

BLM6103 – Olasılık, Rastgele Değişkenler ve Stokastik Süreçler

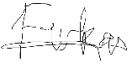
2023-2024 Güz Yarıyılı

Ödev – 1

İsim ve Soyisim: Recep Furkan Koçyiğit

Öğrenci Numarası:22501048

Tarih: 31.10.2023

İmza: 

- “Olasılık, R.D. ve Stokastik Prosesler” isimli Google Classroom grubundan güncel duyuruları takip etmek için ackaraca@yildiz.edu.tr mail adresine mail atabilirsiniz.
- Bu ödev, toplam 2 uygulamadan oluşmaktadır. Soru çözümlerini bu kapak sayfasının arkasına ekleyerek pdf haline dönüştürünüz. Tek bir pdf dosyası halinde mail yukarıdaki mail adresine gönderiniz.
- Dönem boyunca 3 ödev, 1 arasınav ve 1 final yapılacaktır. Bu sebeple, ödevler oldukça önemlidir. Eksiksiz ve kopyasız bir şekilde yapmanız önerilmektedir.
- Birbirinin tamamen aynısı olan çözümlerde hak edilen puan kişi sayısına bölünerek hesaplanacaktır. Bu sebeple, kendi çözümlerinizi paylaşmayınız.
- Puanların sorulara göre dağılımı [aşağıdaki](#) gibidir:

Problem	Gri tonlu görüntüde histogram eşitleme	Renkli Görüntüde Histogram Eşitleme	Toplam
Puanlar	85	15	100

- Bilgisayar üzerinden çözülecek sorular için kod ve ekran çıktıları paylaşılmalıdır. Derste MATLAB üzerinden kodlar paylaşılsa da siz ödevleri farklı programlama dillerinde yapabilirsiniz.
- Soruların tamamı diğer sayfada paylaşılmıştır.
- SON GÖNDERİM TARİHİ **2 KASIM 2023** OLUP ÖDEVLER GOOGLE CLASSROOM ÜZERİNDEN YÜKLENECEKTİR.

SORULAR

Açıklama: Kötü ışıklılık koşullarında çekilen görüntüleri daha anlaşılır ve yorumlanabilir bir hale getirmek için histogram eşitleme kullanılmaktadır. Bu işlem, derste gördüğümüz rastgele değişken dönüşümüne dayalı olup amaç görüntünün kümülatif dağılım fonksiyonunu (CDF) lineer hale getirmektedir. Bunu daha ayrıntılı bir şekilde derste anlatmıştık (*bknz.* Sunum 4).

Ödevde gri tonlu ve renkli olmak üzere ekte paylaşılan iki görüntünün histogramının eşitlenmesi istenmektedir. Rapor hazırlamadan önce sonuçlara bakarak görüntünün daha düzgün bir histograma sahip olduğundan mutlaka emin olunuz. Her iki görüntü için aşamaları aşağıdaki gibi ayrı ayrı raporlamanız gerekmektedir:

- Raporda giriş görüntüsünün orijinal hali ve histogramı paylaşılmalıdır. Histogram yorumlanmalıdır.
- Elde edilen histogram kullanarak CDF elde edilmeli ve raporda paylaşılmalıdır.
- Histogram eşitleme sonrasında lineer hale getirilen CDF raporda verilmeli ve yorumlanmalıdır.
- Son olarak, çıkış görüntüsü ve bu görüntünün histogramı raporlanmalıdır.

Tavsiyem, daha kolay olan gri tonlu görüntü için çalışan bir kod oluşturmanızdır. Çalıştığından emin olduktan sonra renkli görüntüde histogram eşitleme kısmına bakabilirsiniz.

Başarılar dilerim.

