

```
In [1]: import pandas as pd
import numpy as np
import cv2
from matplotlib import pyplot as plt
```

image = 3053 KI67.jpg

```
In [2]: img1=cv2.imread("3053 KI67.jpg")

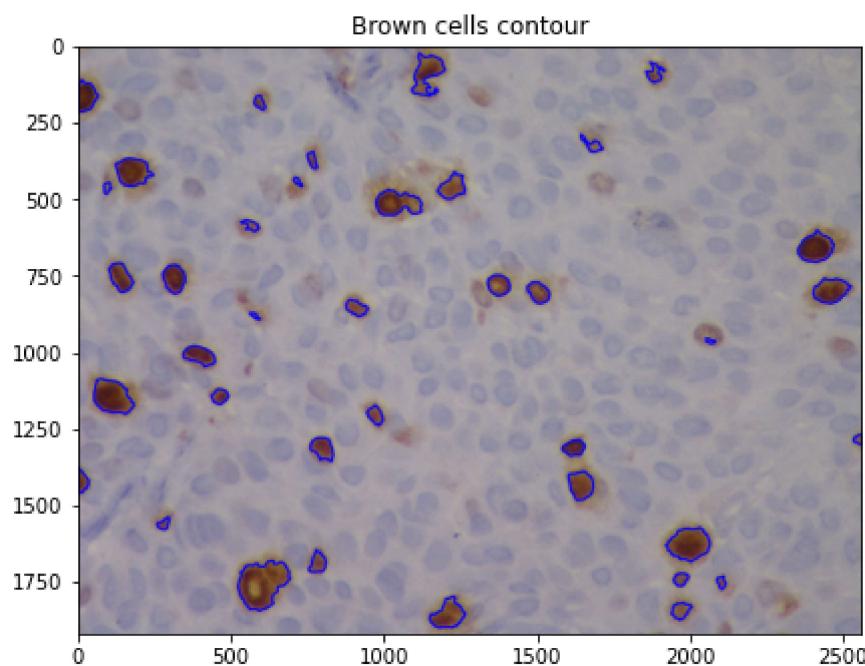
gray=cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
thresh, bin_img = cv2.threshold(gray, thresh=100, maxval=255, type=cv2.THRESH_BINARY_INV)
kernel = np.ones((3,3),np.uint8)
closed = cv2.morphologyEx(bin_img, cv2.MORPH_CLOSE, kernel=kernel, iterations=5)
contour_brown, hierarchy = cv2.findContours(image=closed.copy(), mode=cv2.RETR_EXTERNAL,
                                             contour_retrieval=cv2.CHAIN_APPROX_SIMPLE)

cells=0
for i in range(len(contour_brown)):
    area=cv2.contourArea(contour_brown[i])
    if(area>250):
        cv2.drawContours(image=img1, contours=contour_brown, contourIdx=i, color=(255,0,0))
        cells=cells+1

plt.figure(figsize=(7,7))
plt.title("Brown cells contour")
print("no of brown cells: ",cells)
img1=cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
plt.imshow(img1)
```

