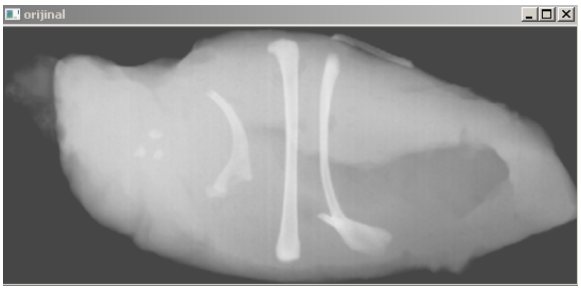
9. Hafta uygulama ödevi

1) Aşağıdaki formatta yeğinlikdilimleme amaçlı bir m file yazın uygulayın.

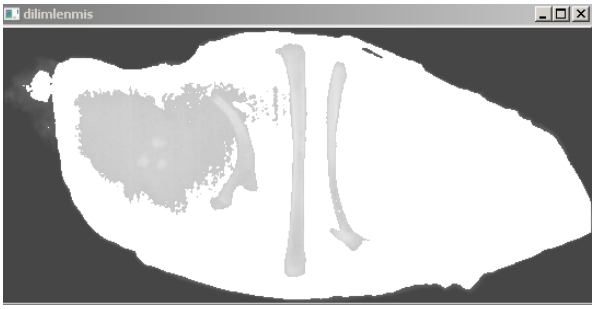
J=yeginlikdilimle(I,100,200,'vurgula');J=yeginlikdilimle(I,100,200,"segment");

```
import cv2
import numpy as np

def yeginlikdilimle(I,lp,hp,metod):
    resim=cv2.imread(I,0)
    dilimlenmis=resim
    if(lp<hp):
        x,y=resim.shape
        cv2.imshow("orijinal",resim)
        for i in range(x-1):
            for j in range(y-1):
                if(metod=="segment"):
                    if(dilimlenmis[i,j]>=lp
and dilimlenmis[i,j]<=hp):
                        dilimlenmis[i,j]=255
                    else:
                        dilimlenmis[i,j]=0
                elif(metod == "vurgula"):
                    if(dilimlenmis[i,j]>=lp
and dilimlenmis[i,j]<=hp):
                        dilimlenmis[i,j]=255

cv2.imshow("dilimlenmis",dilimlenmis)

    else:
        print("geçerli aralık girin.")
#numpy kutuphanesinde yaptığım if
işlemlerinin yazılmışları vardı fakat
kullanmak istemedim.
yeginlikdilimle("meat.png",100,200,"vurgula"
)

cv2.waitKey(0)
```

