

T.C.
AKDENİZ ÜNİVERSİTESİ



HAZIRLAYAN
Büşra OLGUN
20163405002

FEN FAKÜLTESİ
UZAY BİLİMLERİ VE TEKNOLOJİLERİ BÖLÜMÜ
DİJİTAL GÖRÜNTÜ İŞLEME TEKNİKLERİ
1. ÖDEV RAPORU

KASIM 2021
ANTALYA

1. Ödev

Radyometrik çözünürlüğü 3 bit olan, tam sayı (integer) değerlerine sahip 5x5 boyutunda rastgele görüntü oluşturarak frekans ve bağıl frekansı hesaplayınız. Grafiklerini oluşturunuz.

Görüntüyü Python üzerinde kod yazarak oluşturduk. Bu programla ilk önce 5x5 lik bir matris oluşturduk ve bu matrise 3 bitlik görüntü olacağı için 0-7 arasında rastgele değerler atadık. Burada sınıf genişliğini 1 olarak aldık. Her bir değer için frekans değerlerini hesaplattık.

$$\text{Bağıl Frekans} = \text{Sınıf Frekansı} / \text{Toplam Frekans}$$

Formülünü kullanarak her bir değer için bağıl frekans değerlerini hesaplattık.

$$\text{Yoğunluk} = \text{Bağıl Frekans} / \text{Sınıf Genişliği}$$

Formülünü kullanarak da her bir değer için yoğunluk hesabını yaptık.

Son olarak x eksenine sınıf genişliği, y eksenine bağıl frekans olarak grafik çizdirdik ve x eksenini sınıf genişliği, y eksenini ise yoğunluk olacak şekilde grafik çizdirdik. Böylece oluşturduğumuz görüntünün histogramlarını oluşturmuş olduk.

5x5 Boyutunda 3 Bitlik Görüntü

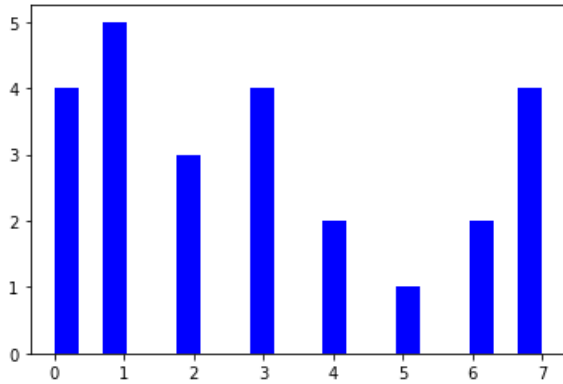
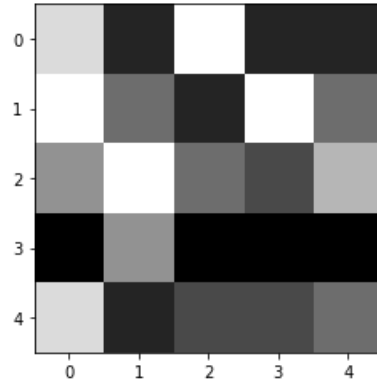


Figure 1: Oluşturulan Görüntünün Frekans Histogramu

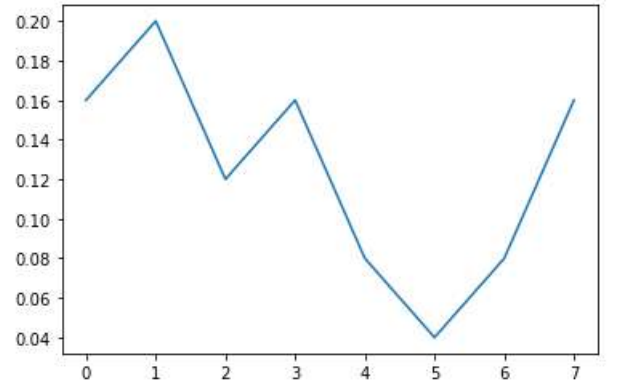


Figure 2: Oluşturulan Görüntünün Bağıl Frekans Histogramu

2. Ödev

Karmaşık olmayan bir fotoğraf üstünde görüntü işleme işlemlerinden histogram öteleme ve histogram eşikleme işlemlerini yapınız.