no of brown cells: 38

Out[2]: <matplotlib.image.AxesImage at 0x1a872cbaac0>



```
In [3]: img1=cv2.imread("3053 KI67.jpg",1)
lab= cv2.cvtColor(img1, cv2.COLOR_BGR2LAB)
```

```

l, a, b = cv2.split(lab)
clahe = cv2.createCLAHE(clipLimit=3.0, tileGridSize=(8,8))
cl = clahe.apply(l)

limg = cv2.merge((cl,a,b))
final = cv2.cvtColor(limg, cv2.COLOR_LAB2BGR)

gray=cv2.cvtColor(final, cv2.COLOR_BGR2GRAY)
thresh, bin_img = cv2.threshold(gray, thresh=100, maxval=255, type=cv2.THRESH_BINARY_INV)
kernel = np.ones((3,3),np.uint8)
closed = cv2.morphologyEx(bin_img,cv2.MORPH_CLOSE, kernel=kernel, iterations=5)
contour2, hierarchy = cv2.findContours(image=closed.copy(), mode=cv2.RETR_EXTERNAL, method=cv2.CHAIN_APPROX_SIMPLE)

tcells=0
for i in range(len(contour2)):

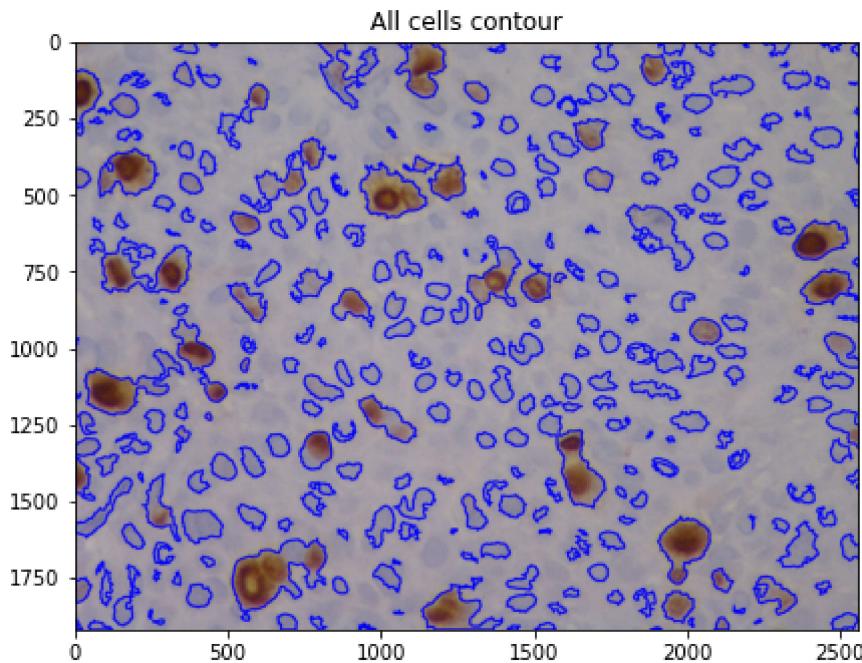
    area=cv2.contourArea(contour2[i])
    if(area>100):
        cv2.drawContours(image=img1,contours=contour2,contourIdx=i, color=(255,0,0), thickness=1)
        tcells=tcells+1

plt.figure(figsize=(7,7))
plt.title("All cells contour")
print("no of total cells: ",tcells)

img1=cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
plt.imshow(img1)

```

no of total cells: 306
Out[3]: <matplotlib.image.AxesImage at 0x1a876566490>



In [4]: percentage_positivity_3053=(cells/tcells)*100
percentage_positivity_3053

Out[4]: 12.418300653594772

```
In [5]: if(percentage_positivity_3053<15):
    print("low grade")
else:
    print("high grade")
```