Dr. Öğr. Üyesi Ali Can KARACA

Histogram Eşitleme

BLM6103 – Olasılık, Rastgele Değişkenler ve Stokastik Süreçler

Recep Furkan Koçyiğit

22501048

furkan.kocyigit@std.yildiz.edu.tr

Bilgisayar Mühendisliği Bölümü

Elektrik Elektronik Fakültesi, Yıldız Teknik Üniversitesi

Özet

Bu ödevde, gri ve RGB olmak üzere 2 farklı kötü ışık altında çekilen resimler daha anlaşılır hale getirmek için Histogram Eşitleme yöntemi uygulanacaktır. Bunun için Kümülatif Dağılım Fonksiyonu (CDF) uygulanıp lineer hale getirilecektir. Sistemin gerçekleştirilmesi için programlama dili olarak Python kullanılmıştır.

Giriş

Ödev kapsamında aşağıdaki resimler kullanılmıştır:

1. Gri Tonlamalı Resim:



2. Renkli Resim:



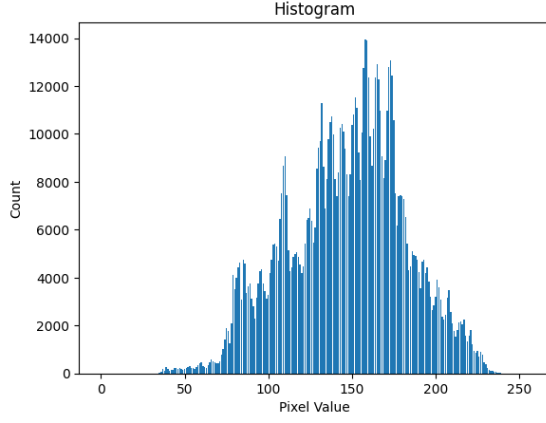
Yöntem

Bu sistem aşağıdaki ana modüllerden oluşmaktadır:

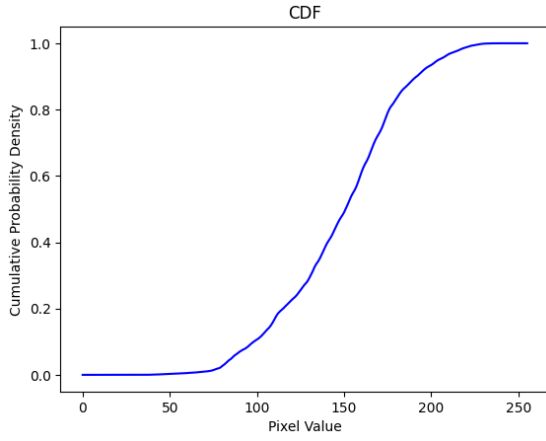
1. **Histogramın Oluşturulması:** Resme piksel verilerini dizi olarak alır ve onun 0-255 arası piksellerinin histogramını verir.
2. **Kümülatif Dağılım Fonksiyonu:** 0-255 piksel aralığında değerler tutan histogramın kümülatif olarak dağılımını 0-1 arasında verir.
3. **Resmin Kanallarına CDF işleminin Uygulanması:** Verilen kümülatif dağılım ve resmin kanal bilgisine göre yeni kanalı oluşturur.
4. **Histogramın Çizdirilmesi:** Verilen histogramın grafiğini oluşturur.
5. **Kümülatif Dağılımın Çizdirilmesi:** Verilen kümülatif dağılımın grafiğinin çizdirilmesi.
6. **Histogram Eşitleme:** Verilen dosya adı ve dosya türüne göre resmi okur, kanallarına ayırır ve her kanal için histogramını, kümülatif dağılımını oluşturup, lineer hale getirir. Daha sonrasında kanalları birleştirir ve dosyayı kaydeder.

