



İSTANBUL MEDENİYET ÜNİVERSİTESİ

BİL370 GÖRÜNTÜ İŞLEME DÖNEM PROJESİ

Hücre Tespiti, Sayımı & Merkezlerinin Belirlenmesi

Proje Yürütücüleri:

- Muhammet Hamza Yavuz - 21120205009
- Muhammed Nihat Aydın - 21120205062

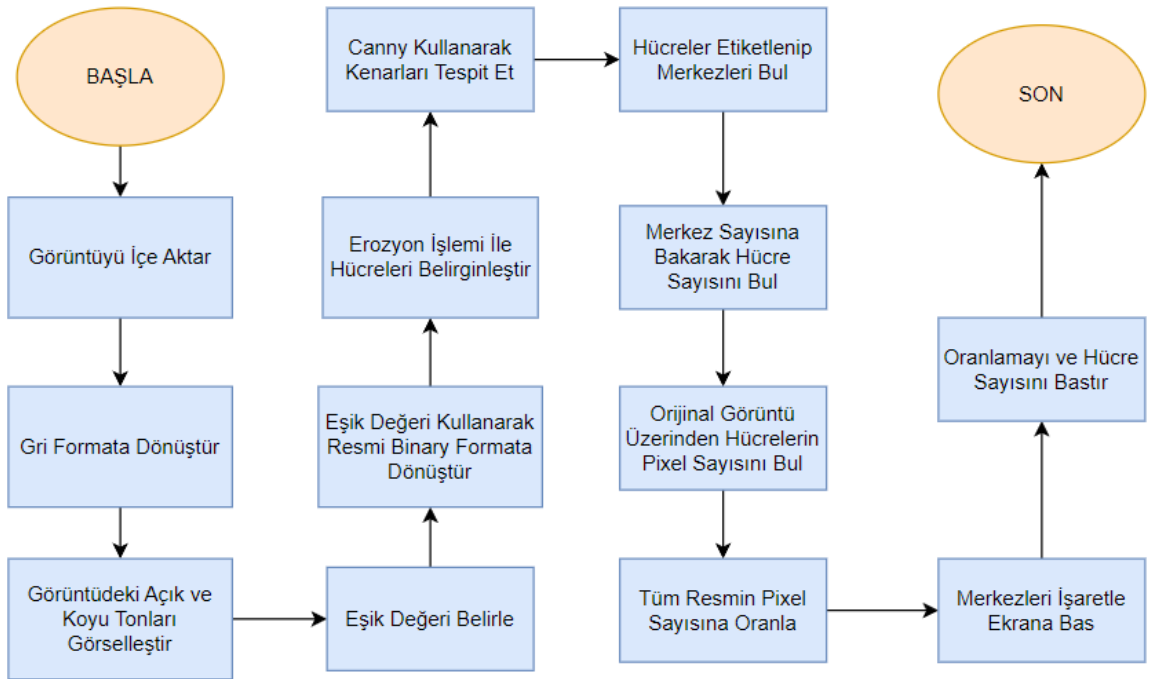
Proje Numarası: 5

Danışman: Nurullah Çalık

Rapor Tarihi: 31.05.2024

ÖZET

Bu projede, verilen görüntüdeki hücrelerin sayısı belirlenmiş, her hücrenin merkezi işaretlenmiş ve hücrelerin tüm görüntüye oranı hesaplanmıştır. İlk olarak, görüntü gri formata dönüştürülüp ikili (binary) hale getirilmiştir. Daha sonra erozyon işlemi uygulanarak hücre kenarları belirginleştirilmiş ve Canny kenar algılama yöntemiyle kenarlar tespit edilmiştir. Etiketleme işlemi sonrasında regionprops kullanılarak her hücrenin merkezi belirlenmiş ve bu merkezler görüntü üzerine işaretlenmiştir. Son olarak, hücrelerin kapladığı piksel sayısı toplam piksel sayısına oranlanarak hücrelerin tüm görsele oranı hesaplanmıştır. Bu işlemler Python yazılım dili içerisinde bulunan opencv, skimage, numpy ve matplotlib kütüphaneleri kullanılarak gerçekleştirilmiştir.