low grade

image = 2141 KI67.jpg

```
In [6]: img1=cv2.imread("2141 KI67.jpg")

gray=cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
thresh, bin_img = cv2.threshold(gray, thresh=120, maxval=255, type=cv2.THRESH_BINARY_IN
kernel = np.ones((3,3),np.uint8)
closed = cv2.morphologyEx(bin_img, cv2.MORPH_CLOSE, kernel=kernel, iterations=5)
contour_brown, hierarchy = cv2.findContours(image=closed.copy(), mode=cv2.RETR_EXTERNAL

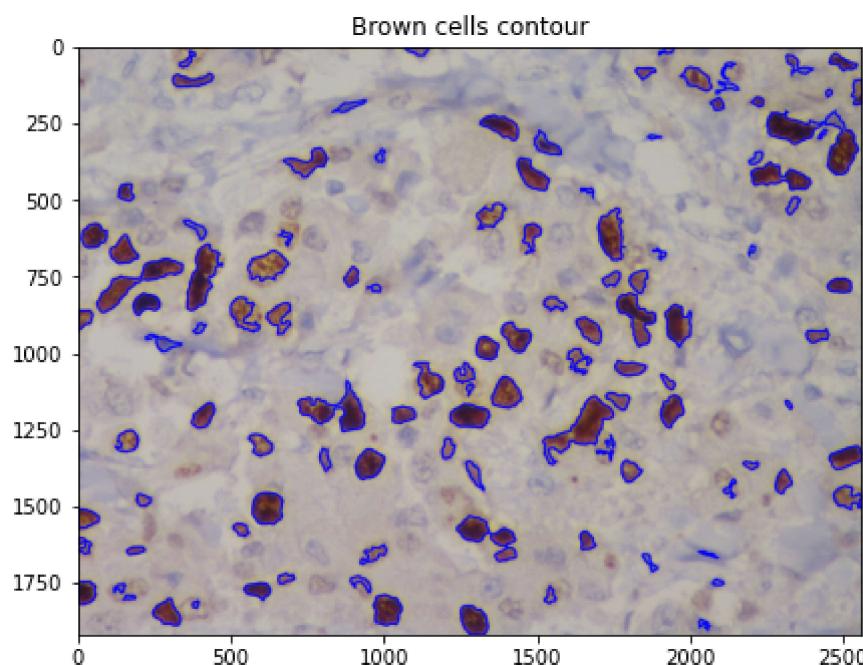
cells=0
for i in range(len(contour_brown)):

    area=cv2.contourArea(contour_brown[i])
    if(area>250):
        cv2.drawContours(image=img1,contours=contour_brown,contourIdx=i, color=(255,0,0
        cells=cells+1

plt.figure(figsize=(7,7))
plt.title("Brown cells contour")
print("no of brown cells: ",cells)
img1=cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
plt.imshow(img1)
```

no of brown cells: 112

Out[6]: <matplotlib.image.AxesImage at 0x1a873ea0b80>



```
In [7]: img1=cv2.imread("2141 KI67.jpg",1)
lab= cv2.cvtColor(img1, cv2.COLOR_BGR2LAB)
```

```

l, a, b = cv2.split(lab)
clahe = cv2.createCLAHE(clipLimit=3.0, tileGridSize=(8,8))
cl = clahe.apply(l)

limg = cv2.merge((cl,a,b))
final = cv2.cvtColor(limg, cv2.COLOR_LAB2BGR)

gray=cv2.cvtColor(final, cv2.COLOR_BGR2GRAY)
thresh, bin_img = cv2.threshold(gray, thresh=100, maxval=255, type=cv2.THRESH_BINARY_INV)
kernel = np.ones((3,3),np.uint8)
closed = cv2.morphologyEx(bin_img,cv2.MORPH_CLOSE, kernel=kernel, iterations=5)
contour2, hierarchy = cv2.findContours(image=closed.copy(), mode=cv2.RETR_EXTERNAL, method=cv2.CHAIN_APPROX_SIMPLE)

tcells=0
for i in range(len(contour2)):

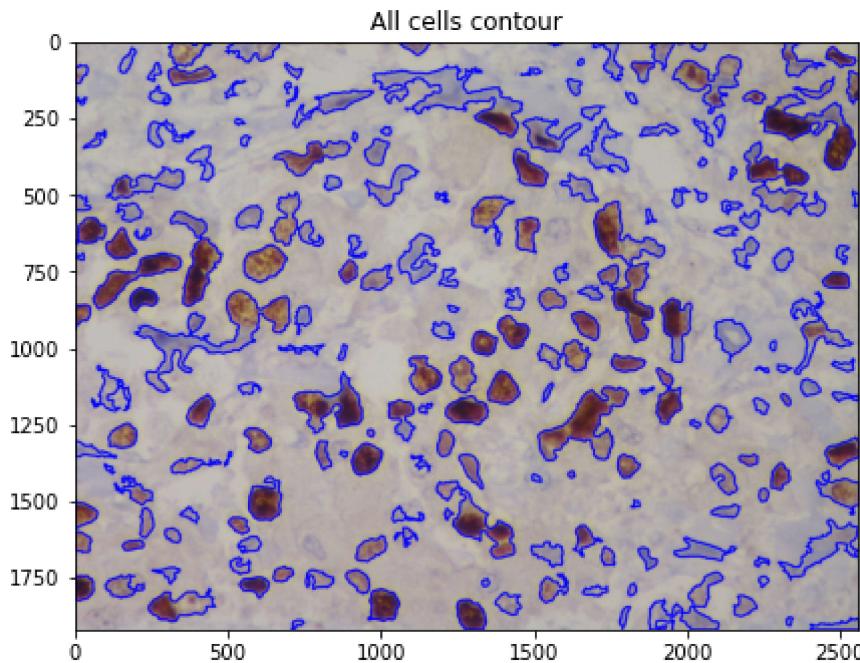
    area=cv2.contourArea(contour2[i])
    if(area>250):
        cv2.drawContours(image=img1,contours=contour2,contourIdx=i, color=(255,0,0), thickness=1)
        tcells=tcells+1

plt.figure(figsize=(7,7))
plt.title("All cells contour")
print("no of total cells: ",tcells)

img1=cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
plt.imshow(img1)

