Activity 10: Blob Analysis

Mary Chris Go / 2014 11122

Goal

To use blob analysis to identify the number of cells in an image and to identify its properties.

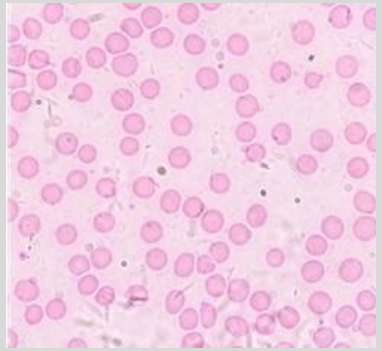
Mary Chris Go / 2014 11122

Blob Analysis

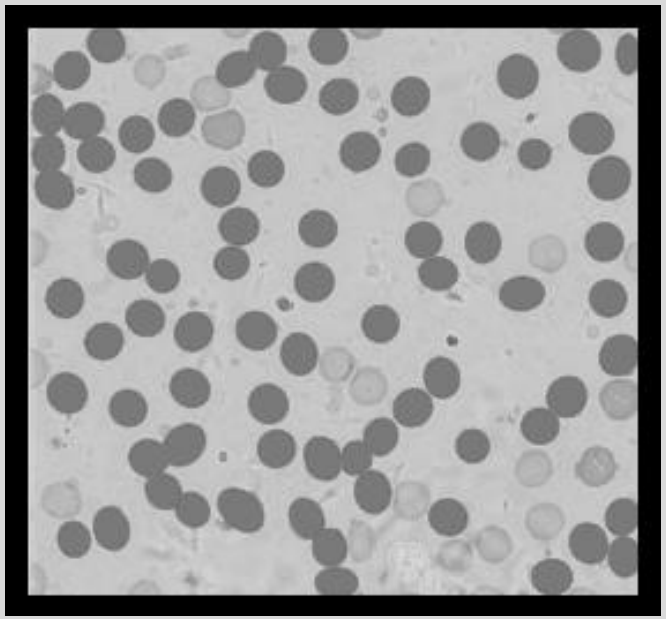
A fundamental technique of machine video that is being based from the consistent shapes is called blob analysis. A tool for object detection that is being used for many application.

Detection of Cells.

For this activity, we were asked to identify the cells by using various image processing techniques. Apart from it, the cell size, major axis length, and perimeter were also identified.



Trial #1



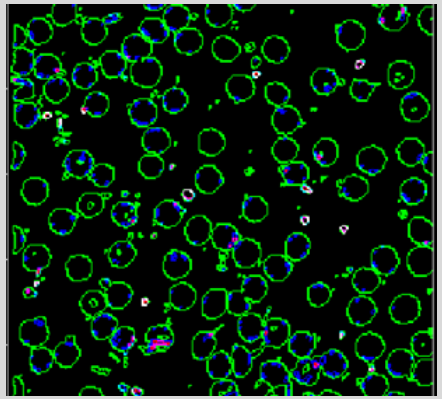
I got frustrated. Therefore, I tried the first technique. This doesn't involve old packages. But compared to trial #2, this is efficient.

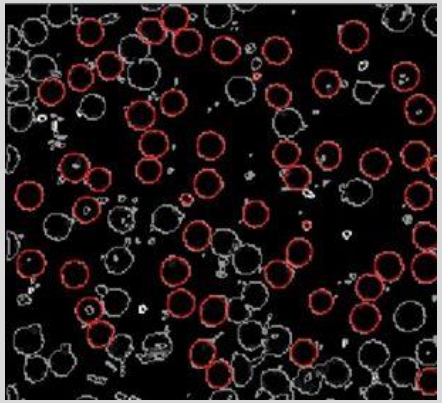
No more SimpleCreateBlob() but instead, I treated the image first using Gaussian Blur, Otsu's method, a bit of bias because of some trial and error for the values to assign. I detected 145 cells from the image as you can see from the image. Although, I had a hard time extracting its properties. I had to switch my method to achieve the goal.

```
A = cv.imread('3468final.jpg', 0)
fig = plt.figure()
ax = fig.add_subplot(111)
Gb = cv.GaussianBlur(A, (1, 1), 0)
thresh, Gb = cv.threshold(A, 127, 255, cv.THRESH_OTSU)
Gb = (Gb < thresh).astype('uint8') * 255
Gb = cv.morphologyEx(Gb, cv.MORPH_OPEN,
                     cv.getStructuringElement(cv.MORPH_ELLIPSE, (3, 3)))
```

Mary Chris Go / 2014 11122

Trial #2





You have to identify the right image processing techniques you need to use to be able to 'clean' the image. Some cells are attached to each other that's why you need to find the right techniques for the 'blob analysis' to work.

