ÜSKÜDAR ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ YÜKSEK LİSANS PROGRAMI

2023-2024

FİNAL ÖDEV RAPORU

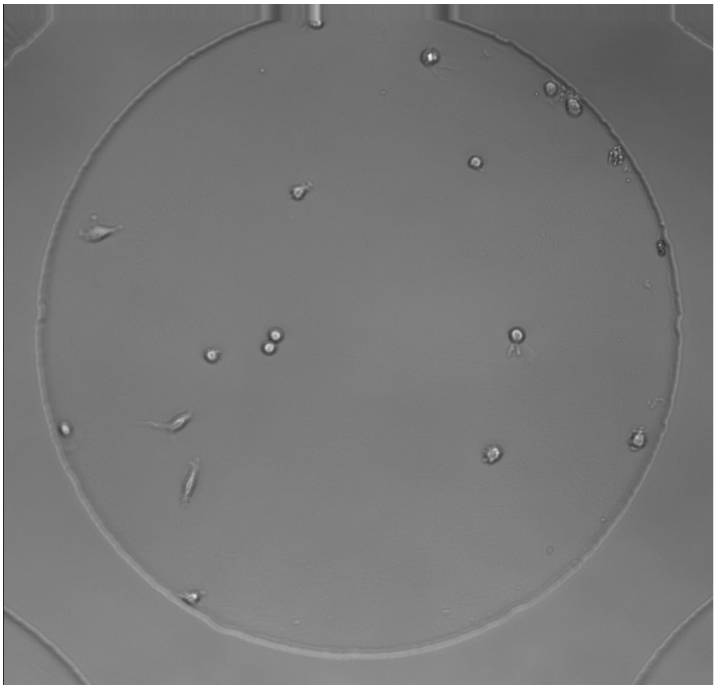
ÖĞRENCİ NUMARASI: 214327027

ÖĞRENCİ ADI-SOYADI: HAMZA TALİP EKŞİ

1. **GİRİŞ**

Bu çalışmada , gri seviyeli bir mikroskop hücre görüntüsü yüklenip görüntünün hücre sayısı bulunup öznitelikleri elde edilecek.

2. MATERYALLER VE YÖNTEMLER



Bu çalışmayı yapmak için yukarıdaki ‘png’ formatındaki resim kullanılmıştır.

3. ARAÇLAR

Çalışmanın yapılması için gerekli kod python dili ile Spyder üzerinde yazılmıştır.

4. KODLAR

```
import numpy as np
import cv2
from matplotlib import pyplot as plt
```

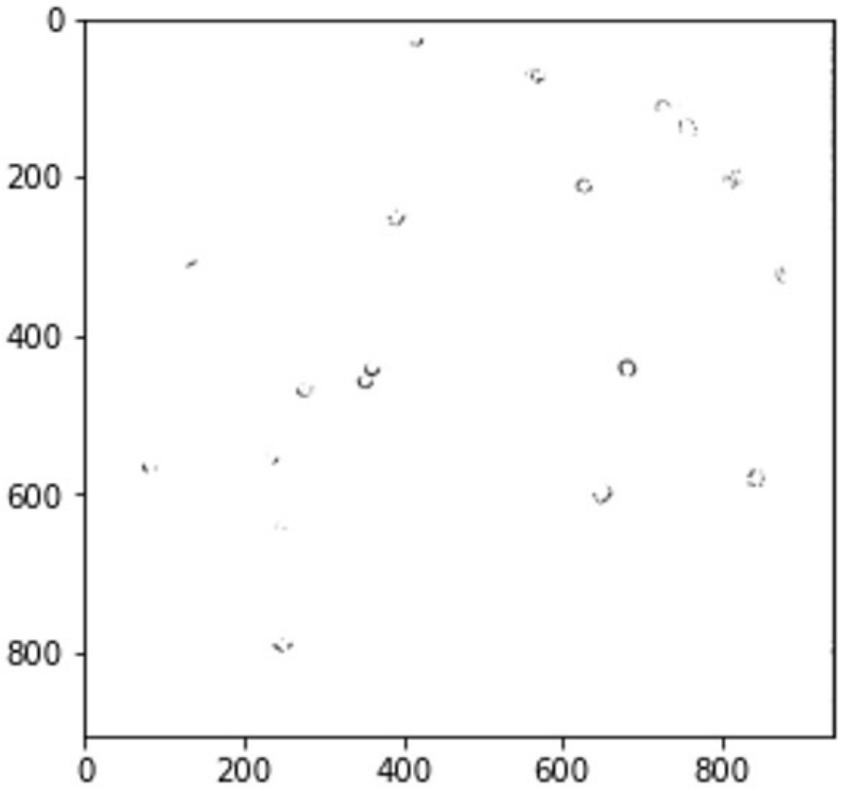
-Dizi ve matrislerle çalışmak için numpy,görüntü işleyebilmek için cv2,grafik elde etmek için plt kullanıldı.

```
path = "C:/Users/Talip/Desktop/FINAL/hucre.png"
img = cv2.imread(path,0)
```