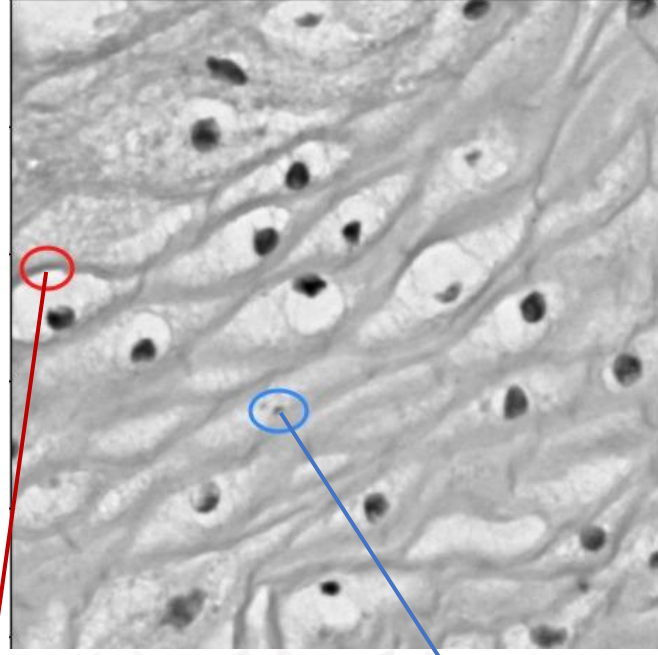
Şekil 1 [Proje adımları akış şeması]

YÖNTEM

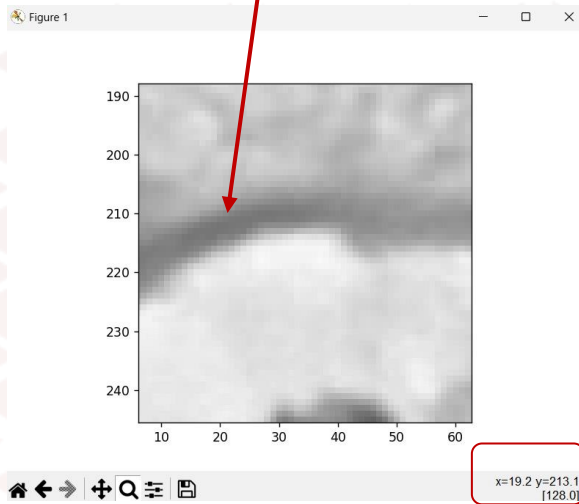
```
15 # Resmi okuyalım
16 image = cv2.imread('I.png')
17
18 # Görüntünün yüklenip yüklenmediğini kontrol edelim
19 if image is None:
20     raise ValueError(f"Görüntü dosyası acılamadı. Lütfen dosya yolunu kontrol edin!")
21
22 # Gri Formata dönüştürelim
23 # Gri formata dönüştürdüğümüz resim üzerinden hücrelerin pixel değerlerini inceleyelim
24 gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
25
26 # İncelediğimiz gri formattaki resimde 127 eşik değerinin uygun olduğuna karar verildi.
27 _, binary_image = cv2.threshold(gray, 127, 255, cv2.THRESH_BINARY)
```

Şekil 2 [Görüntünün gri formata dönüştürülmesi ve eşik değeri belirlenmesi]

