GÖRÜNTÜ İŞLEME ÖDEV 1 RAPOR

Hazırlayan : Yunus Karatepe, 17011051

Ders : Görüntü İşleme BLM4540

YÖNTEM:

Algoritmayı kısaca anlatmak gerekirse :

Öncelikle openCV kütüphaneleri ile resmimizi 3 renk grubu da ayrı ayrı matrislerde saklanacak şekilde bir veri yapısına alıyoruz. Toplamda red, green ve blue değerleri için 3 farklı matris tutmuş oluyoruz. Daha sonra clustring() fonksiyonu ile bir label matrisi oluşturuyoruz. Clustring fonksiyonu resmi verilen k değeri kadar renk parçasına bölüyor. Bunu yaparken de her pixelin hangi renge daha yakın olduğunu bulup o rengin randomCreator() fonksiyonu ile oluşturulmuş randomPixel() renklerinden hangisine daha yakınsa onun indis değerini label olarak veriyor. Toplamda label matrisimizde 0, 1, 2, ... , k-1 sayıları bulunmuş oluyor. Bu şekilde label matrisimize randomPixel[3][k] dizisinde bulunan renk değerlerini verdikten sonra yani :

```
img.at<Vec3b>(i, j)[0] = randomPixel[0][ labelMat[i][j] ];

img.at<Vec3b>(i, j)[1] = randomPixel[1][ labelMat[i][j] ];

img.at<Vec3b>(i, j)[2] = randomPixel[2][ labelMat[i][j] ];
```

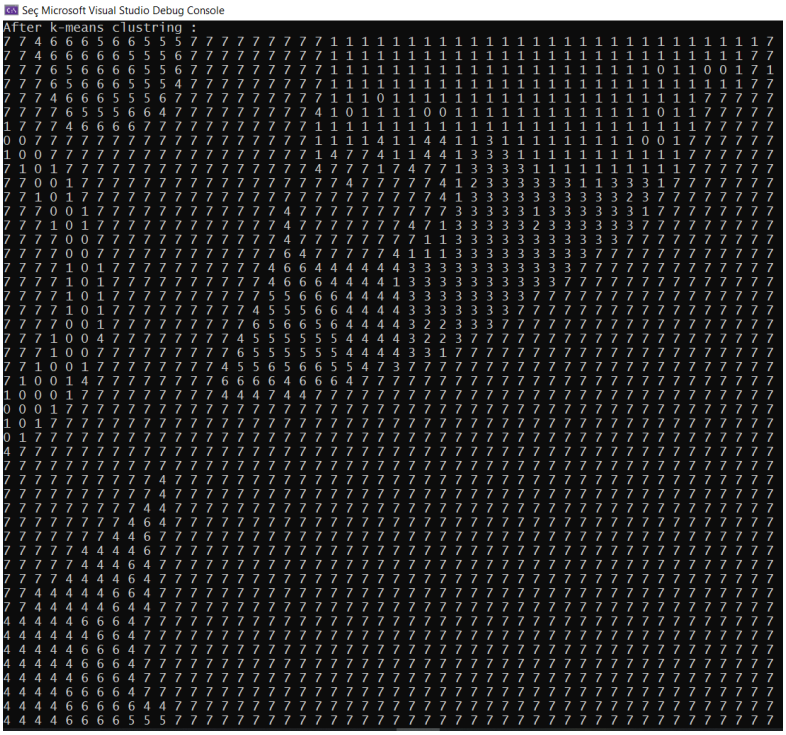
şeklinde clustered Imagemizi oluşturmuş oluyoruz.

Clustring kısmı bittikten sonra ise resmi birbiriyle bağlantılı olmayan segmentlerini belirleyip yeniden labellama işlemi yapıyoruz. Bunu da yaparken sol üst satırdan başlayıp

(ilk satırı önce kendi arasında labelliyoruz) daha sonra for döngüsünde matrisin sonuna kadar giderek her pixeli, solu, sol yukarı çaprazı, yukarısı ve sağ üst çaprazıyla kontrol ediyoruz. Eğer solu ya da sol üstüyle aynı ise ayrıyeten tekrar sağ üst çaprazı ile kontrol ediyoruz ki bunlar label matrisinde aynı iseler yeni label matrisimizde farklı labellanma olması muhtemel. Eğer daha önceden labellanmış ise sağ üstün bütün labellarını o indis değerine verdiğimiz label ile değiştiriyoruz. Böylece çaprazdan bağlantı var ise farklı renk ile göstermemiş oluyoruz. En sonunda ise bu labellara random renk değerleri vererek segmentleri elde ediyoruz.