-Resimin çekileceği lokasyon yazıldı ve opencv kütüphanesi sayesinde işlenebilecek duruma getirildi.

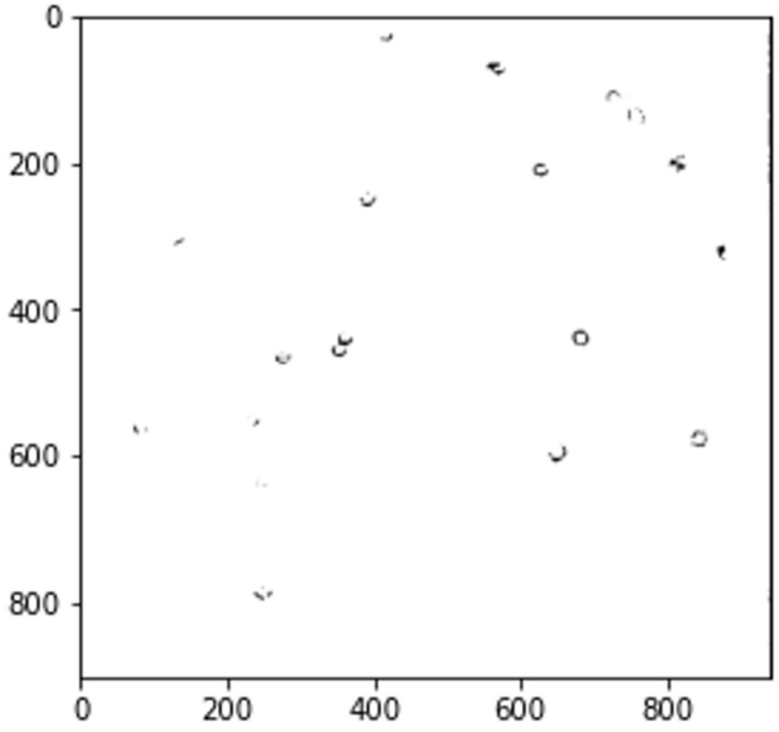
```
th1 = cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,\
            cv2.THRESH_BINARY,17,19)
```

-Adaptive filter uygulanarak görüntü önce gaussian filtresinden geçirilip binary threshold kullanılarak siyah-beyaz bir görüntü haline getirildi.



