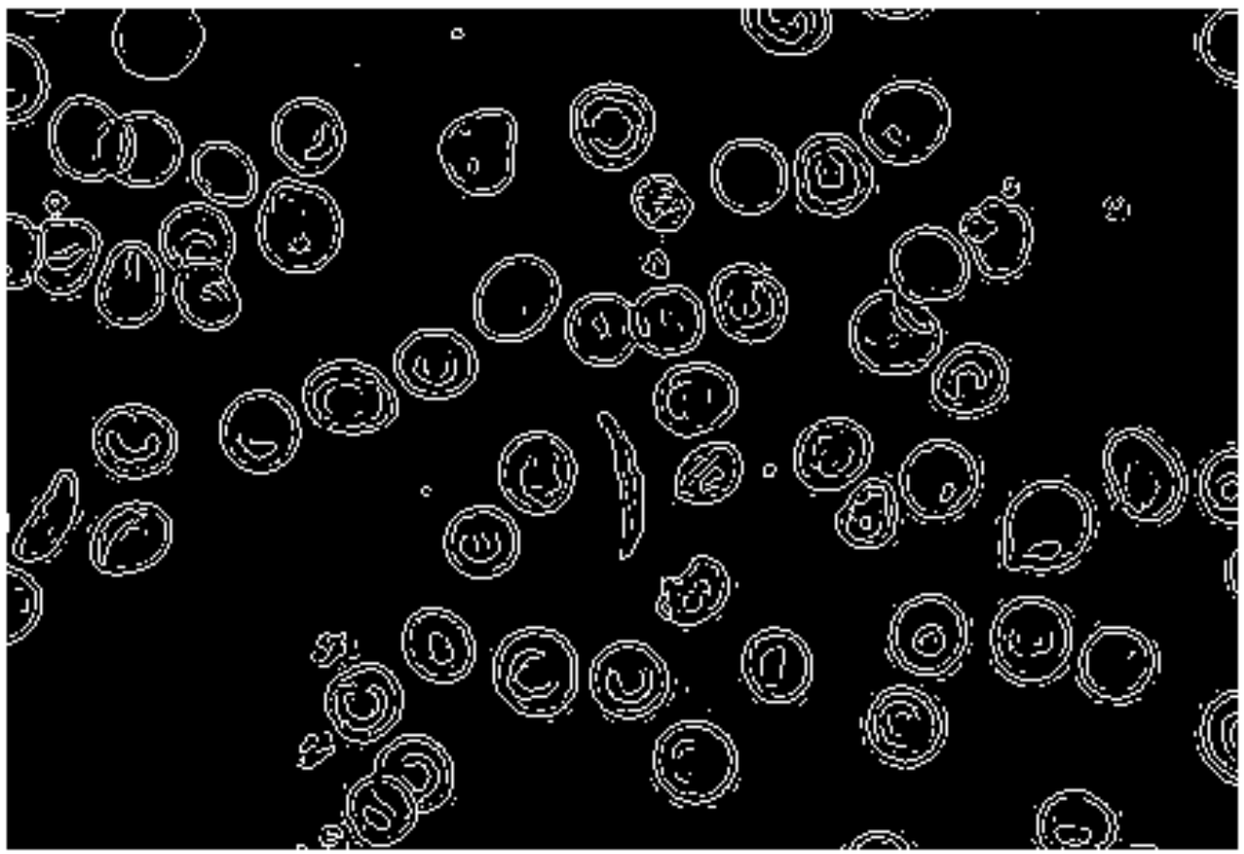
# Computer Vision for Blood Disease Detection