For the second trial, I simple use the cv2 module and use SimpleBlobDetector_create(). I treated the image using morphological operations such as dilation and thresholding. Thanks to cv2 module for a convenient coding.

Unfortunately, only 55 cells were detected in a 134 365 area of an image.

```
struct = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,(3,3))
img = cv2.imread('3468edited.png',0)
dilation = cv2.dilate(img, struct,iterations = 2)
if cv2.__version__.startswith('2.'):
    detector = cv2.SimpleBlobDetector()
else:
    detector = cv2.SimpleBlobDetector_create()
keypoints = detector.detect(img)
print(len(keypoints))
imgKeyPoints = cv2.drawKeypoints(img, keypoints, np.array([]), (0,0,255), cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
cv2.imshow("Keypoints", imgKeyPoints)
cv2.waitKey(0)
```

But looking at the brighter side, this kind of method made it easy to know the perimeter, area, centroid, etc. Here, we assume the blobs to be circle.

Mary Chris Go / 2014 11122

Trial #2

Mary Chris Go / 2014 11122

```
size = []
for keypoint in keypoints:
    x = keypoint.pt[0]
    y = keypoint.pt[1]
    s = keypoint.size
    dia = (np.pi)*((s/2)**2)
    size.append(dia)
```

Now, we test the major axis length. By printing ‘s’ from the snippet code, we can identify the diameter of the object. From the diameter, we can also identify the perimeter by multiplying it to pi. On the other hand, average cell size is 381.3181 + 100.118. This was obtained using the code above.

The values are the major axis length of the blobs. These were then treated to get the perimeter and average are using formulas.

```
21.016799926757812
22.120454788208008
22.25127601623535
22.418947219848633
22.08047103881836
24.741146087646484
21.915420532226562
24.270442962646484
19.614601135253906
23.269269943237305
22.25347328186035
23.24005699157715
23.785924911499023
24.130123138427734
25.155912399291992
24.09615707397461
23.702585220336914
22.44837188720703
23.56025505065918
```

```
22.44837188720703
23.56025505065918
21.863136291503906
21.899398803710938
9.181890487670898
22.19458770751953
22.810678482055664
23.321094512939453
19.704301834106445
21.487136840820312
20.7227783203125
13.100981712341309
24.24188804626465
24.308515548706055
23.091218948364258
21.605867385864258
22.909725189208984
```

Now we look for the centroid of the 55 cells. Through this snippet of code, we can identify the x and y which will tell the centroid. Here are some values:

```
373.08465576171875 327.43243408203125
190.19212341308594 321.7047424316406
264.06292724609375 302.58502197265625
175.7670440673828 303.4995422363281
86.80374145507812 286.3838195800781
317.0697326660156 272.393310546875
77.42631530761719 265.7113952636719
185.68992614746094 266.2162780761719
279.0753479003906 258.267333984375
157.1007843017578 259.889404296875
321.0706787109375 245.7313690185547
241.822509765625 234.2508087158203
64.69883728027344 234.5166473388672
151.91583251953125 231.89173889160156
370.30279541015625 228.36048889160156
337.69384765625 227.6009521484375
26.512065887451172 225.2443389892578
259.8764953613281 215.4974365234375
171.2710723876953 200.89976501464844
222.0518341064453 181.8561248779297
75.21893310546875 178.052490234375
162.0299530029297 169.1226348876953
363.3277893066406 166.0602569580078
25.362899780273438 165.9829864501953
180.37075805664062 155.79454040527344
85.61483764648438 151.9972686767578
258.03045654296875 150.50308227539062
129.32484436035156 144.11326599121094
377.26513671875 141.0283203125
64.41864013671875 141.5225830078125
326.11279296875 138.29116821289062
361.26617431640625 131.07398986816406
248.84339904785156 129.52059936523438
182.7545166015625 122.60248565673828
```

