

### **BM 5121 Görüntü İşleme ve Uygulamaları**

#### **Ödev 1:**

8 bit gri bir resmi açan ve 7,6,5...3,2,1 bit ile kodlayan programı yazınız.

Program çıktılarını raporlayıp, yorumlayınız. kodlama yaptıktan sonra sonucu gösterebilmek için kuantalama işleminin terslenmesi ve resmin tekrar 0-255 arasına döndürülmesi gerekmektedir. Buna dikkat ediniz. Raporadaki sonuçları aşağıdaki görselde belirtildiği şekilde gösterebilirsiniz.

#### **Yorum:**

RGB formatında uint8 kodlaması ile okunan görsel, öncelikle float64 veri tipinde numpy matrisi olacak şekilde 255 'e bölündü.

Bu 0-1 aralığında float64 veri tipine sahip matris istenilen bit sayısının sağladığı üst değer ile tekrar çarpılarak doğru piksel değer aralığına getirildi.

Daha sonra tekrar uint8 formatına dönüştürülerek, istenilen formatta kodlanmış oldu. Bu veri tipi operasyonu esnasında değerlerin en yakın tam sayıya yuvarlanması ile piksel verisinde bir kayıp meydana gelmektedir.

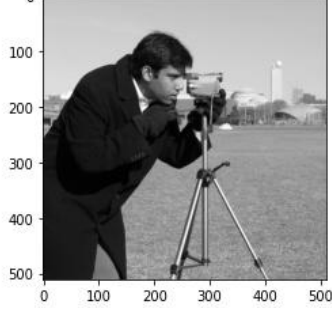
Görselleştirme için piksel değerleri 0-1 aralığına yeni bit sayısının sağladığı maksimum değere bölünerek ve ardından 0-255 arasına 255 ile çarpım yapılarak döndürüldü.

Bu sonuçlar görselleştirilerek sunuldu.