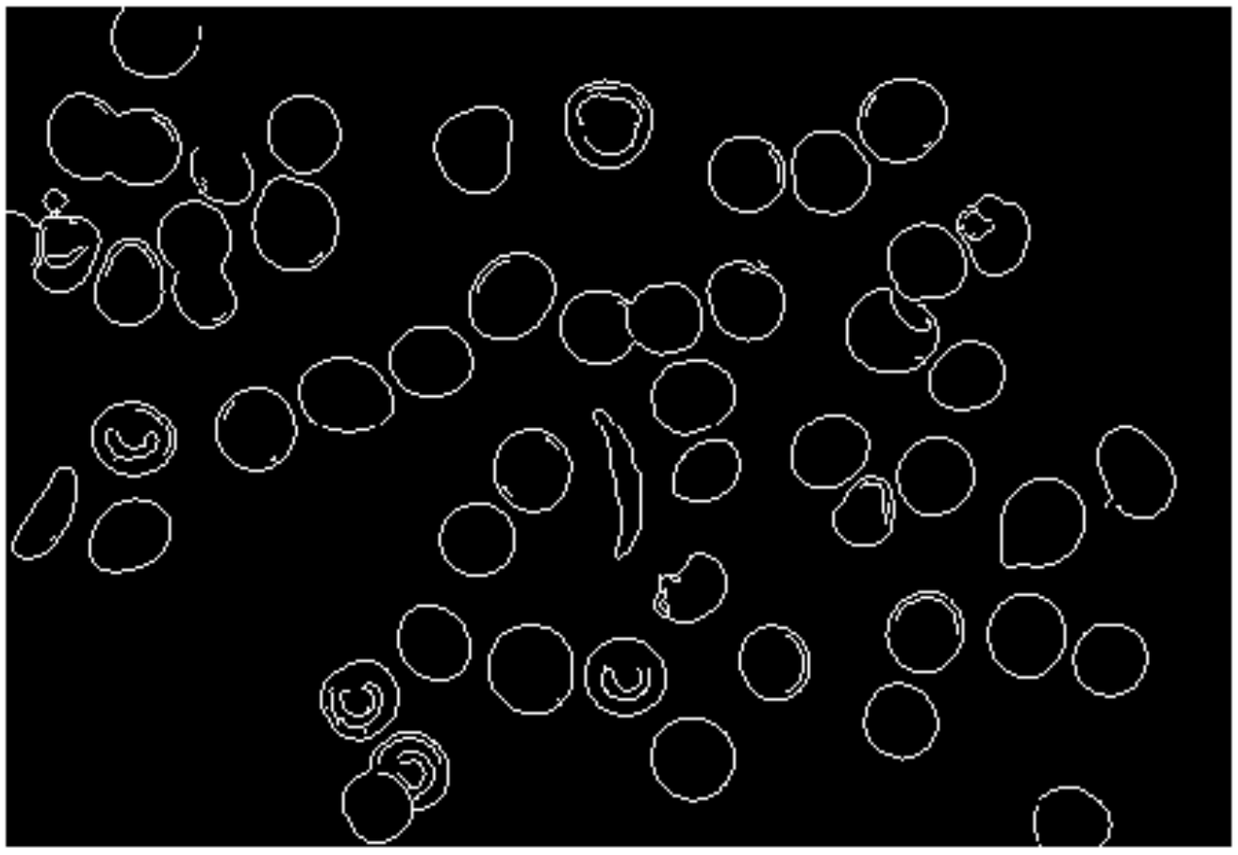
In my current science project, I am trying to detect diseases such as sickle-cell anemia in patient blood smears, by analyzing the shape area and perimeter of red blood cells (RBC). An example blood smear is the image to the left. At the center is a sickle-cell, and to the left of the center is an elongated cell (not quite a sickle-cell, but not a circular healthy RBC). My algorithm identifies both of these irregular cells, and counts all of the RBC present in the given blood smear image.