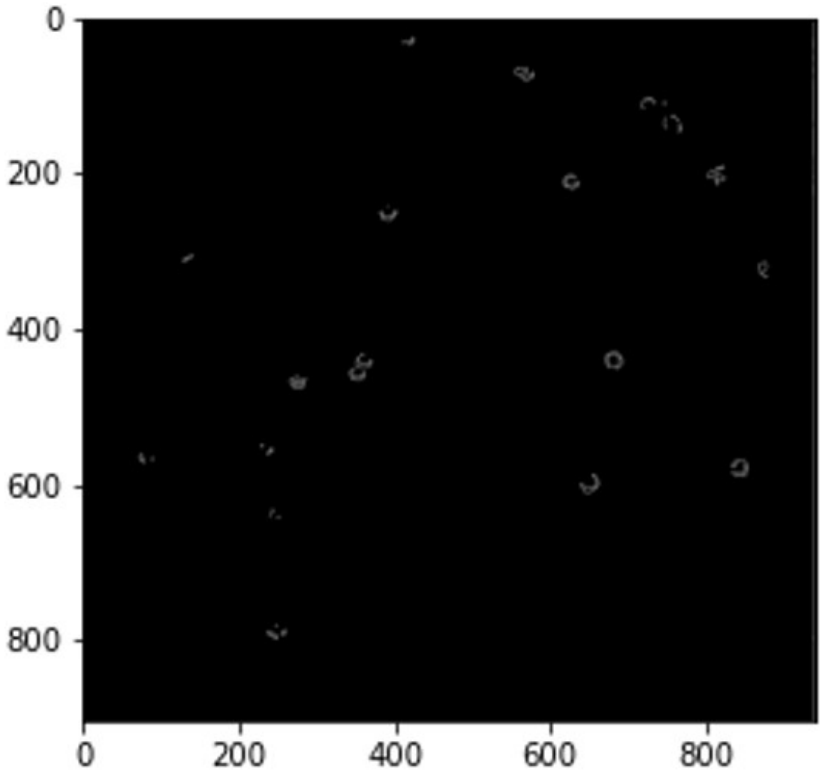

```
I = th1>70
I=I.astype(np.uint8)
I = 255*I

kernel = np.ones((5,9), np.uint8)
kernel2=np.ones((7,9),np.uint8)
erosion = cv2.erode(I, kernel, iterations = 1)
dilation = cv2.dilate(erosion, kernel, iterations=1)
```

-Görüntüdeki fazla bileşenlerden kurtulmak ve daha iyi bir görüntü elde etmek için iki kernel seçildi.İlk kernel ile erosion ve dilation uygulandı.



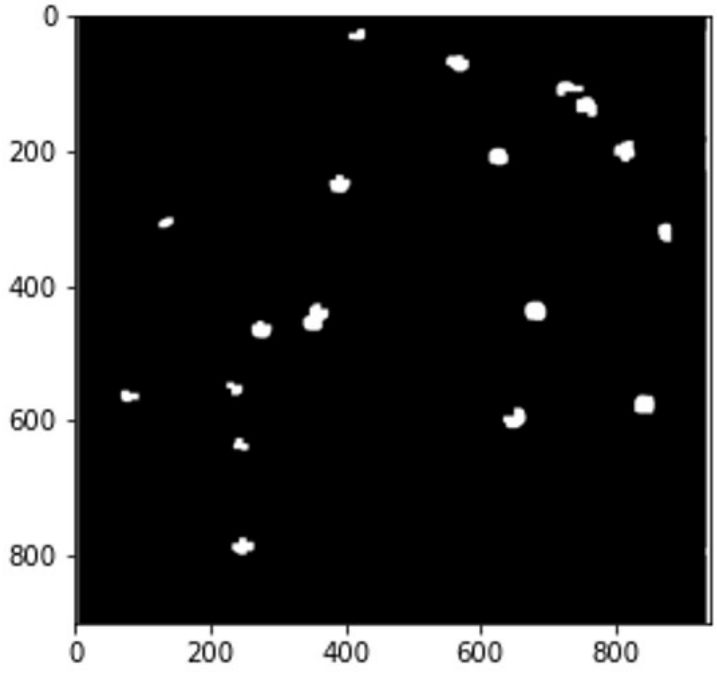
```
edges = cv2.Canny(dilation, 0, 255)
```

-Hassas bir kenar tespiti yapılabilmesi için edge canny kullanıldı.



