Melih Sami Özaydın 30.10.2021
502131003

BM 5121 Görüntü İşleme ve Uygulamaları

Ödev 2:

2.1)

8 bit bir resmi, TH=127 ile, ortalama alarak ve medyan alarak siyah beyaza çevirip gösterin. (bknz aşağıdaki şekil 2.1).

2.2)

8 bit bir resmi şekilde gösterildiği gibi orjinal, gamma 2 ve gamma 0.5 ile dönüşmüş versiyonlarını ve karşılık gelen histogramlarını gösteriniz. (bknz aşağıdaki şekil 2.2).

2.3) (BONUS artı 10 puan):

Dar kontrastlı bir imgeyi derste anlatılan lineer dönüşümie kontrastını arttırın. Histogram ve imgelerin önceki ve sonraki hallerini gösterin (bknz aşağıdaki şekil 2.3).

Yorum:

2.1)

- Eşikleme, bir görüntünün ikili hale getirilmesidir. Genel olarak, gri tonlamalı bir görüntüyü, piksellerin 0 veya 255 olduğu ikili bir görüntüye dönüştürmeye çalışıyoruz.

- Basit bir eşikleme örneği, bir T eşik değeri seçmek ve ardından T'den küçük tüm piksel yoğunluklarını 0'a ve T'den büyük tüm piksel değerlerini 255'e ayarlamak olabilir. Bu şekilde, görüntünün ikili bir temsilini oluşturabiliriz.
- Buradaki gerçeklemede t değeri 127, resim değer ortalaması ve resim değer medyanı olarak belirlenerek 3 farklı şekilde gösterildi.

2.2)

- İnsan gözü koyu tonlardaki değişikliklere açık tonlardan çok daha duyarlıdır. Bunu hesaba katmak için, gama düzeltmesini uygulayabiliriz.
- Gama düzeltmesi, Güç Yasası Dönüşümü (Power Law Transform) olarak da bilinir. İlk olarak, görüntü piksel yoğunluğumuz [0, 255] ila [0, 1.0] aralığında ölçeklendirilmelidir.

- Buradan, aşağıdaki denklemi uygulayarak çıktı gama düzeltilmiş görüntümüzü elde ederiz:
  O = I ^ (1 / G)

- I giriş resmimiz ve G gama değerimizdir. Çıktı görüntüsü O daha sonra [0, 255] aralığına geri ölçeklenir.

- < 1 gama değerleri görüntüyü spektrumun daha koyu ucuna doğru kaydırırken, > 1 gama değerleri görüntünün daha açık görünmesini sağlar. G=1 gama değerinin giriş görüntüsü üzerinde hiçbir etkisi olmayacaktır:

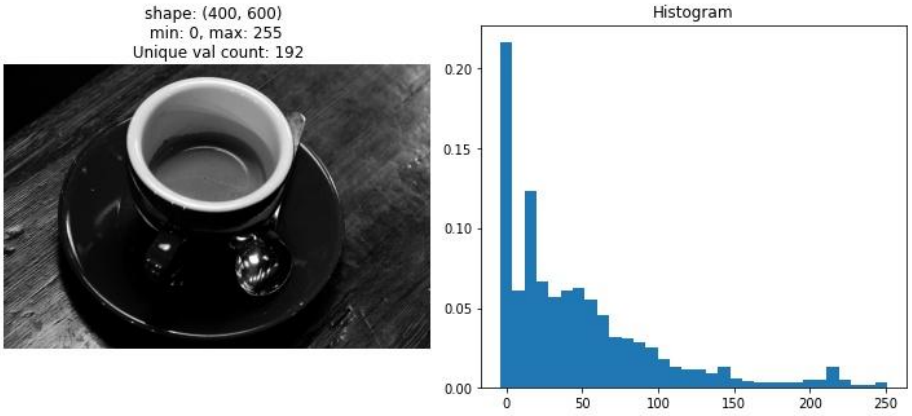
2.3) (BONUS artı 10 puan):

- Kontrastın bütün piksel değer aralığını kullanacak şakilde düzenlenmesini sağlayan bir operasyondur.
- Bu Operasyon `g = (f-f.min())(f.max()-f.min())` formülü baz alınarak gerçeklenmiştir.
- İmgenin kontrastını normalize edere daha iyi bir görsel ortaya çıkmasını sağlar.