# Step 1: Edge Detection Filter



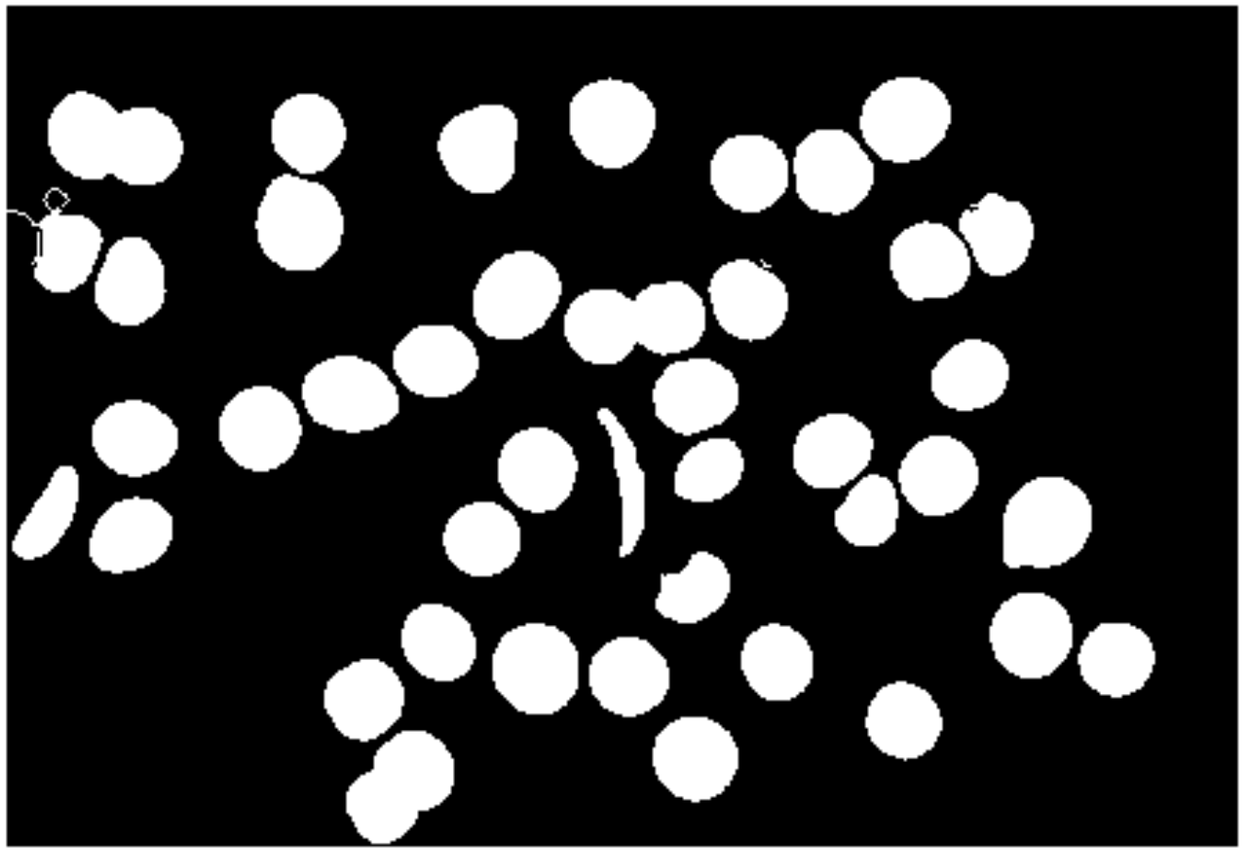
The first step of the algorithm is to use an edge detection filter. This analyzes the difference in color between each pixel and the four pixels around it. Based on the color difference between a pixel and its neighbors, a black-and-white image is produced. The whiter the markings on this image, the higher contrast in that region of the image. Thus, between RBC and blood plasma, there is a high contrast and thus very white markings. The output of this step produces this image.

# Step 2: Removing Noise



The next step is a clean up step. I first convert the gradient black-and-white image from the previous step into a binary file (pure black or pure white) using an absolute threshold. The previous image had many small lines inside and outside of cells. This unwanted noise would throw the algorithm off, so I remove them by removing any 'chains' of white pixels in the image that are shorter than 10 pixels. The result is this image.

# Step 3: Smart Filling



In this step, I take the end product of the last step and smartly fill in all of the cell shapes. I also clear the borders because if the algorithm were to process such partial cells, it could be thrown off.