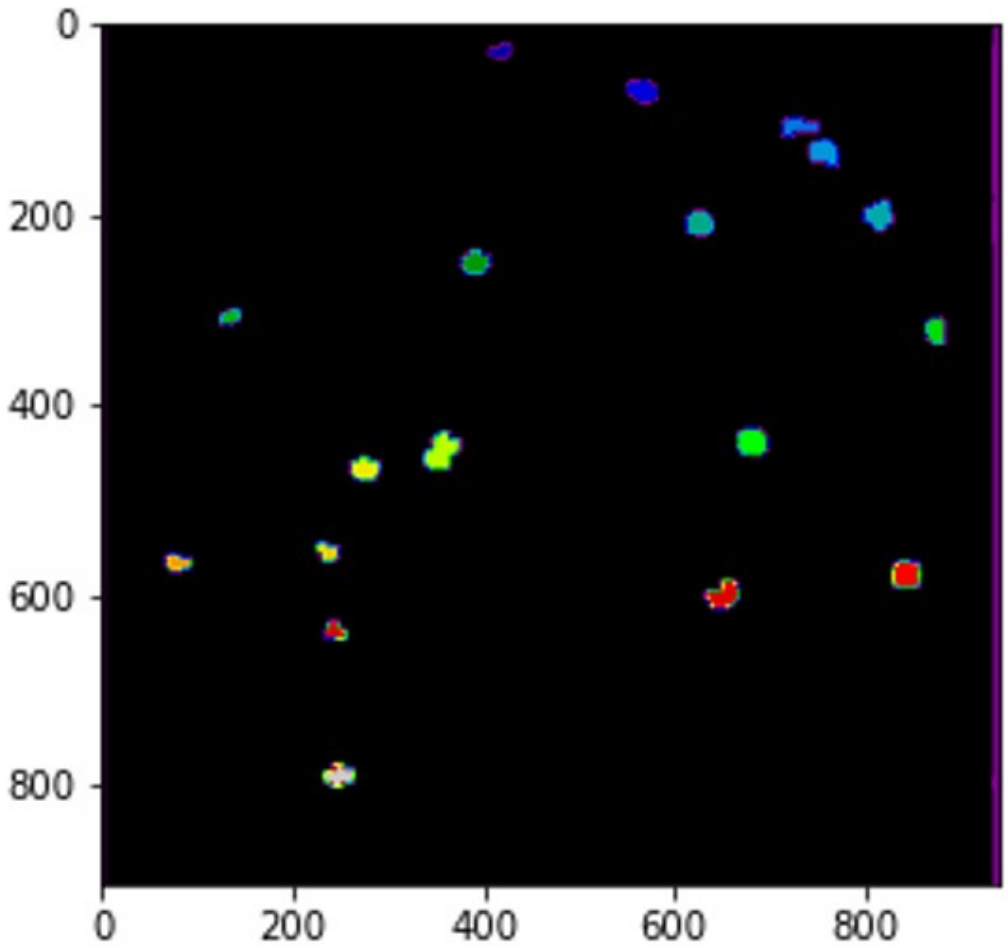

```
dilation2 = cv2.dilate(edges, kernel2, iterations=1)
opening = cv2.morphologyEx(dilation2, cv2.MORPH_OPEN, kernel2)
closing = cv2.morphologyEx(opening, cv2.MORPH_CLOSE, kernel2)
```

-Görüntüdeki nesneleri daha iyi hale getirmek için ikinci kernel ile dilation,opening ve closing yapıldı.Böylece daha iyi bir görüntü elde etmiş olduk.



```
marked_imgs, comp, stats, centroids = cv2.connectedComponentsWithStats(closing, connectivity=8)
sizes = stats[1:, -1]; marked_imgs = marked_imgs - 1
```