```

no of total cells: 199
Out[7]: <matplotlib.image.AxesImage at 0x1a87654b4f0>



In [8]: percentage_positivity_2141=(cells/tcells)*100
percentage_positivity_2141
Out[8]: 56.28140703517588

```
In [9]: if(percentage_positivity_2141<15):
    print("low grade")
else:
    print("high grade")
```

high grade

image = 1714 KI67.jpg

```
In [10]: img1=cv2.imread("1714 KI67.jpg")

gray=cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
thresh, bin_img = cv2.threshold(gray, thresh=125, maxval=255, type=cv2.THRESH_BINARY_IN
kernel = np.ones((3,3),np.uint8)
closed = cv2.morphologyEx(bin_img, cv2.MORPH_CLOSE, kernel=kernel, iterations=5)
contour_brown, hierarchy = cv2.findContours(image=closed.copy(), mode=cv2.RETR_EXTERNAL

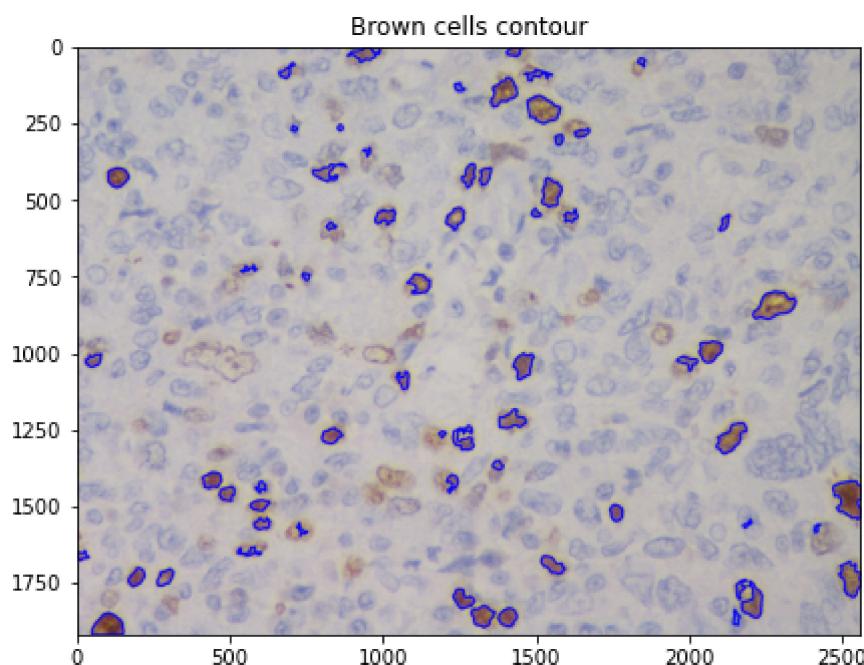
cells=0
for i in range(len(contour_brown)):

    area=cv2.contourArea(contour_brown[i])
    if(area>200):
        cv2.drawContours(image=img1,contours=contour_brown,contourIdx=i, color=(255,0,0
        cells=cells+1

plt.figure(figsize=(7,7))
plt.title("Brown cells contour")
print("no of brown cells: ",cells)
img1=cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
plt.imshow(img1)
```