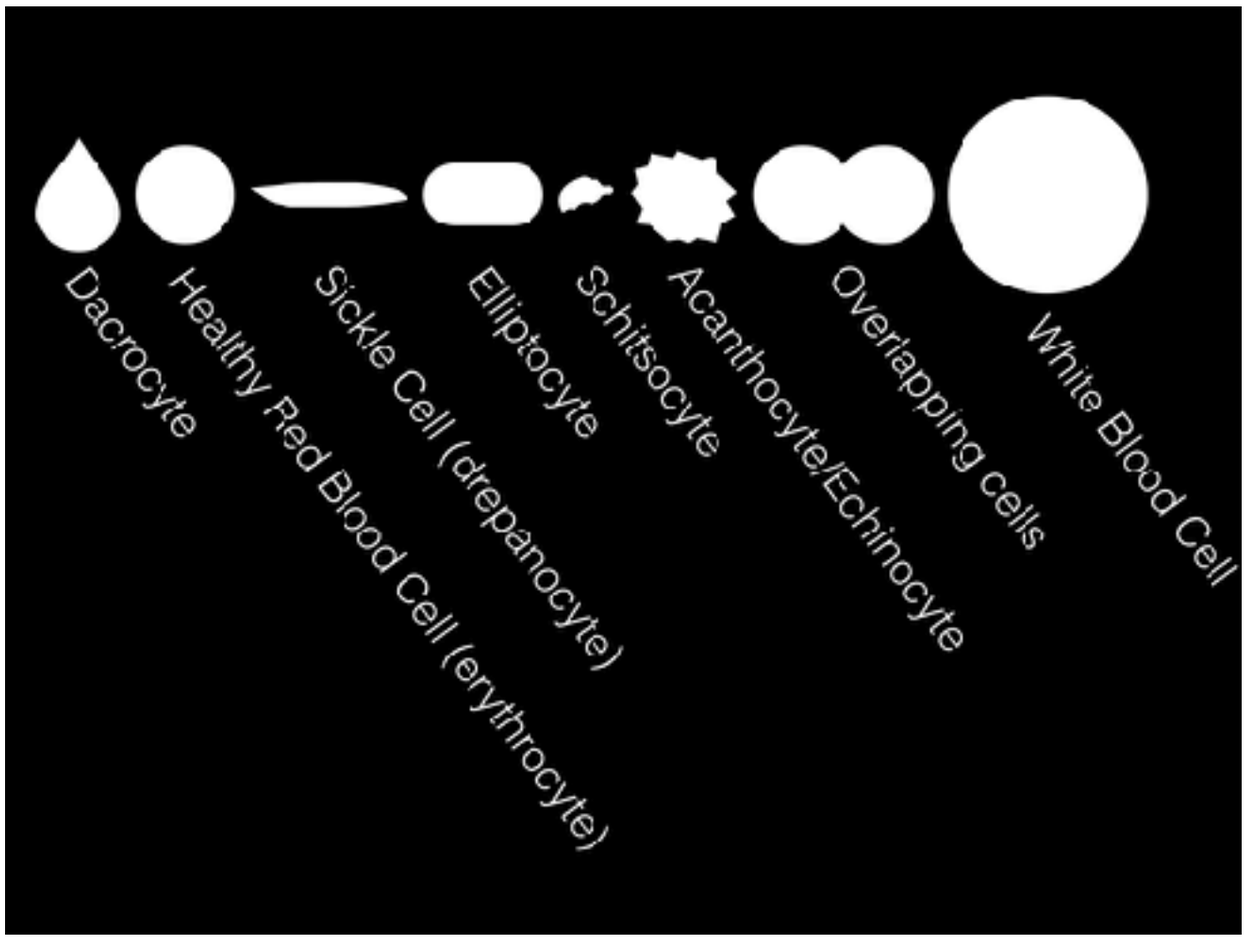
# Step 4: Calculating Shape Factor, Relative Area, and Perimeter to Classify each Cell

Finally, for each cell I calculate the area and perimeter and store these in a matrix. I also calculate the total sum area of all the pixels in the image and divide it by the total number of cells, to find the average cell area. I divide each individual cell's area by this average, and the result is a number indicating each cell's size relative size. I do this because depending on the tools used to capture the smears, different images could have different resolutions, so by processing area on a relative scale I overcome this complication. I then calculate the shape factor for each cell based on perimeter and relative area using this formula:

shape factor = 4*pi*Relative Area/Perimeter^2

For a perfect circle, this yields a result of 1, and for ellipses it yields around 0.7. For extremely flat objects like sickle cells, it yields values as low as 0.4. Then, based on area and shape factor, I can classify the cells. Here are an identified sickle-cell and elliptocyte.

# Classification Explanation




Here is an outline of the cell types I'm currently trying to classify with the algorithm, along with rough examples of how each cell type is shaped. Besides the healthy RBC and White Blood Cells (WBC), the presence of the other types is an indicator for several diseases. Also, an over-presence of WBC indicates a form of leukemia. Note that one of the groups I'm classifying is 'overlapping cells'. This is because sometimes in blood smears multiple cells can overlap. To not get any false-positives, I need to specially look out for this case.

Each of these types has a different shape factor and relative area. By dividing both these factors into zones (ex: for relative area the zones are <0.5, 0.5-0.6, 0.6-0.95, 0.95-1.05, >1.05), and checking which two zones each cell falls in, I can classify the cell into a group, which I store in several arrays and return as output. This whole process takes around 3.4 seconds, at virtually no cost. I strongly believe this is the future of medicine and healthcare.