-Görüntüdeki her nesneyi etiketleyip renklendirdik.Bunu yaparken cv2.connectedComponentsWithStats kullanarak nesnelerle ilgili bazı verileri de elde etmiş olduk.



```
cx=np.zeros(20)
cy=np.zeros(20)
area=np.zeros(20)
perimeter=np.zeros(20, dtype=int)
aspect_ratio=np.zeros(20)
equi_diameter=np.zeros(20)
orientation = np.zeros(20)
circularity = np.zeros(20)
compactness = np.zeros(20)
area_to_perimeter_ratio = np.zeros(20)

for i in range(0, marked_imgs):
    img2 = np.zeros((comp.shape),dtype = np.uint8)
    img2[comp == i ] = 255
    # moments
    contours,hierarchy = cv2.findContours(img2, 1, 2)
    cnt = contours[0]
    M = cv2.moments(cnt)
    # center
    cx[i] = round(M['m10']/M['m00'])
    cy[i] = round(M['m01']/M['m00'])
    # contour area
    area[i] = cv2.contourArea(cnt)
    # perimeter
    perimeter[i] = cv2.arcLength(cnt,True)
    # aspect ratio
    x,y,w,h = cv2.boundingRect(cnt)
    aspect_ratio[i] = float(w)/h
    # eq dia
    are = cv2.contourArea(cnt)
    equi_diameter[i] = np.sqrt(4*are/np.pi)
    # orientation
    angle = 0.5 * np.arctan2((2 * M['mu11']), (M['mu20'] - M['mu02']))
    orientation[i] = np.degrees(angle)
    # circularity
    circularity[i] = (4 * np.pi * area[i]) / (perimeter[i] ** 2)
   # compactness
    compactness[i] = (perimeter[i] ** 2) / area[i]
   # area to perimeter ratio
    area_to_perimeter_ratio[i] = area[i] / perimeter[i]


min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(img,mask = img2)
```

-Öznitelikler formülize edilerek bulundu

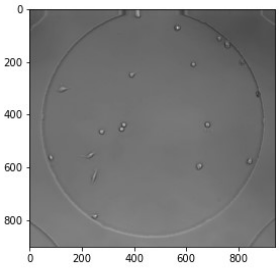
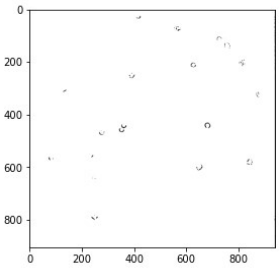
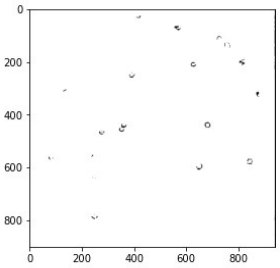
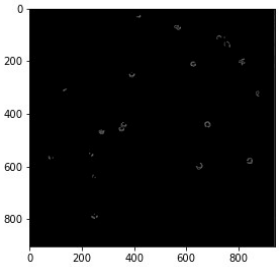
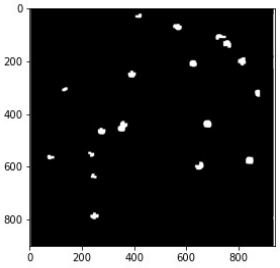
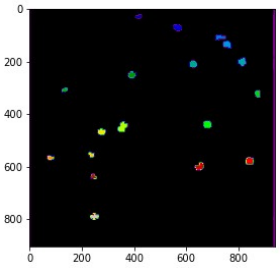
angle	float64	1	0.4836095477018729
are	float	1	249.5
area	Array of float64	(20,)	[600.5 3608. 7426. ... 665. 620.5 249.5]
area_to_perimeter_ratio	Array of float64	(20,)	[5.66509434 1.99116998 4.07574094 ... 6.71717172 5.64090909 3.51408451 ...
aspect_ratio	Array of float64	(20,)	[1.25925926 0.0055371 0.0110742 ... 1.07407407 1.1 1.22222222 ...
centroids	Array of float64	(21, 2)	[[467.53990013 452.02308272] [2. 451.]]
circularity	Array of float64	(20,)	[0.67160071 0.01380893 0.02811047 ... 0.852631 0.64441595 0.62196181 ...
closing	Array of uint8	(903, 942)	[[255 255 255 ... 255 255 255] [255 255 255 ... 255 255 255]
cnt	Array of int32	(17, 1, 2)	ndarray object of numpy module
comp	Array of int32	(903, 942)	ndarray object of numpy module
compactness	Array of float64	(20,)	[18.7110741 910.01773836 447.03528144 ... 14.73834586 19.5004029 . ...
contours	tuple	1	(Numpy array)