Uygulama

Gri tonlamalı resmin histogramı aşağıdaki gibidir:



Histograma baktığımızda ortalaması 150 civarında ve normal dağılan bir yapıda olduğunu görüyoruz. Histogramın kümülatif dağılımı aşağıdaki gibidir:

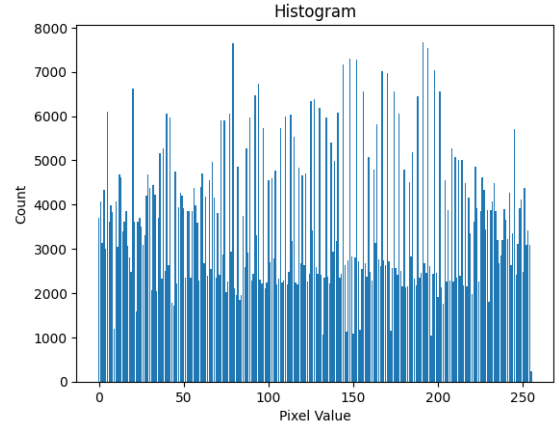


CDF' e baktığımızda ilk 40 piksele kadar herhangi bir piksel olmadığından sıfır olduğu görülmektedir. 100 ile 150 arasında yüksek artış olduğu 150 ile 200 arası ise yüksek düşüş olduğu CDF' e bakarak söylenebilir.

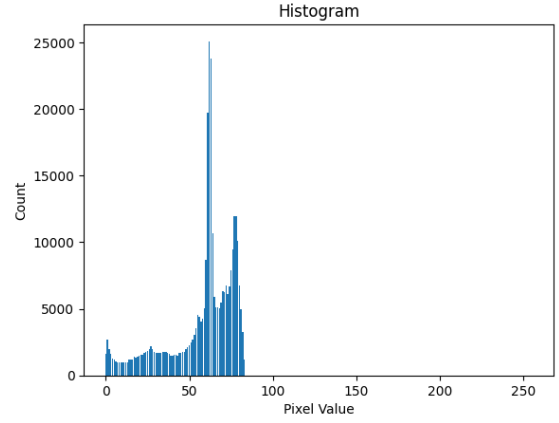
Histogram eşitleme işlemi uygulandıktan sonra çıktı olarak elde edilen resim şu şekildedir:



Histogram eşitleme işlemi uygulandıktan sonra histogramı ise şu şekildedir:

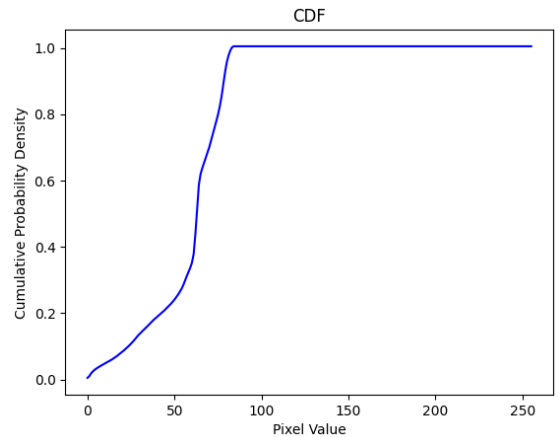


Renkli resmin kırmızı kanalı için histogram aşağıdaki gibidir:



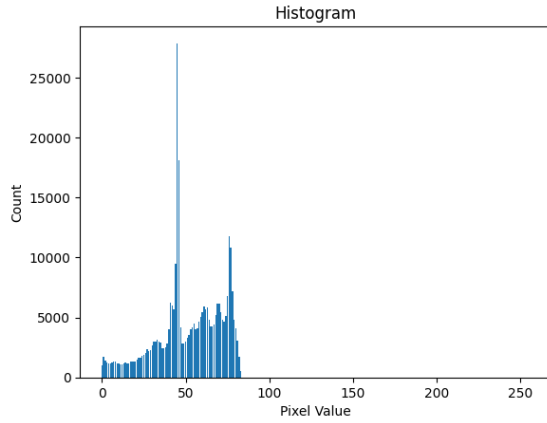
Bu histograma baktığımızda kırmızı renge ait kanaldaki piksel değerlerinin 0 ile 85 arasında olduğunu ve 60-65 ve 75-80 arasında yüksek sayıda aynı piksel değerine sahip pikseller olduğunu görülmektedir. Histogramın dağılımına baktığımızda sağ tarafa yatık olduğu söylenebilir.