no of brown cells: 63

Out[10]: <matplotlib.image.AxesImage at 0x1a8009c7f40>



```
In [11]: img1=cv2.imread("1714 KI67.jpg",1)
lab= cv2.cvtColor(img1, cv2.COLOR_BGR2LAB)
```

```

l, a, b = cv2.split(lab)
clahe = cv2.createCLAHE(clipLimit=3.0, tileGridSize=(8,8))
cl = clahe.apply(l)

limg = cv2.merge((cl,a,b))
final = cv2.cvtColor(limg, cv2.COLOR_LAB2BGR)

gray=cv2.cvtColor(final, cv2.COLOR_BGR2GRAY)
thresh, bin_img = cv2.threshold(gray, thresh=100, maxval=255, type=cv2.THRESH_BINARY_INVERSE)
kernel = np.ones((3,3),np.uint8)
closed = cv2.morphologyEx(bin_img,cv2.MORPH_CLOSE, kernel=kernel, iterations=5)
contour2, hierarchy = cv2.findContours(image=closed.copy(), mode=cv2.RETR_EXTERNAL, method=cv2.CHAIN_APPROX_SIMPLE)

tcells=0
for i in range(len(contour2)):

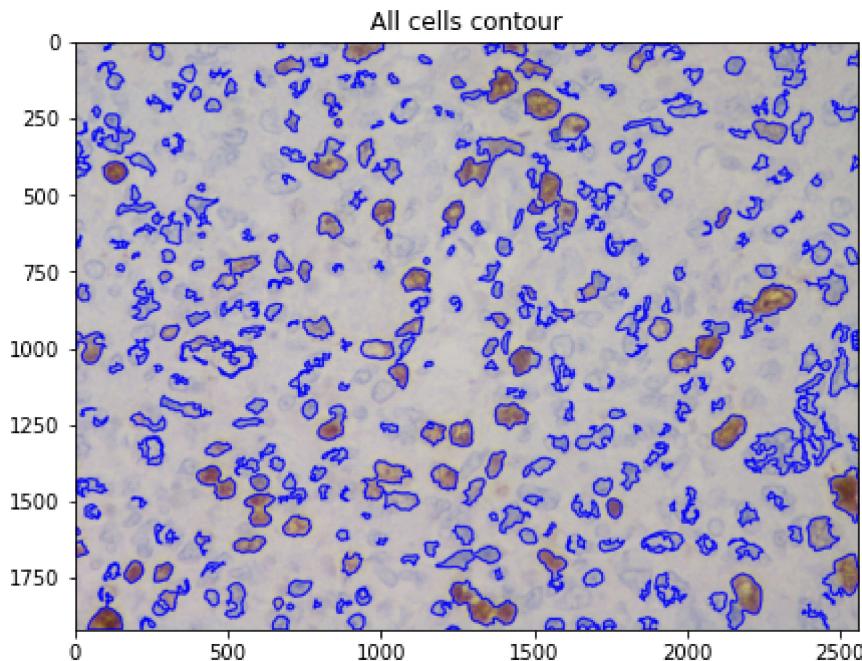
    area=cv2.contourArea(contour2[i])
    if(area>200):
        cv2.drawContours(image=img1,contours=contour2,contourIdx=i, color=(255,0,0), thickness=1)
        tcells=tcells+1

plt.figure(figsize=(7,7))
plt.title("All cells contour")
print("no of total cells: ",tcells)

img1=cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
plt.imshow(img1)