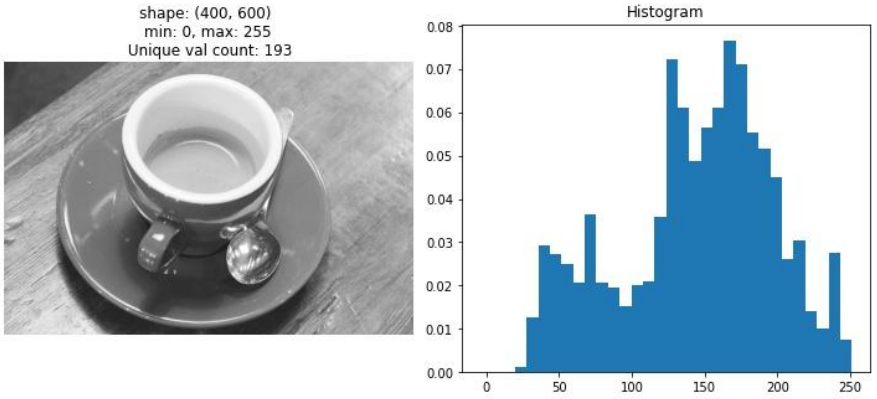
Program Çıktısı:
2.1)

2.2)

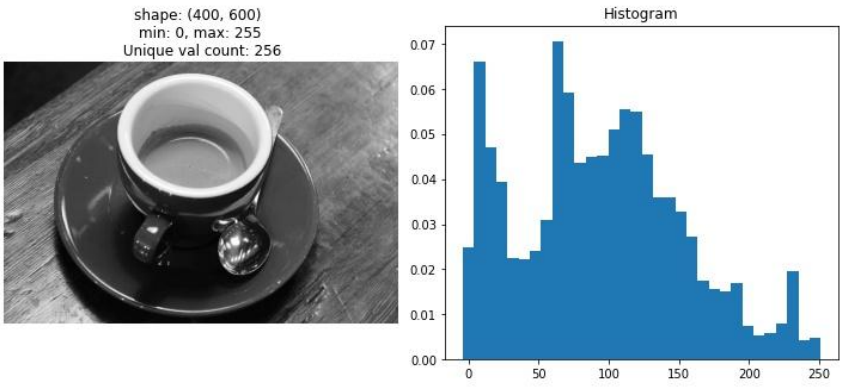
Gamma: 0.5



Gamma: 2.0



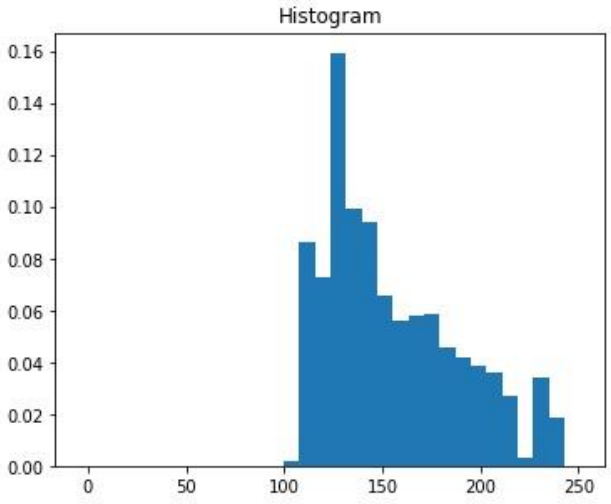
Original



2.3) (BONUS artı 10 puan):

