

ÜSKÜDAR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ YÜKSEK LİSANS PROGRAMI
2023-2024
VİZE ÖDEV RAPORU



ÖĞRENCİ NUMARASI: 214327027

ÖĞRENCİ ADI-SOYADI: HAMZA TALİP EKŞİ

1. GİRİŞ

Bu çalışmada , gri seviyeli resim yüklenip histogramı çizilecek, eşitleme uygulanacak ve eşitleme uygulanan resimin histogramı çizilecek,eşitlenen resime bicubic interpolasyon ile iki kat büyütme yapılacak ve histogramı çizilecek,büyütülen resime unsharp masking ve highboost filtering yapıp histogramı çizilecek ve sonucun genlik spektrumu çizilecek,görüntüye yarı çapı 200 olan gauss alçak geçiren filtre uygulaması yapılarak çizilecek görüntünün ters dft2sini alarak son hali çizilecek.

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.

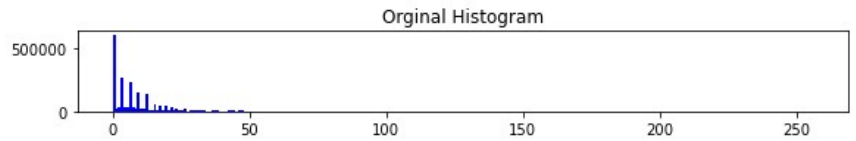
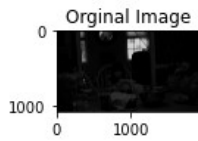
4. KODLAR

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

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

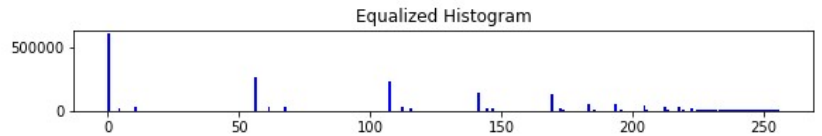
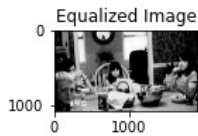
```
path = "C:/Users/Talip/Desktop/goruntu3.png"  
img = cv2.imread(path,0)
```

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



```
18 equ = cv2.equalizeHist(img)
```

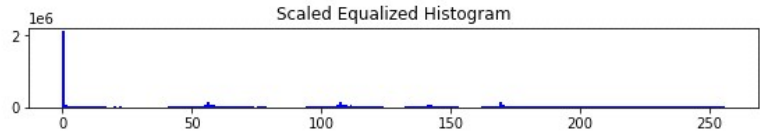
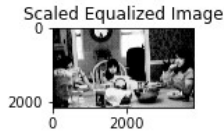
-Histogram eşitlemesi yapılarak kaynak resime eşitleme uygulandı.Histogram eşitleme belirli noktalarda kümelenen renk değerlerinin yol açtığı renk bozukluğunu gidermek için kullanılan bir tekniktir. Görüntünün verimli çalışması için imgenin gri tonlarında olması önemlidir. Eğer renkli bir görüntü ile çalışma yapılıyorsa görüntünün renk değerleri 3 ana renge ayrılır (kırmızı/mavi/yeşil).Ayrılan her renk kategorisi için ayrı ayrı histogram eşitleme tekniği uygulanır.



```
h,c = equ.shape  
  
scale_up_x = 2  
scale_up_y = 2  
  
scaled_f_up = cv2.resize(equ, None, fx= scale_up_x, fy= scale_up_y, interpolation= cv2.INTER_CUBIC)
```

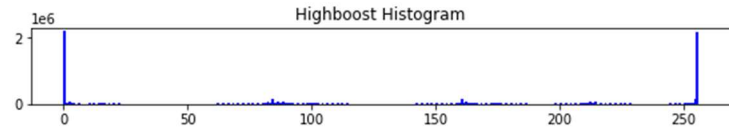
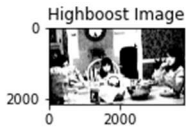
-Görüntüyü iki kat büyütmek için bikübik interpolasyon kullanıldı. Sayısal görüntü işlemede görüntüyü büyütme ya da küçültme için sıkça kullanılmaktadır. İnterpolasyon fonksiyonlarının önemli özelliği örneklenen veri ile interpolasyon noktalarında kesişme göstermeleridir. Bikübik interpolasyon dört nokta arasına üçüncü dereceden bir denklem kullanarak yüzey oturtmaya çalışır. Bu yüzey denklemi için 16 katsayıya ihtiyaç duyar ve bu değerlerin dördü köşelerin yoğunluk değerinden, sekizi yatay ve dikey türevlerinden, son dördü de köşegen türevlerinden elde edilir. Bilineer ve en yakın

konuşulukluinterpolasyona göre matematiksel olarak daha karmaşık olmasına rağmen daha az bozulmayla interpolasyon işlemini gerçekleştirir.



```
gauss = cv2.GaussianBlur(scaled_f_up, (7,7), 0)
unsharp_image = cv2.addWeighted(scaled_f_up, 2, gauss, -0.5, 0)
```