Histogram Eşikleme

Eşikleme işlemini aşağıdaki python kodlarını kullanarak yaptım.

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

img = cv.imread('resim1.jpg',0)
img = cv.medianBlur(img,5)
ret,th1 = cv.threshold(img,75,255,cv.THRESH_BINARY)
images = [th1]
for i in range(1):

    plt.subplot(2,2,i+1),plt.imshow(images[i],'gray')

    plt.xticks([]),plt.yticks([])
plt.show()
```

Histogram eşikleme yaparken bir veya iki değer belirlememiz gerekmektedir. Ben bu görüntü için ilk uyguladığımda eşik değerini 125 olarak belirledim. İkinci uyguladığımda ise eşik değerimi 75 olarak belirledim. Eşikleme işlemini yaptığımda karşıma siyah beyaz değerlere sahip bir görüntü çıkmasını ve histogramında sadece sıfır ve bir değerlerinden oluşmasını bekleriz.



Figure 3: Seçilen Görüntü

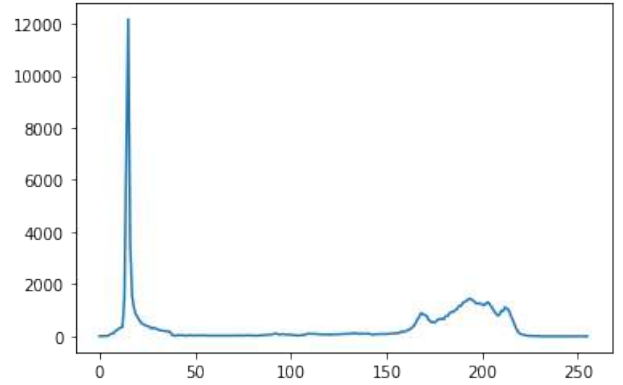


Figure 4: Görüntünün Histogramı

Aşağıda görüldüğü gibi eşik değerini azaltınca nesnenin üzerindeki sembol daha net hale gelmiştir. Eğer nesnenin üzerindeki sembolleri de tespit etmek istersek 75 veya altında bir eşik değeri seçmemiz gerekmektedir.

Nesnenin üzerindeki sembol belirginleştiği için siyah değerinde azalma, beyaz değerinde artma gözlemliyoruz. Histogramlarda siyah 0 değerini, beyaz ise 255 değerini ifade etmektedir. Eşik değeri azalınca siyah değerler azalmış, beyaz değerler artmıştır.

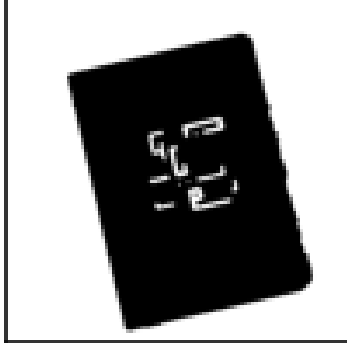


Figure 5: Eşik Değeri 125 Olan Eşikleme Uygulanmış Görüntü

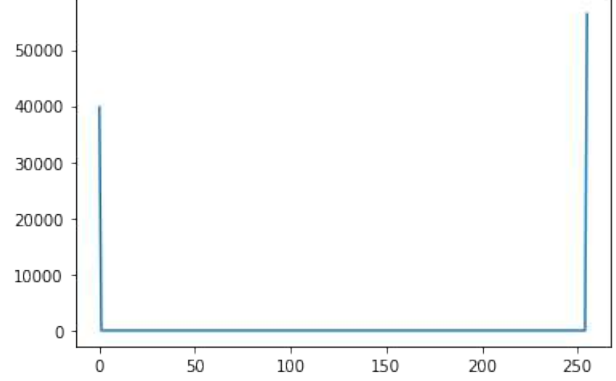


Figure 6: Eşikleme Uygulanan Görüntünün Histogramı

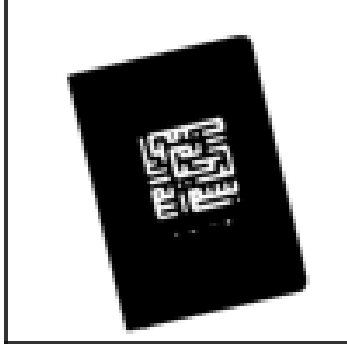


Figure 7: Eşik Değeri 75 Olan Eşikleme Uygulanmış Görüntü

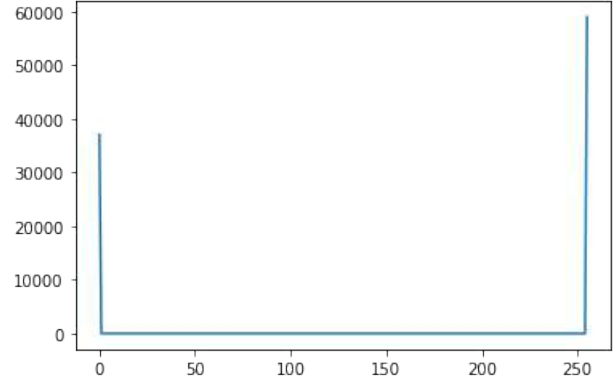


Figure 8: Eşikleme Uygulanan Görüntünün Histogramı

Histogram Öteleme

Histogram öteleme işlemini aşağıdaki python kodlarını kullanarak yaptım.