## Program Çıktısı:

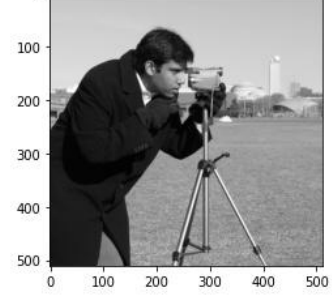
**7-bit Kuantizasyon**

shape:(512, 512), Min\_value:0, Max\_value:255, unique\_pixel\_values:128



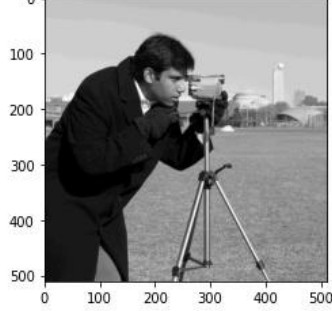
**Orjinal GörSEL**

shape:(512, 512), Min\_value:0, Max\_value:255, unique\_pixel\_values:256



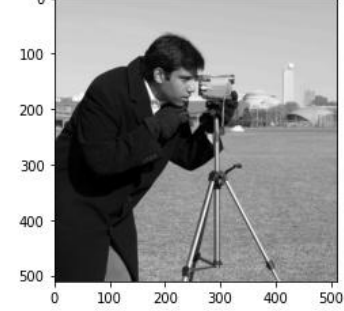
**5-bit Kuantizasyon**

shape:(512, 512), Min\_value:0, Max\_value:255, unique\_pixel\_values:32



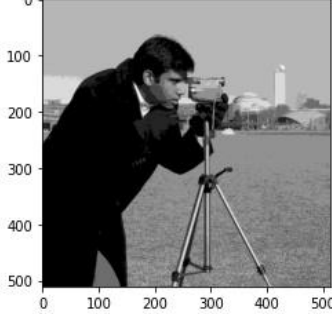
**6-bit Kuantizasyon**

shape:(512, 512), Min\_value:0, Max\_value:255, unique\_pixel\_values:64



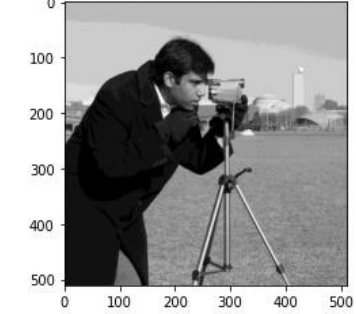
**3-bit Kuantizasyon**

shape:(512, 512), Min\_value:0, Max\_value:255, unique\_pixel\_values:8



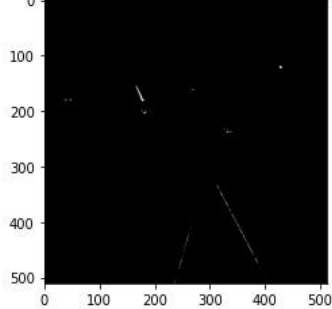
**4-bit Kuantizasyon**

shape:(512, 512), Min\_value:0, Max\_value:255, unique\_pixel\_values:16



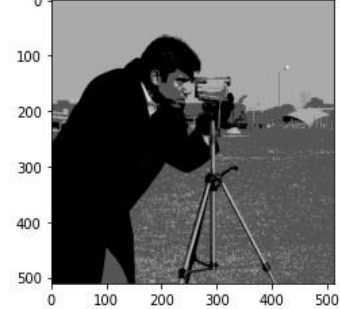
**1-bit Kuantizasyon**

shape:(512, 512), Min\_value:0, Max\_value:255, unique\_pixel\_values:2



**2-bit Kuantizasyon**

shape:(512, 512), Min\_value:0, Max\_value:255, unique\_pixel\_values:4



## Program Kaynak Kodları:

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

```
def quantize_image_colors(img_array, bit=1, verbose=False):

    if verbose:
        print("input: ")
        print(img_array.dtype, img_array.shape)
        print(f"{len(np.unique(img_array))} values. Max: {img_array.max()}, Min: {img_array.min()}", )

        print(f"\nQuantizing image to {bit} bits.")

    # divide by 255
    img_array = img_array.astype(np.uint8)
    binary_img_array_float = img_array / 255 # (2**8)-1

    # multiply new max
    quantized_img_array = binary_img_array_float * ((2**bit)-1)

    # Cast to uint8 and return to 0-1 Float64
    quantized_img_array = quantized_img_array.astype(np.uint8) / ((2**bit)-1)

    # return back to 8 bit 0-255
    quantized_img_array = quantized_img_array * ((2**8)-1)

    out = np.asarray(quantized_img_array, dtype=np.uint8)
    if verbose:
        print("\nout: ")
        print(out.dtype, out.shape)
        print(f"{len(np.unique(out))} values. Max: {out.max()}, Min: {out.min()}", )
        print(np.unique(out))

    return out
```

```
def show_img(img, cmap=None, save=False, title="Title", bold_title="Bold Title"):

    fig = plt.figure()
    ax = fig.add_subplot()
    fig.subplots_adjust(top=0.85)

    # Set titles for the figure and the subplot respectively
    fig.suptitle(bold_title, fontsize=14, fontweight='bold')
    ax.set_title(title)

    ax.imshow(img, cmap=cmap)
    if save:
        plt.savefig(f"{bold_title}.jpg")
    plt.show()
```

```
def test(fn, img=None, gray=False):

    if img is None:
        img = io.imread('https://i.stack.imgur.com/QCl8D.jpg')

    if gray:
        img = rgb2gray(img)
    if img.max() <= 1:
        img = img*255

    img = np.asarray(img, np.uint8)
    show_img(img,
              cmap="gray",
              title=f"shape:{img.shape}, Min_value:{img.min()}, Max_value:{img.max()},
unique_pixel_values:{len(np.unique(img))}",
              bold_title=f"Orjinal Görşel")

    for i in reversed(range(1, 8)):
        out=fn(img, bit=i)

        show_img(out,
                  save=True,
                  cmap="gray",
                  title=f"shape:{out.shape}, Min_value:{out.min()}, Max_value:{out.max()},
unique_pixel_values:{len(np.unique(out))}",
                  bold_title=f"{i}-bit Kuantizasyon")

        print()
        #io.imshow(out, cmap="gray")
        #plt.show()

test(img=data.camera(),fn=quantize_image_colors, gray=True)
```