Histogramın kümülatif dağılımı aşağıdaki gibidir:

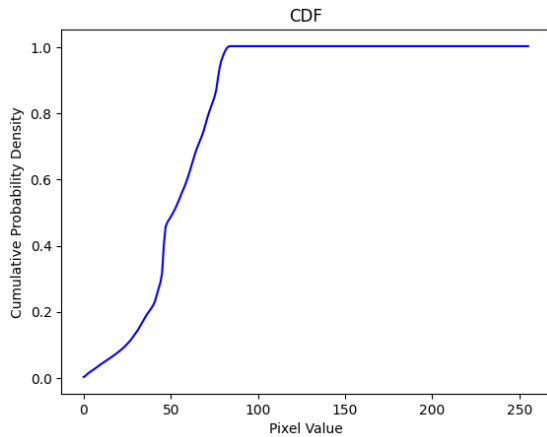


CDF' e baktığımızda 50-80 arası sert bir artış görülmektedir. 85' ten sonra ise CDF 1 olarak devam etmektedir. Bunun nedeni kümülatif olarak gittiği için 85' ten sonra başka bir değer olmadığı için artış olmamasıdır. Histogramın sağa yatık olmasını CDF' e bakarak söyleyebiliriz. Artış hızı 60 ve 75 gibi noktalarda daha fazla olmaktadır.

Yeşil kanala ait histogram aşağıdaki gibidir:

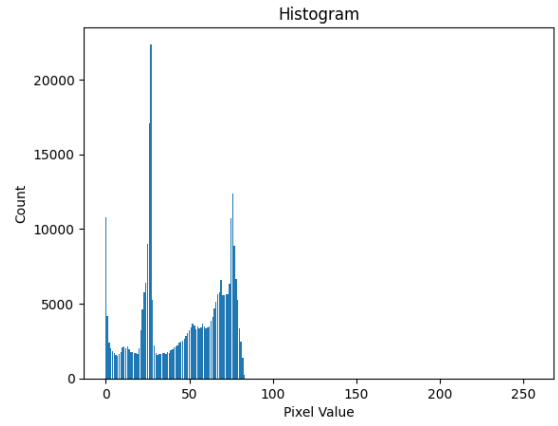


Bu histograma baktığımızda yeşil renge ait kanaldaki piksel değerlerinin 0 ile 85 arasında olduğunu ve 40-45 en çok sayıda piksel değerlerine ve 75-80 arasında yüksek sayıda aynı piksel değerine sahip pikseller olduğu görülmektedir. Histogramın kümülatif dağılımı aşağıdaki gibidir:

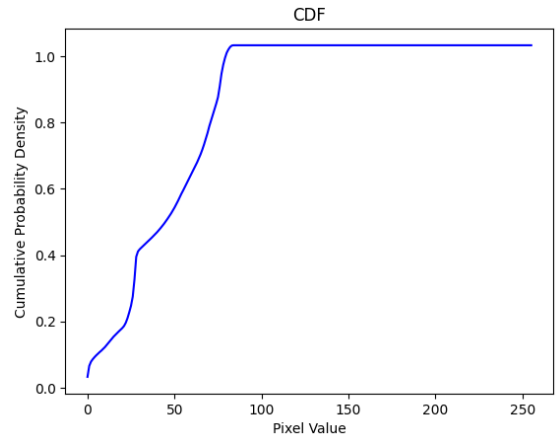


CDF' e baktığımızda 40-45 arası çok sert bir artış görülmektedir. 85' ten sonra ise CDF 1 olarak devam etmektedir. CDF' in 50' den 75' e kadar artış hızının azaldığı görülmektedir. 75-80 arası tekrar artış hızı yükselmiştir.