```
brightness = 75
img= np.int16(img)
img = img + brightness
img =np.clip(img,0,255)
img =np.uint8(img)
cv2_imshow(img)
hist = cv2.calcHist([img],[0],None,[256],[0,256])
plt.plot(hist)
plt.show()
```

Histogram öteleme yaparken aşağıdaki formül uygulanır.

$$G(x, y) = F(x, y) \pm b$$

$F(x, y)$ = Orjinal Görüntü

$G(x, y)$ = Çıktı Görüntü

b = Sabit

Burada orjinal görüntüye 50 değerini eklediğimizde aşağıdaki görüntüyü ve histogramını elde ederiz.



Figure 9: Orjinal Görüntüye 50 Birim Parlaklık Eklenmiş Hali

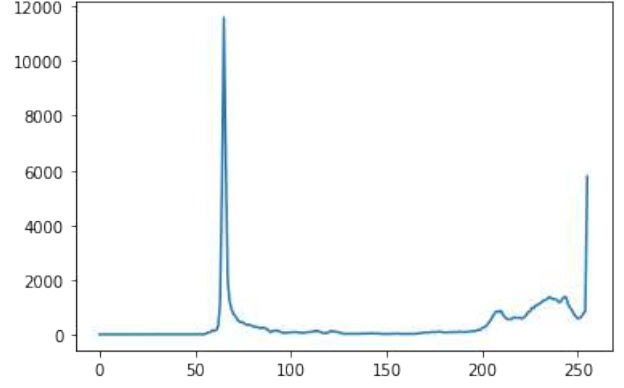


Figure 10: Öteleme Uygulanmış Görüntünün Histogramı

Orjinal görüntüden 50 değerini çıkararak aşağıdaki görüntüyü ve histogramını elde ederiz.



Figure 11: Orjinal Görüntüye 50 Birim Parlaklık Çıkarılmış Hali

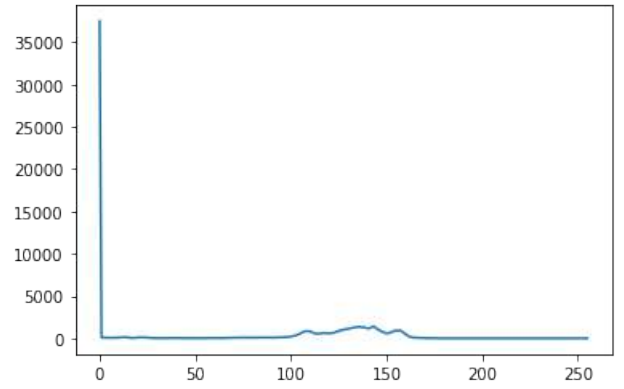


Figure 12: Öteleme Uygulanmış Görüntünün Histogramı

İlk görüntüde 50 değerini eklediğim zaman histogram sağa ötelenmiş ve görüntüdeki parlaklığın arttığını görüyoruz. Ayrıca görüntü değerleri sağ tarafa toplandığı için histogramın sağ tarafta 6000 değerine yakın bir veri topluluğu mevcuttur.

İkinci görüntüde ise 50 değerini çıkardığım zaman histogram sola ötelenmiş ve parlaklığın orjinal görüntüye göre azaldığını görmekteyiz. Ayrıca görüntü değerleri sol tarafa toplandığı için histogramın sol tarafındaki değerlerin 35000 değerine yakın bir veri topluluğu mevcuttur.

3. Ödev

Karmaşık bir fotoğraf üstünde histograma lineer germe işlemini uygulayınız.

Histograma lineer germe işlemi uygularken aşağıdaki python kodlarını kullanarak yaptım.

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

img= cv2.imread("resim2.jpg")
ori = img.copy()
xp=[0,64,128,192,255]
fp=[0,16,128,240,255]
x=np.arange(256)
table = np.interp(x,xp,fp).astype('uint8')
img = cv2.LUT(img, table)
cv2.imshow(ori)
cv2.imshow(img)
cv2.waitKey(0)
cv2.destroyAllWindows()

hist = cv2.calcHist([img],[0],None,[256],[0,256])
plt.plot(hist)
plt.show()
hist2 = cv2.calcHist([ori],[0],None,[256],[0,256])
plt.plot(hist2)
plt.show()
```