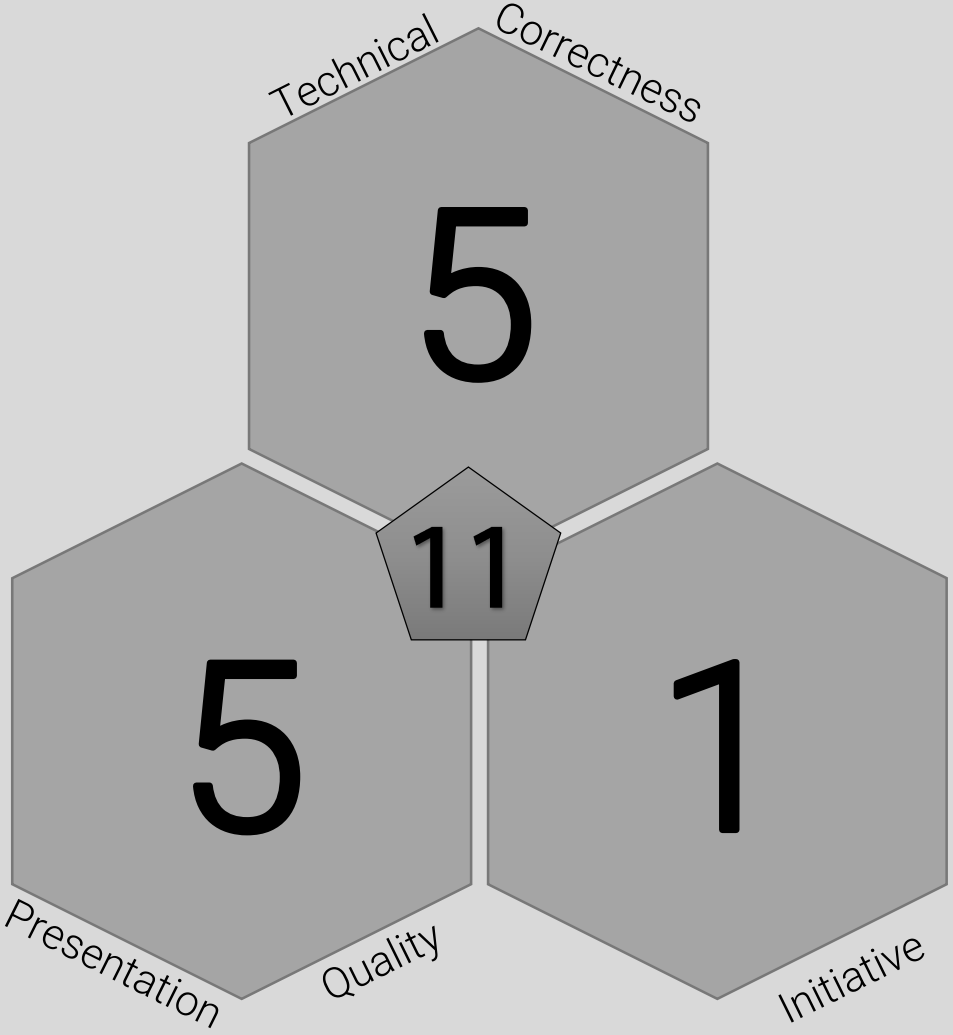
Summary

The activity was relevant even though the foundations lie deeply to other topics. Though the rising of neural network is a good idea for detection also (hehe). All in all, it was hard to ‘clean’ the image because each cell demands a different imaging processing technique. I had to make sure that my technique is general. I found the first trial to be the efficient but I had a hard time using skimage to extract the desired properties. The second one was not efficient but I was able to produce the needed values. I believe it detected less cells because the cells were hard to differentiate knowing that most are attached to each other.

All in all, the activity was challenging. I probably won’t do it again. I still find training neural networks to be the most efficient.

Mary Chris Go / 2014 11122

Self-Evaluation



References

- Soriano, M., “Blob Analysis”. 2019
- https://www.microscopeworld.com/p-3468-microscope-resolution-explained-using-blood-cells.aspxhttps://raaromeroap186.wordpress.com/2016/12/04/a9-playing-notes-by-image-processing/
- https://www.image-line.com/support/flstudio_online_manual/html/glossary_envelope.htm