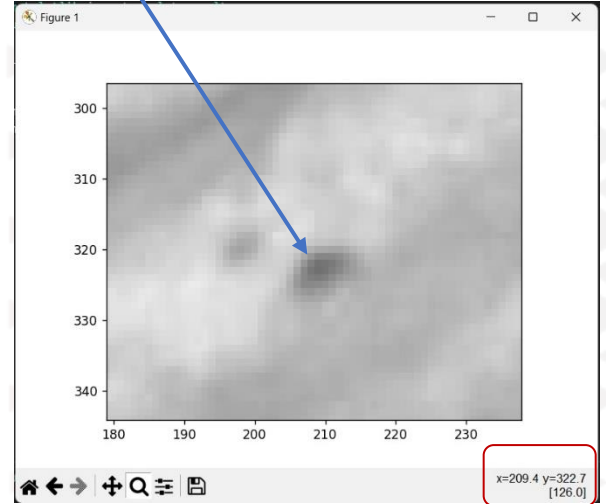
Görüntü Şekil 1’de de görüldüğü üzere ilk önce gri formata dönüştürülmüştür çünkü görüntüdeki hücreler koyu tonlardadır. Böylece görüntüdeki koyu tondaki bölgeler piksel tonlarına göre daha kolay ayırt edilebilecektir. Görüntü gri formata dönüştürüldükten sonra koyu renkteki hücreler arasında en açık tonlu olan hücre ve görüntüdeki hücre olmayan bölgelerdeki en koyu tonlu bölge belirlenmiştir. Ardından Şekil 2, Şekil 3 ve Şekil 4’te görüldüğü gibi bu bölgelerin piksel değerlerine dair ayrıntılı incelemeler yapılmıştır. Şekil 2 de kırmızı ile gösterilen yer görüntüdeki en koyu tonlu bölgelerden biridir. Yine Şekil 2 de mavi ile gösterilen yer ise en açık tonlu hücrelerden biridir. Bu bölgeler yaklaştırılarak Şekil 3 ve Şekil 4’teki gibi piksel değerleri gözlemlenmiştir. Şekil 3’te de görüldüğü üzere en koyu tonlu bölgedeki en düşük piksel değeri 128 olarak gözlemlenmiştir. Şekil 4’te ise en açık tonlu hücre için en yüksek piksel değeri olarak 126 ölçülmüştür. Bu sebeplerden dolayı görüntünün



Şekil 3 [Görüntüdeki en açık tonlu hücre ve en koyu tonlu bölge]

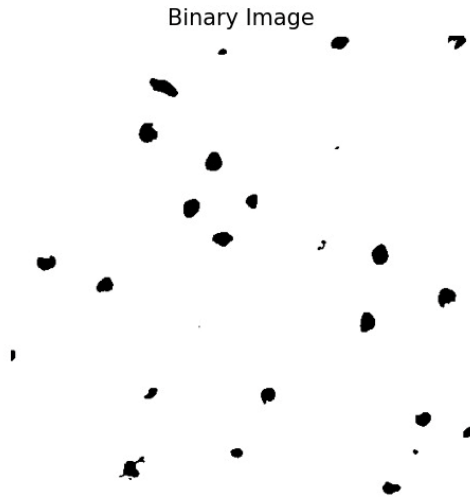


Şekil 4 [En koyu tonlu bölge]

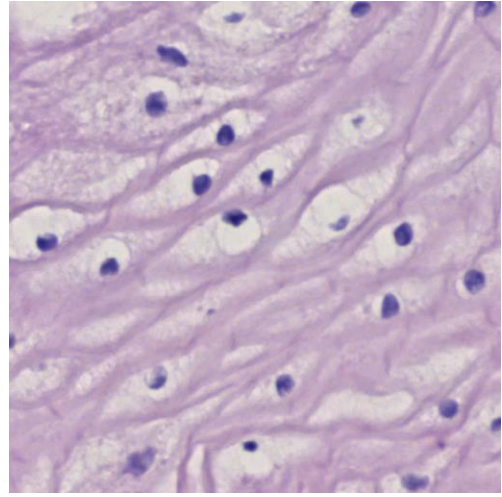


Şekil 5 [En açık tonlu hücre]

İkili (binary) formata dönüştürülmesine ve eşik değeri olarak ise 127 piksel değerinin belirlenmesine karar verilmiştir. Böylelikle görüntüdeki hücre olan kısımlar siyah (0 piksel değeri) olarak kalırken diğer (hücre olmayan) kısımlar ise beyaz (255 piksel değeri) olacaktır. Bu işlem Şekil 2’de görüldüğü gibi opencv kütüphanesi kullanılarak yapılmıştır. Gri formatına dönüştürdüğümüz görüntü girdi olarak verilmiştir. Şekil 5 ve Şekil 6’da göreceğiniz üzere orijinal görüntü ile kıyaslandığında çok az bir miktarda hücrelerdeki küçülme dışında görüntüdeki tüm hücreler elde edilmiştir.