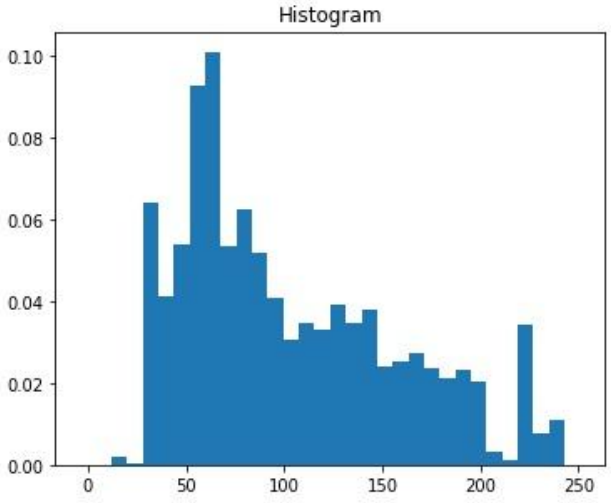
Original

shape: (511, 511)
min: 93, max: 250
Unique val count: 145



Contrast Stretched

shape: (511, 511)
min: 0, max: 255
Unique val count: 145



Program Kaynak Kodları:

```
import numpy as np
import os
import matplotlib.pyplot as plt
from skimage import data
from skimage import io
from skimage.color import rgb2gray
```

2.1)

```
def threshold_img(img, method="mean"):
    if isinstance(method, int):
        th_value = method
    elif method == "mean":
        th_value = img.mean()
    elif method == "median":
        th_value = np.median(img)

    return (img > th_value)*255

img = (rgb2gray(data.astronaut()) * 255).astype(np.uint8)
plt.imshow(threshold_img(img, method=127), cmap="gray")
```

```
test_images = [data.astronaut(), data.coffee(), data.chelsea()]

methods = ["mean", "median", 127]
for img in test_images:
    img = (rgb2gray(img) * 255).astype(np.uint8)
    show_img(img,
             bold_title="Original",
             save=True, save_dir="./2.1",
             cmap="gray")

    for method in methods:
        out_img = threshold_img(img, method=method)
        show_img(out_img, bold_title=f"Threshold Method: {method}",
                 save=True, save_dir="./2.1",
                 cmap="gray")
```

2.2)

```
def adjust_gamma(image, gamma=1.0):
    # build a lookup table mapping the pixel values [0, 255] to
    # their adjusted gamma values
    invGamma = 1.0 / gamma
    table = np.array([((i / 255.0) ** invGamma) * 255
        for i in np.arange(0, 256)]).astype("uint8")
    # apply gamma correction using the lookup table
    return np.take(table, image)
```

```
img = (rgb2gray(data.coffee()) * 255).astype(np.uint8)
gammas = [0.5, 2.0]

show_img(img, cmap="gray",
         bold_title="Original",
        save=True, save_dir="./2.2",
        histogram=True)
for gamma in gammas:
    img_gamma = adjust_gamma(img, gamma=gamma)
    show_img(img_gamma, cmap="gray",
             bold_title=f"Gamma: {gamma}",
             save=True, save_dir="./2.2",
             histogram=True)
```

2.3) (BONUS artı 10 puan):

```
def contrast_stretch(img):
    out = (img-img.min())/(img.max()-img.min())
    return ((out) * 255).astype(np.uint8)
```

```
img_url = "https://wiki.cuvilib.com/images/3/36/Eqbefore.jpg"
img = img = io.imread(img_url)
print(img.shape)
img = (rgb2gray(img) * 255).astype(np.uint8)

img_contrast_stretched = contrast_stretch(img)

show_img(img, cmap="gray",
         bold_title="Original",
        save=True, save_dir="./2.3",
        histogram=True)

show_img(img_contrast_stretched, cmap="gray",
         bold_title=f"Contrast Stretched",
         save=True, save_dir="./2.3",
         histogram=True)
```