Histogram germe işleminden sonra görüntü kontrastında artma mevcuttur. Histogramlarda ise verilerin 0 ve 255 değerine doğru kaydıkları hatta bazılarının orada toplandıkları görülmektedir.



Figure 13: Orjinal Görüntü

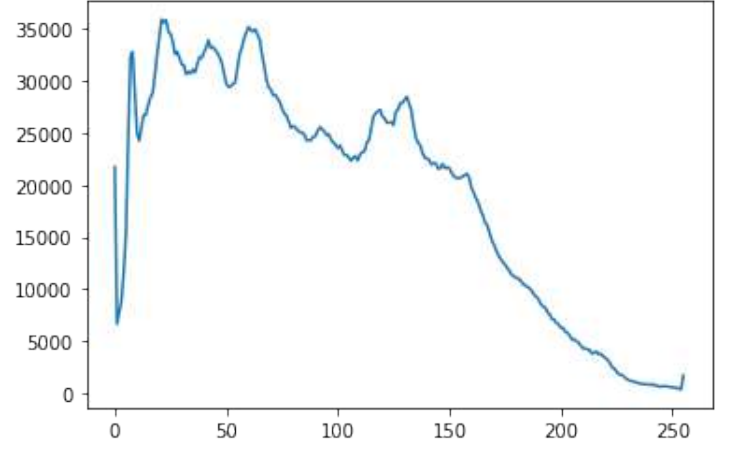


Figure 14: Orjinal Görüntü Histogramı



Figure 15: Linner Germe Uygulanmış Görüntü

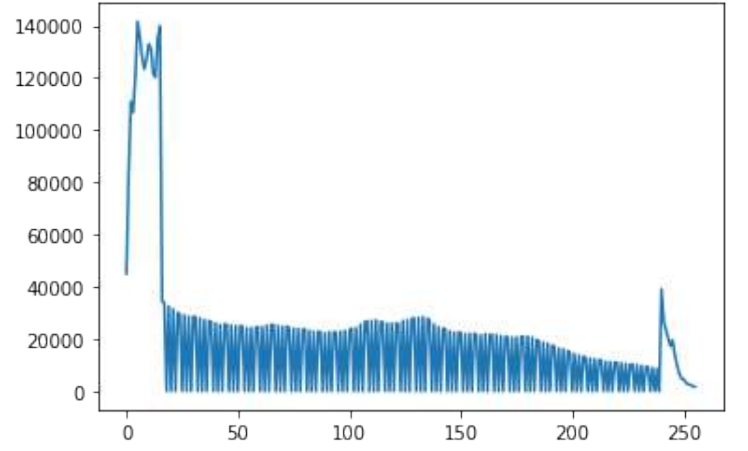


Figure 16: Linner Germe Uygulanmış Görüntünün Histogramı