Mavi kanala ait histogram bilgileri aşağıdaki gibidir:

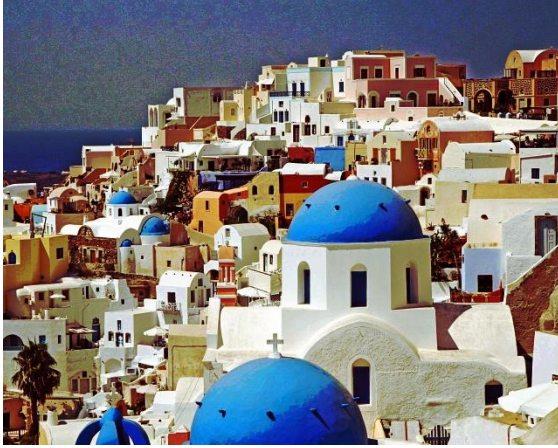


Bu histograma baktığımızda mavi renge ait kanaldaki piksel değerlerinin 0 ile 85 arasında olduğunu ve 30-35 en çok sayıda piksel değerlerine ve 0-5 ve 70-75 arasında yüksek sayıda aynı piksel değerine sahip pikseller olduğu görülmektedir. Histogramın kümülatif dağılımı aşağıdaki gibidir:

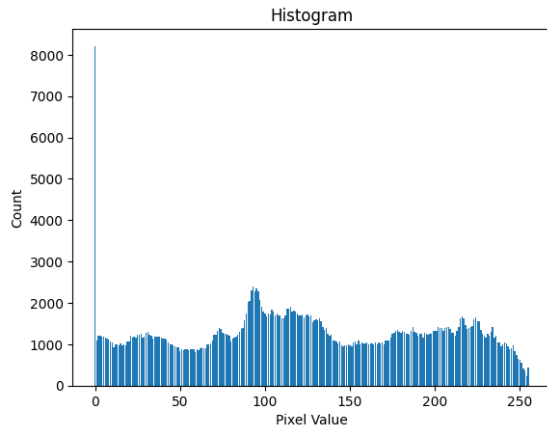


CDF' e baktığımızda 30-35 arası çok sert bir artış görülmektedir. 85' ten sonra ise CDF 1 olarak devam etmektedir. CDF' in 35' den 70' e kadar artış hızının azaldığı görülmektedir. 70-75 arası tekrar artış hızı yükselmiştir.

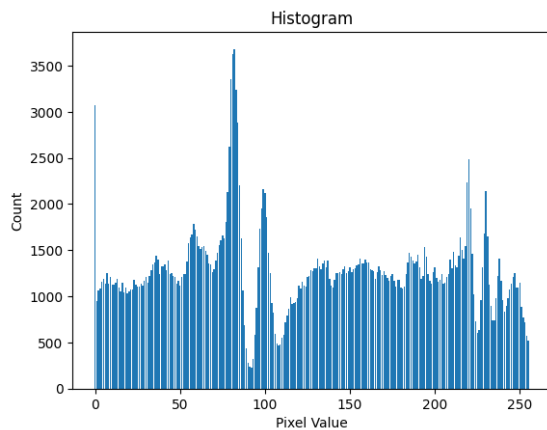
Histogram eşitleme işlemi uygulandıktan sonra çıktı olarak elde edilen resim şu şekildedir:



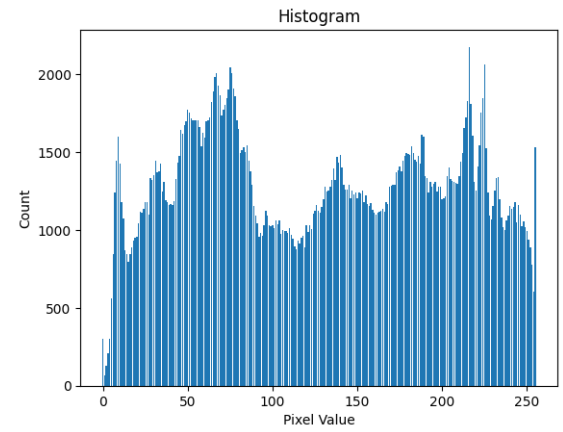
Histogram eşitleme işlemi uygulandıktan sonra resmin kırmızı kanalına ait histogram şu şekildedir:



Histogram eşitleme işlemi uygulandıktan sonra resmin yeşil kanalına ait histogram şu şekildedir:



Histogram eşitleme işlemi uygulandıktan sonra resmin mavi kanalına ait histogram şu şekildedir:



Sonuç

Bu çalışmada Histogram Eşitleme yöntemi kullanarak düşük ışık yoğunluğundaki resimler iyileştirilmiştir. Bu işlem için resimlerin kümülatif dağılımlarından faydalanılmıştır. Resimdeki değerlerin belirli değer aralığı içerisinde kümelenmesinden yararlanarak renk bozukluğu giderilmesi sağlanmıştır.

Kaynak Kod:

```
# %%
```

```
import cv2
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
# %%
```

```
def get_histogram(arr):
```

```
    """Bu metot array olarak verilen resmin histogramını çıkarır"""
```

```
    hist = [0] * 256
```

```
    for val in arr:
```

```
        hist[val] += 1
```

```
    return hist
```

```
# %%
```

```
def cumulative_distribution(hist):
```

```
    """Bu metot histogramı verilen resmin kümülatif
    dağılımını verir"""
```

```
    # cdf i tutacak array
```

```
    arr = [0] * 256
```

```
    # toplam piksel sayısı
```

```
    s = sum(hist)
```

```
    # histogram için kümülatif olarak artacak değer
```

```
    c = hist[0]
```

```
    arr[0] = c / s
```

```
    for i in range(1, 256):
```

```
        c += hist[i-1]
```

```
        arr[i] = c / s
```

```
    return arr
```

```
# %%
```

```
def get_equalized_channel(cdf, channel):
```

```
    """Bu metot verilen resmi cdf ine göre channeli
    eşitler"""
```

```
    newChannel = []
```

```
    for row in channel:
```

```
        newRow = []
```

```
        for pixel in row:
```

```
            newRow.append(int(cdf[pixel] * 255 / cdf[-
1]))
```

```
        newChannel.append(newRow)
```

```
    return newChannel
```

```
# %%
```

```
def plot_hist(hist):
```

```
    """Bu metot histogramı çizdirir"""
```

```
    plt.bar(range(256), hist)
```

```
    plt.xlabel("Pixel Value")
```

```
    plt.ylabel("Count")
```

```
    plt.title("Histogram")
```

```
    plt.show()
```

```
# %%
```

```
def plot_cdf(cdf):
```

```
    """Bu metot kümülatif dağılımını çizdirir"""
```

```
    plt.plot(cdf, color="b")
```

```
    plt.xlabel("Pixel Value")
```

```
    plt.ylabel("Cumulative Probability Density")
```

```
    plt.title("CDF")
```

```
    plt.show()
```

```
# %%
```

```
def histogram_equalization(filename, image_type):
```

```
    """Bu metot histogram eşitleme işlemini
    yapar"""
```

```
    image = cv2.imread(filename, image_type)
```

```
    dot_index = filename.find('.')
```

```
    channels = cv2.split(image)
```

```
    equalized_channels = []
```

```
    for channel in channels:
```

```
        hist = get_histogram(channel.flatten())
```

```
        plot_hist(hist)
```

```

    cdf = cumulative_distribution(hist)

    print(cdf)

    plot_cdf(cdf)

equalized_channels.append(np.asarray(get_equalized_channel(cdf, channel)))

equalized = cv2.merge(equalized_channels)

cv2.imwrite(filename[:dot_index] +
"HistogramEqualized.jpg"
            , equalized)

# %%

histogram_equalization("GrayImage.jpeg",
cv2.IMREAD_GRAYSCALE)

# %%

histogram_equalization("ColorImage.tif",
cv2.IMREAD_COLOR)

```