

Report

Task 1.

At first, it reads image.

```
7 #read the c2 image.
8 img = cv2.imread('c2.jpg',1)
9 #print(img.shape)
10 shape_1=img.shape[0]
11 shape_2=img.shape[1]
12 shape_3=img.shape[2]
13
```

Secondly, it needs to flatten the image and calculate frequency of pixels.

```
14 #flatten image
15 def get_flatten(img):
16     return img.flatten()
17 #get frequency of each pixel.
18 def get_fre(img_02):
19     #img_02=img.flatten()
20     fre=Counter(img_02)
21     whole=len(img_02)
22     for k,v in fre.items():
23         fre[k]=fre[k]/whole
24     return fre
```

Then calculate mean intensity of pixel.

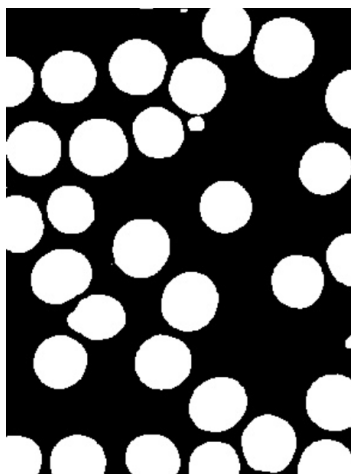
The fourth step is to implement methods to get variances. Sort the results, find the largest value so that the threshold could be found. Then divide pixels into two values, 0 and 255.

Then show image and print threshold.

```
plt.subplot(1,3,1)
plt.imshow(img)
plt.title('input img')
#OSTU method
plt.subplot(1,3,2)
plt.imshow(img_03)
plt.title('ostu method')
#HSV space method, using channel S.
plt.subplot(1,3,3)
plt.imshow(img_s)
plt.title('HSV method')
cv2.imwrite('image_task1.jpg',img_s)
```

Because of parameters in function `plt.imshow()`, result images are not like the samples in Assignment Specification, so I save one image in local file which is the same as the image in Assignment Specification. The name is 'image\_task1.jpg'.

This is 'image\_task1.jpg'.

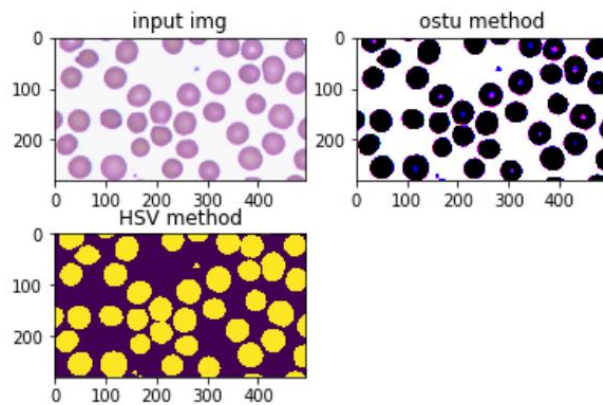


For HSV space method,I think S channel could better show image,so I use S channel in HSV space method.

This is for 'c2.jpg'.

Threshold of ostu method:206  
Threshold of hsv method:102

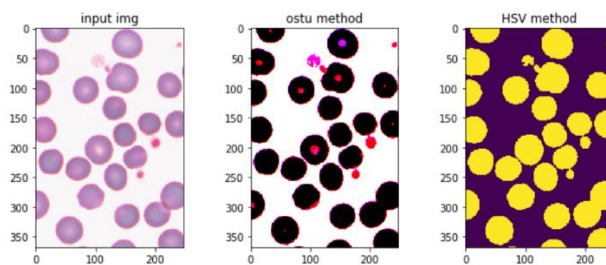
|: True



This is for 'c3.jpg'

Threshold of ostu method:206  
Threshold of hsv method:170

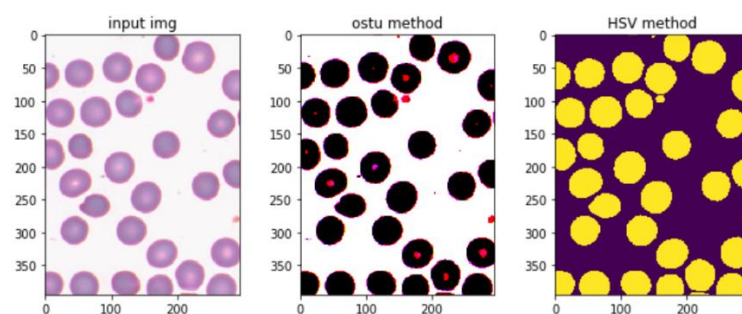
True



This is for 'c4.jpg'.

Threshold of ostu method:206  
Threshold of hsv method:110

True

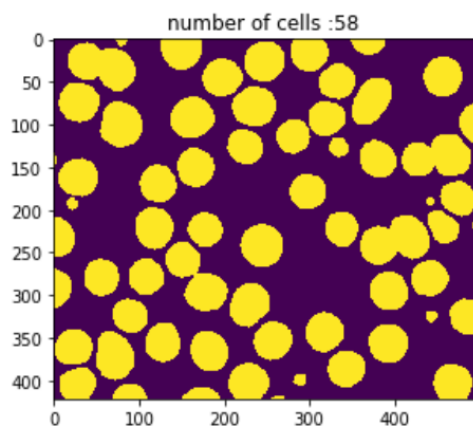


## Task 2

Firstly,I implement median filter function to remove noise from image I get in Task 1.  
Then I use two-pass connected component labelling algorithm to count number of cell in that image. min\_area and overlap\_area is 100 and 8000 seperately.

This is for 'c1.jpg.

```
Text(0.5, 1.0, 'number of cells :58')
```



This is for 'c4.jpg.

```
Text(0.5, 1.0, 'number of cells :31')
```