Şekil 6 [İkili (binary) görüntü]



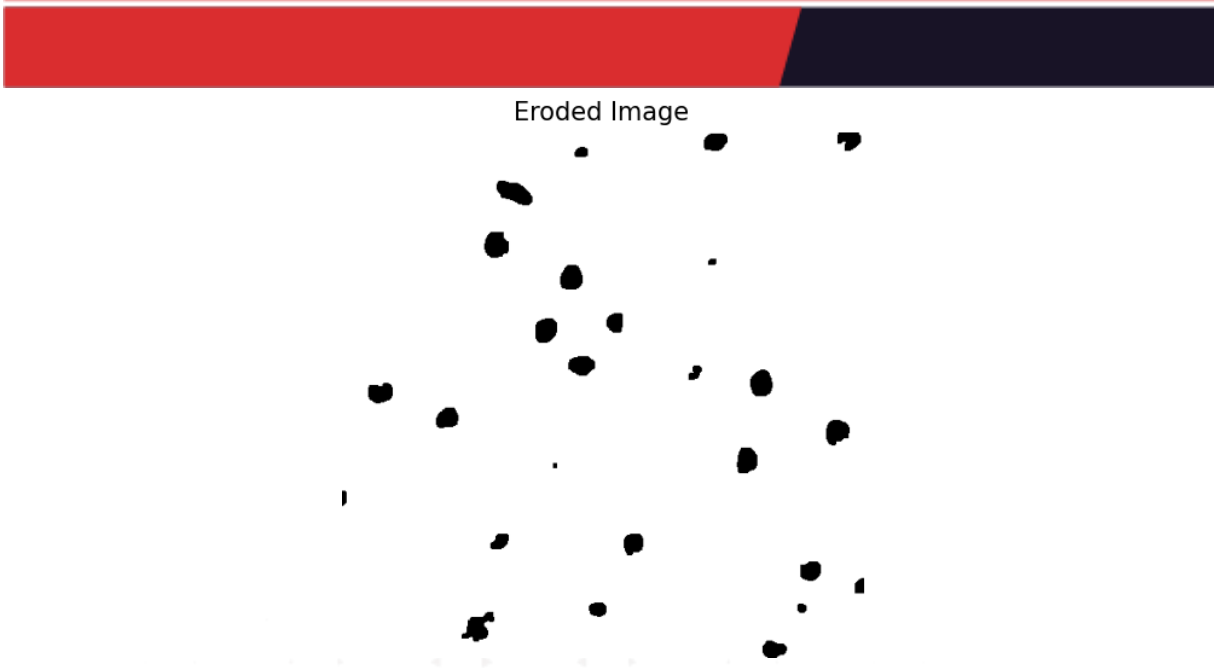
Şekil 7 [Orijinal görüntü]

Tüm hücreler ikili görüntüde gösterilmiştir. Burada gözüken hücreleri belirginleştirmek için bir morfolojik işlem uygulamamız gerekecektir. Hücrelerin belirginleşmesi kenar tespitinde ve hücrelerin merkezinin bulunmasında etkili olacağı düşünülmektedir. Şekil 8’de gösterilen kod çalıştırılmıştır.

```
29 # 2*2 lik 1 lerden oluşan matris oluşturalım
30 kernel = np.ones((2, 2), np.uint8)
31
32 # 1 lerden oluşan matris erozyon işleminde siyahları belirginleştirecektir.
33 # İterasyon sayısının az olmasının sebebi merkezlere konulacak noktalar sapmamalıdır.
34 eroded_image = cv2.erode(binary_image, kernel, iterations=3)
```

Şekil 8 [Morfolojik İşlem Kodu]

Morfolojik işlem olarak erozyon işlemi seçilmiştir. Çünkü erozyon işlemi ikili bölgedeki siyah alanları çoğaltması beyaz alanları azaltması sebebiyle tercih edilmiştir. Erozyon işlemi 2x2’lik birlerden oluşan matrisle 3 iterasyon boyunca uygulanmıştır. Erozyon işlemi yapıldıktan sonra oluşan görüntü Şekil 9’da verilmiştir.