```

no of total cells: 321
Out[11]: <matplotlib.image.AxesImage at 0x1a8018524c0>



In [12]: percentage_positivity_1714=(cells/tcells)*100
percentage_positivity_1714

Out[12]: 19.626168224299064

```
In [13]: if(percentage_positivity_1714<15):
    print("low grade")
else:
    print("high grade")
```

high grade

image = 1718 KI67.jpg

```
In [21]: img1=cv2.imread("1718 KI67.jpg")

gray=cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
thresh, bin_img = cv2.threshold(gray, thresh=100, maxval=255, type=cv2.THRESH_BINARY_INV)
kernel = np.ones((3,3),np.uint8)
closed = cv2.morphologyEx(bin_img, cv2.MORPH_CLOSE, kernel=kernel, iterations=5)
contour_brown, hierarchy = cv2.findContours(image=closed.copy(), mode=cv2.RETR_EXTERNAL,
                                             contour_retrieval=cv2.CHAIN_APPROX_SIMPLE)

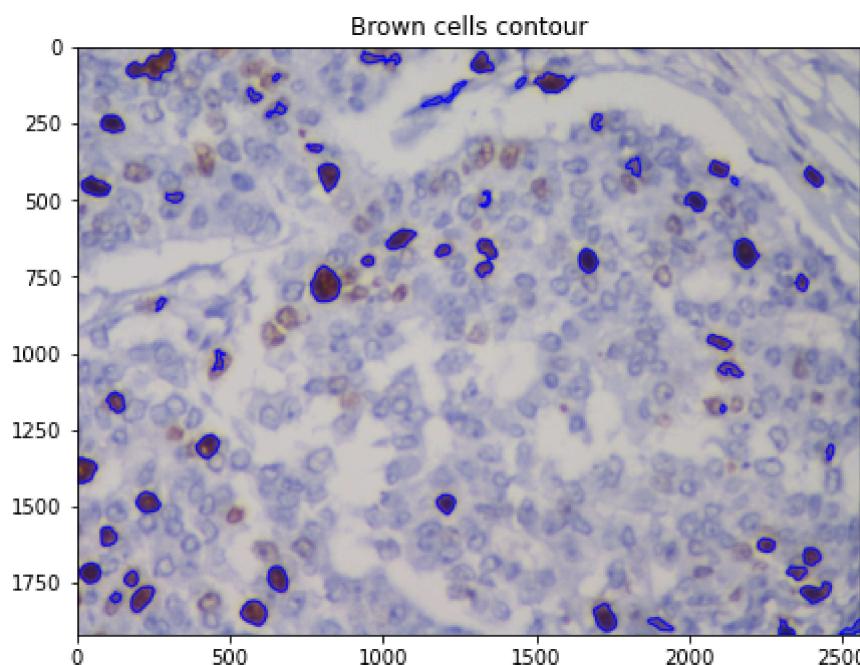
cells=0
for i in range(len(contour_brown)):

    area=cv2.contourArea(contour_brown[i])
    if(area>250):
        cv2.drawContours(image=img1, contours=contour_brown, contourIdx=i, color=(255,0,0))
        cells=cells+1

plt.figure(figsize=(7,7))
plt.title("Brown cells contour")
print("no of brown cells: ",cells)
img1=cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
plt.imshow(img1)
```