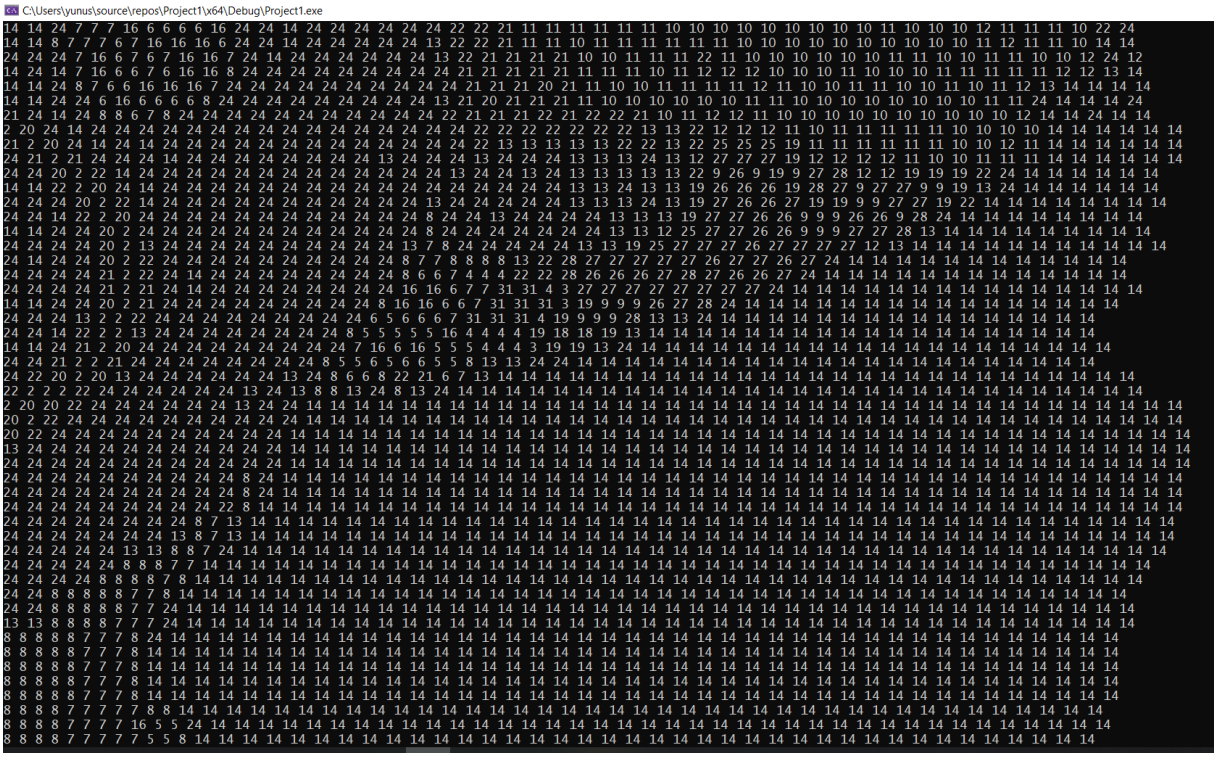
Clustring yapıldıktan sonra bazı matris örnekleri :

clown.jpg için k = 8, treshold = 30, i = 150-200 ve j = 150-200 aralığı :



clown.jpg için k = 16, treshold = 30, i = 150-200 ve j = 150-200 aralığı :

clown.jpg için k = 32, treshold = 30, i = 150-200 ve j = 150-200 aralığı :



gibi farklı label matrisleri oluşmaktadır.

Bunların görsel resimleri ise sırasıyla k = 8, k = 16 ve k = 32 (treshold = 30) için şöyledir:

k = 8

k = 16

k = 32

Aynı şekilde sırasıyla k = 8, 16, 32 (treshold = 30) için connectedComponentSegmentation sonrası imageler ise şöyledir:

k = 8

k = 16

k = 32

Sonuç:

Sonuç olarak bize verilen bir resmi önce istenilen kadar renk parçasına bölme işlemi yaptık (clustring() fonksiyonu ile k adet renk parçasına). Daha sonra ise bu parçalarına böldüğümüz resmin her parçasını farklı bir renk ile gösterdik (connectec_Component_Segmentation() fonksiyonu ile). Böylece connected component segmentatiton algoritmasını anlamış olduk.

NOT: Kod çıktı olarak 2 resim vermektedir. Bunlardan birisi "Clustring.png" : resmi k parçaya böldükten hemen sonra k adet renk ile kümelenmiş halidir. Diğeri ise son olarak çıktı verdiğimiz "Connected_Component_Segmantation.png" : resmin birbirine değmeyen her segmentinin farklı renk ile gösterildiği resimdir.