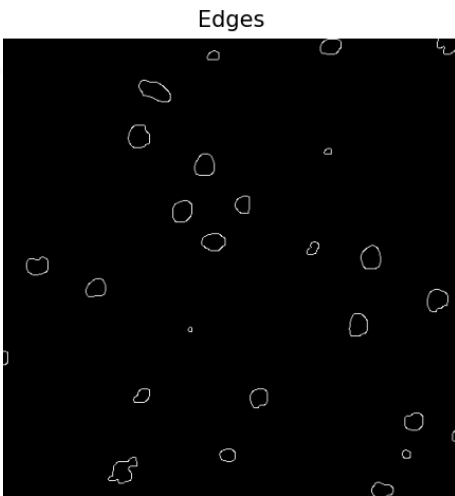


Şekil 9 [Morfolojik İşlem Sonucu]

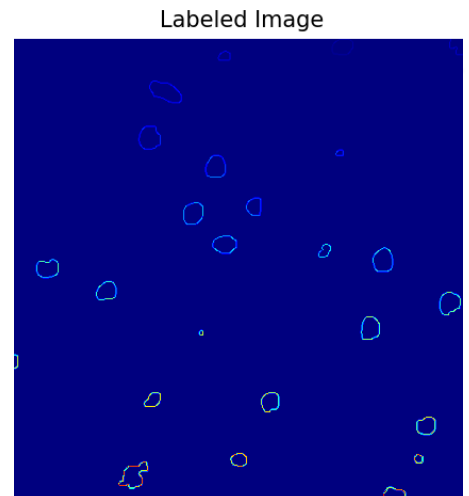
Elde ettiğimiz son görüntü üzerinden kenar tespiti işlemini gerçekleştirilmiştir. Kenar tespiti gerçekleştirilmesi opencv kütüphanesindeki canny fonksiyonu kullanılarak yapılmıştır. Kenar tespit kodları ve parametreleri Şekil 10'da gösterilmiştir. Ayrıca Şekil 10'da skimage kütüphanesindeki label fonksiyonu kullanılmıştır. Bu fonksiyon yardımıyla kapalı konturlar etiketlenmiş ve etiketler Şekil 11'de farklı renklerde görselleştirilmiştir.

```
36 # Merkezleri işaretlemek için kenarları tespit edelim
37 edges = cv2.Canny(eroded_image, 30, 140)
38
39 # Birleşik olan kontörler bulunur etiketlenir
40 labeled_image = label(edges)
```

Şekil 10 [Kenar Tespit ve Etiketleme Kodu]



Şekil 11 [Kenar Tespiti]



Şekil 12 [Etiketli Hücreler]