no of brown cells: 55

Out[21]: <matplotlib.image.AxesImage at 0x1a80016f0a0>



```
In [22]: img1=cv2.imread("1718 KI67.jpg",1)
lab= cv2.cvtColor(img1, cv2.COLOR_BGR2LAB)
```

```
l, a, b = cv2.split(lab)
clahe = cv2.createCLAHE(clipLimit=3.0, tileGridSize=(8,8))
cl = clahe.apply(l)

limg = cv2.merge((cl,a,b))
final = cv2.cvtColor(limg, cv2.COLOR_LAB2BGR)

gray=cv2.cvtColor(final, cv2.COLOR_BGR2GRAY)
thresh, bin_img = cv2.threshold(gray, thresh=100, maxval=255, type=cv2.THRESH_BINARY_INVERSE)
kernel = np.ones((3,3),np.uint8)
closed = cv2.morphologyEx(bin_img,cv2.MORPH_CLOSE, kernel=kernel, iterations=5)
contour2, hierarchy = cv2.findContours(image=closed.copy(), mode=cv2.RETR_EXTERNAL, method=cv2.CHAIN_APPROX_SIMPLE)

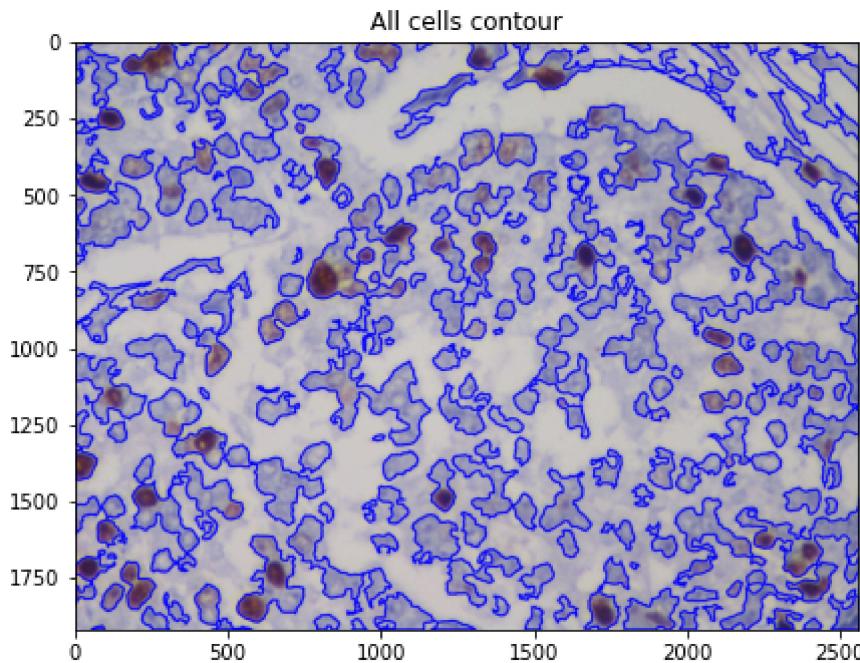
tcells=0
for i in range(len(contour2)):

    area=cv2.contourArea(contour2[i])
    if(area>250):
        cv2.drawContours(image=img1,contours=contour2,contourIdx=i, color=(255,0,0), thickness=1)
        tcells=tcells+1

plt.figure(figsize=(7,7))
plt.title("All cells contour")
print("no of total cells: ",tcells)

img1=cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
plt.imshow(img1)
```

no of total cells: 175
Out[22]: <matplotlib.image.AxesImage at 0x1a8001eef70>



In [23]: percentage_positivity_1718=(cells/tcells)*100
percentage_positivity_1718

Out[23]: 31.428571428571427

```
In [24]: if(percentage_positivity_1718<15):
    print("low grade")
else:
    print("high grade")
```

high grade

```
In [ ]:
```

image	manual count		program count		positivity		accuracy
	brown	total	brown	total	manual	prog	
3053	49	280	38	306	17.50	12.42	70.97
1714	76	358	63	321	21.23	19.63	92.46
1718	70	301	55	175	23.26	31.42	64.92
2141	105	212	112	199	49.52	56.28	86.34

```
In [ ]:
```

```
In [ ]:
```