-Highboost yapabilmek adına önce resim blurlanmış ve blurlanan resim mevcut görüntüden çıkarılarak,sonuç mevcut resimle toplanmıştır.Böylece daha keskin bir görüntü elde etmiş oluruz.



```
row, col = unsharp_image.shape

I = cv2.dft(np.float32(unsharp_image),flags = cv2.DFT_COMPLEX_OUTPUT)
I_shift = np.fft.fftshift(I)

magnitude_spectrum = 20*np.log(cv2.magnitude(I_shift[:,0],I_shift[:,1]))

D0=200

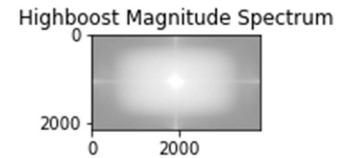
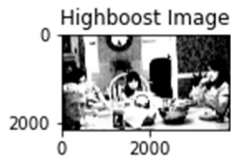
H=[math.exp(-((i-col/2)**2+(j-row/2)**2)/(2*D0**2)) for i in range(col) for j in range(row)]

If=np.zeros((row,col,2))

If[:,0]=I_shift[:,0]*H
If[:,1]=I_shift[:,1]*H

magnitude_spectrum1 = 20*np.log(cv2.magnitude(If[:,0],If[:,1]))
f_ishift = np.fft.ifftshift(If)
img_back = cv2.idft(f_ishift)
```

-Görüntünün genlik spektrumu,yarı çapı 200 olan gauss alçak geçiren filtre uygulaması ile gauss filtresi ve görüntünün ters dft2si bulunmuştur.





```
plt.rcParams["figure.figsize"] = [15, 7]
plt.rcParams["figure.autolayout"] = True

hist,bins = np.histogram(img.flatten(),256,[0,256])
plt.subplot(6, 2, 2).hist(img.flatten(),256,[0,256], color = 'b')
plt.title('Original Histogram')

hist,bins = np.histogram(equ.flatten(),256,[0,256])
plt.subplot(6, 2, 4).hist(equ.flatten(),256,[0,256], color = 'b')
plt.title("Equalized Histogram")

hist,bins = np.histogram(scaled_f_up.flatten(),256,[0,256])
plt.subplot(6, 2, 6).hist(scaled_f_up.flatten(),256,[0,256], color = 'b')
plt.title("Scaled Equalized Histogram")

hist,bins = np.histogram(unsharp_image.flatten(),256,[0,256])
plt.subplot(6, 2, 8).hist(unsharp_image.flatten(),256,[0,256], color = 'b')
plt.title("Highboost Histogram")

plt.subplot(6, 2, 1)
plt.title("Original Image")
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.subplot(6, 2, 3)
plt.title("Equalized Image")
plt.imshow(cv2.cvtColor(equ, cv2.COLOR_BGR2RGB))

plt.subplot(6, 2, 5)
plt.title("Scaled Equalized Image")
plt.imshow(cv2.cvtColor(scaled_f_up, cv2.COLOR_BGR2RGB))

plt.subplot(6, 2, 7)
plt.title("Highboost Image")
plt.imshow(cv2.cvtColor(unsharp_image, cv2.COLOR_BGR2RGB))

plt.subplot(6, 2, 9)
plt.title("Highboost Image")
plt.imshow(cv2.cvtColor(unsharp_image, cv2.COLOR_BGR2RGB))

plt.subplot(6,2,10)
plt.imshow(magnitude_spectrum, cmap = 'gray',vmin = 0, vmax = 255)
plt.title('Highboost Magnitude Spectrum')

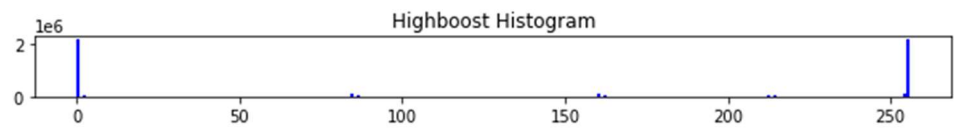
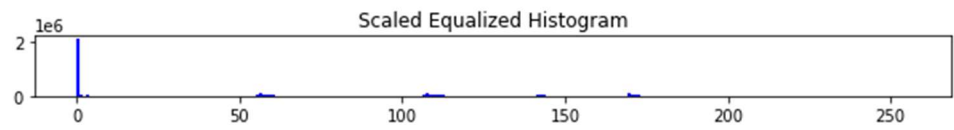
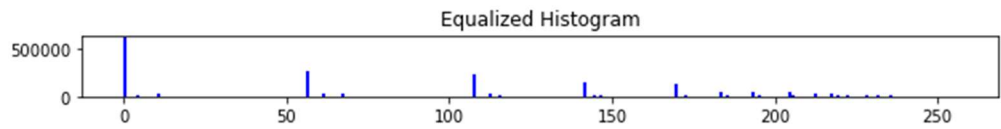
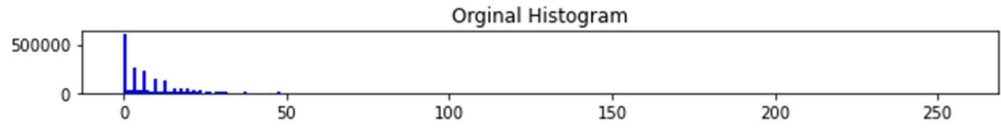
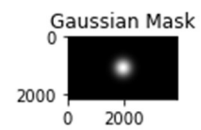
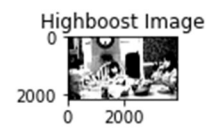
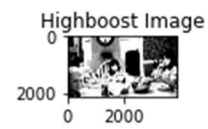
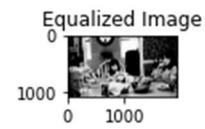
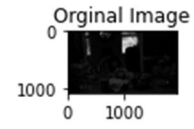
plt.subplot(6,2,11)
plt.imshow(H, cmap = 'gray')
plt.title('Gaussian Mask')

plt.subplot(6,2,12)
plt.imshow(img_back[:, :,0], cmap = 'gray')
plt.title('Gaussian Filtered Image')

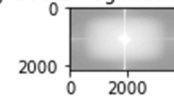
plt.show()
```

-İstenen sonuçlar tek bir plotta toplanmıştır.

5.SONUÇLAR



Highboost Magnitude Spectrum



Gaussian Filtered Image