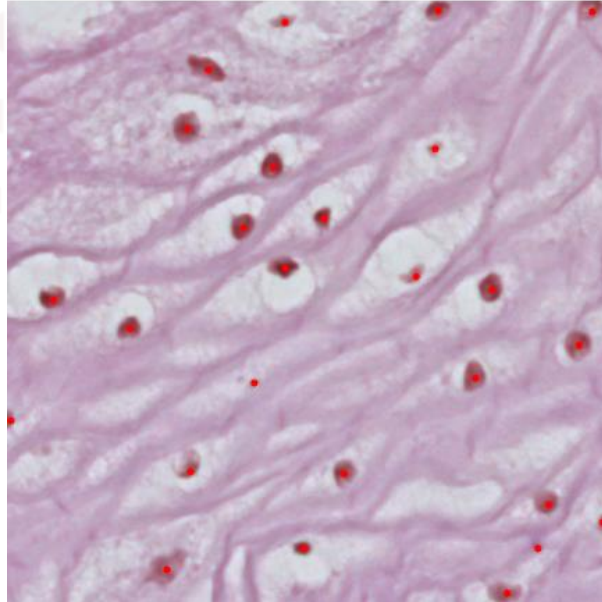
Etiketlenmiş hücrelerin her biri için merkez bulma işlemini skimage kütüphanesi içerisinde regionprop fonksiyonu kullanılarak gerçekleştirilmiştir. Ayrıca döngü içerisinde hücre sayma işlemi bir değişken yardımıyla yapılmıştır. Bu fonksiyonun kullanımı, hücrelerin merkezine nokta koyma ve hücre sayma işlemi Şekil 13'teki kod ile yapılmıştır.

```
42 # Region prop ile merkezleri bulalım
43 props = regionprops(labeled_image)
44
45 # Hücre sayısını 0 dan başlatalım
46 i=0
47 # merkezleri gezelim ve noktaları koyalım
48 for prop in props:
49     # Hücreleri sayalım
50     i+=1
51     center_y, center_x = prop.centroid
52     cv2.circle(image, (int(center_x), int(center_y)), 3, (0, 0, 255), -1)
```

Şekil 13 [Hücrelerin Merkezlerinin İşaretlenmesi ve Hücre Sayımı]

Yapılan işlem sonucunda hücrelerin merkezleri doğru bir şekilde işaretlenmiştir. Bu durum orijinal görüntü üzerinde Şekil 14'te gösterilmiştir.

Original Image



Şekil 14 [Merkezleri İşaretlenmiş Hücreler]

Merkezi tespit edilmiş hücrelerin tüm görüntüye oranı Şekil 6 kullanılarak Şekil 15'teki gibi hesaplanmıştır. Hesaplama sonucu yüzdeye çevrilerek konsola bastırılmıştır. Bununla beraber hücrelerin toplam piksel sayısı, resimde bulunan piksel sayısı ve hücre sayısı konsola yazılmıştır.

```
54 # Hücrelerin tüm resme oranını bulalım öncelikle hücrelerin bulunduğu pixelleri sayalım
55 black_pixels = sum(1 for pixel in binary_image.flatten() if pixel == 0)
56 print(f"Hücrelerin bulunduğu pixel sayısı {black_pixels}")
57
58 # Toplam piksel sayısını bulalım
59 total_pixels = binary_image.size
60 print(f"Resimde bulunan pixel sayısı: {total_pixels}")
61 # Siyah piksellerin tüm resme oranını hesaplayalım
62 black_ratio = (black_pixels / total_pixels) * 100
63 print(f"Hücrelerin tüm resme oranı : %{black_ratio}")
64 print(f"Hücre Sayısı: {i} ")
```

Şekil 15 [Görüntüdeki Hücrelerle İlgili Hesaplamalar]

SONUÇ

Bu projede, yüklenen 'l.png' görüntüsündeki hücrelerin sayısı, merkez noktalarının işaretlenmesi ve hücrelerin tüm görsele oranının hesaplanması hedeflenmiştir. Gri formata dönüştürülen görüntüde, uygun eşik değeri belirlenmiş ve ardından erozyon işlemi uygulanmıştır. Kenar tespit işlemi sonrasında etiketlenmiş görüntü üzerinde regionprops ile hücrelerin merkezleri belirlenmiştir. Yapılan analiz sonucunda:

- Görüntüde toplam 26 hücre tespit edilmiştir.
- Her hücrenin merkezi, kırmızı noktalarla işaretlenmiştir.
- Hücrelerin kapladığı piksel sayısı 4964 olarak hesaplanmıştır.
- Görüntüdeki toplam piksel sayısı 262144 olup, hücrelerin tüm görüntüye oranı yaklaşık olarak %1,89 olarak bulunmuştur.

Proje için oluşturulmuş olan web sitesine ait çıktı streamlit.pdf şeklinde proje kapsamında yazılan kodları ise 2 dosya halinde streamlit.py ve main.py olacak şekilde ekte paylaşılmıştır.