cx	Array of float64	(20,)	[249. 2. 937. ... 841. 651. 245.]
cy	Array of float64	(20,)	[789. 451. 443. ... 577. 599. 637.]
dilation	Array of uint8	(903, 942)	[[0 255 255 ... 0 0 0] [0 255 255 ... 0 0 0]
dilation2	Array of uint8	(903, 942)	[[255 255 255 ... 255 255 255] [255 255 255 ... 255 255 255]
edges	Array of uint8	(903, 942)	[[255 0 0 ... 0 0 0] [255 0 0 ... 0 0 0]
equi_diameter	Array of float64	(20,)	[27.65104603 67.77793356 97.23721952 ... 29.09818374 28.10774159 17.8 ...
erosion	Array of uint8	(903, 942)	[[0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0]
h	int	1	18
hierarchy	Array of int32	(1, 1, 4)	ndarray object of numpy module
I	Array of uint8	(903, 942)	[[0 255 255 ... 0 0 255] [0 255 255 ... 0 0 255]
i	int	1	19
img	Array of uint8	(903, 942)	[[14 91 94 ... 95 95 255] [14 91 94 ... 94 94 255]
img2	Array of uint8	(903, 942)	[[0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0]
kernel	Array of uint8	(5, 9)	[[1 1 1 ... 1 1 1] [1 1 1 ... 1 1 1]
kernel2	Array of uint8	(7, 9)	[[1 1 1 ... 1 1 1] [1 1 1 ... 1 1 1]
M	dict	24	{'m00':249.5, 'm10':61115.83333333333, 'm01':159022.5, 'm20':14977824. ...
marked_imgs	int	1	20
max_loc	tuple	2	(248, 630)
max_val	float	1	171.0
min_loc	tuple	2	(251, 641)
min_val	float	1	90.0
opening	Array of uint8	(903, 942)	[[255 255 255 ... 255 255 255] [255 255 255 ... 255 255 255]
orientation	Array of float64	(20,)	[-7.72093714 90. 89.96992008 ... -6.51124317 -36.2022503 . ...
path	str	38	C:/Users/Talip/Desktop/FINAL/hucre.png
perimeter	Array of int32	(20,)	[106 1812 1822 ... 99 110 71]
sizes	Array of int32	(20,)	ndarray object of numpy module
stats	Array of int32	(21, 5)	ndarray object of numpy module
th1	Array of uint8	(903, 942)	[[0 255 255 ... 0 0 255] [0 255 255 ... 0 0 255]
w	int	1	22
x	int	1	235
y	int	1	628

```
plt.figure(figsize=(30, 30))
plt.subplot(6, 1, 1)
plt.imshow(img, cmap="gray")

plt.subplot(6, 1, 2)
plt.imshow(th1,cmap="gray")

plt.subplot(6, 1, 3)
plt.imshow(dilation,cmap="gray")

plt.subplot(6, 1, 4)
plt.imshow(edges,cmap="gray")

plt.subplot(6, 1, 5)
plt.imshow(closing,cmap='gray')

plt.subplot(6, 1, 6)
plt.imshow(comp,cmap='nipy_spectral')

plt.show()
```

-İstenen sonuçlar tek bir plotta toplandı.

5.SONUÇLAR

are	float	1	249.5
area	Array of float64	(20,)	[600.5 3608. 7426. ... 665. 620.5 249.5]
area_to_perimeter_ratio	Array of float64	(20,)	[5.66509434 1.99116998 4.07574094 ... 6.71717172 5.64090909 3.51408451 ...
circularity	Array of float64	(20,)	[0.67160071 0.01380893 0.02811047 ... 0.852631 0.64441595 0.62196181 ...
compactness	Array of float64	(20,)	[18.7110741 910.01773836 447.03528144 ... 14.73834586 19.5004029 . ...
orientation	Array of float64	(20,)	[-7.72093714 90. 89.96992008 ... -6.51124317 -36.2022503 . ...