orijinal



vurgula

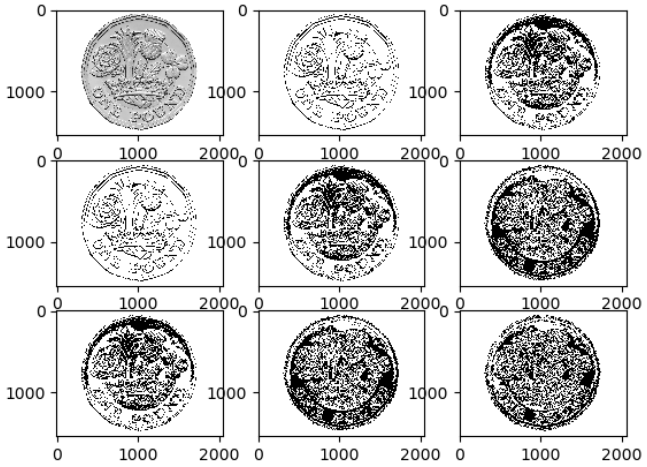


segment

2) J=bitdilimle(I); %siyah beyaz imge

Bit katmanlarına ayrılmış imge

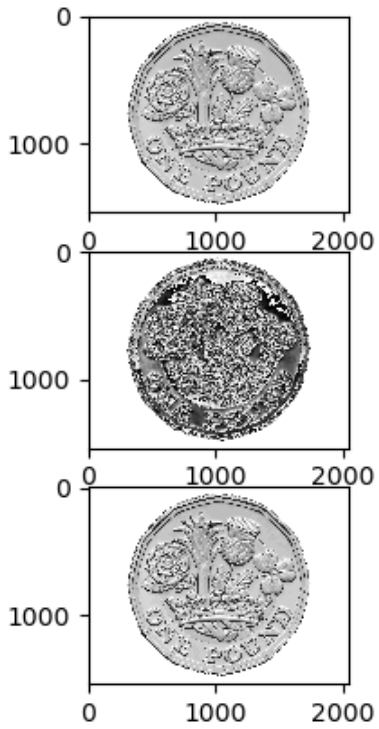
```
import cv2
import numpy as np
import matplotlib.pyplot as plt
def bitdilimle(foto):
    I=cv2.imread(foto,0)
    #imgenin binary şeklinde çıktısı alınması
için sekiz bitlik bir goruntu için bu işlemleri
yaparız
    sekizinci=np.bitwise_and(I,128)
#[1v0,*,*,*,*,*,*,*] dönüş 1 veya 0 olur

yedinci=np.bitwise_and(I,64)#[*,1v0,*,*,*,*,*,*]
dönüş 1 veya 0 olur

altinci=np.bitwise_and(I,32)#[*,*,1v0,*,*,*,*,*]
dönüş 1 veya 0 olur
    besinci=np.bitwise_and(I,16)
    dorduncu=np.bitwise_and(I,8)
    ucuncu=np.bitwise_and(I,4)
    ikinci=np.bitwise_and(I,2)
    birinci=np.bitwise_and(I,1)



katmanlar=[I,sekizinci,yedinci,altinci,besinci,d
orduncu,ucuncu,ikinci,birinci]
    f,a=plt.subplots(3,3)
    for i in range(3):
        for j in range(3):

a[i][j].imshow(katmanlar[i+j],cmap="gray")

    plt.show()
    cv2.waitKey(0)
bitdilimle("new-pound.jpg")
```



2) J=bitdilimle(I); %siyah beyaz imge

Bit katmanlarına ayrılmış imge

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

I=cv2.imread("new-pound.jpg",0)
#imgenin binary şeklinde çıktısı alınması için
sekiz bitlik bir goruntu için bu işlemleri
yaparız
sekizinci=np.bitwise_and(I,128)
#[1v0,*,*,*,*,*,*,*] dönüş 1 veya 0 olur
yedinci=np.bitwise_and(I,64)#[*,1v0,*,*,*,*,*,*]
dönüş 1 veya 0 olur
altinci=np.bitwise_and(I,32)#[*,*,1v0,*,*,*,*,*]
dönüş 1 veya 0 olur
besinci=np.bitwise_and(I,16)
dorduncu=np.bitwise_and(I,8)
ucuncu=np.bitwise_and(I,4)
ikinci=np.bitwise_and(I,2)
birinci=np.bitwise_and(I,1)

alttoplam=birinci+ikinci+ucuncu+dorduncu+besinci
usttoplam=sekizinci+yedinci+altinci+besinci+dord
uncu
f,a=plt.subplots(3,1)
a[0].imshow(I,cmap="gray")
a[1].imshow(alttoplam,cmap="gray")
a[2].imshow(usttoplam,cmap="gray")

plt.show()
cv2.waitKey(0)
```



orijinal, alttan beş katman toplam, ustten beş katman toplam

3) Histogram Denkleştirme

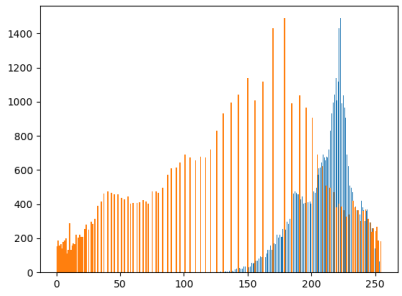
```
import cv2
import numpy as np
import matplotlib.pyplot as plt

img=cv2.imread("gamma1.jpg",0)
cv2.imshow("orijinal",img)
hist,bins=np.histogram(img,bins=256,rang
e=(0,256))#8 bitlik(rk) görüntüde
yeğinlik-sıklık histogram
değerlerini(nk) dizi halinde alıyoruz
normal_hist=hist/img.size
#normalleştirilmiş histogram/olasılık
degeri pr(rk)=nk/MN size komutu x*y
dondurur
birikimli=np.cumsum(normal_hist)#
Birikimli dağılım fonx uygulanmış hali
donusum=(256-1)*birikimli
#(L-1)*birikimli
x,y=img.shape
ravel=img.ravel()#dizini 2 boyuttan tek
boyuta düşürüyoruz
histfoto=ravel #histogram denklestirme
yapmak icin fotografı kopyalıyorum

for i,pixel in enumerate(ravel): #
    histfoto[i]=donusum[pixel]#
donusumde sum of k degeri icin k ya
kadar olan toplamın degerinin, donusume
indis olarak gönderilip ordaki yeğinlik
değerini yeni fotoğrafın i. indisine
atanır
foto=histfoto.reshape(x,y).astype(np.uin
t8)# fotograf 8 bit olarak tekrar 2
boyuta dönüdürlür
equ= cv2.equalizeHist(img)#in-built fonx

cv2.imshow("manual",foto)
cv2.imshow("inbuilt",equ)
```

4)Histogram Denkleştirme (Built in)

orijinal

manual

inbuilt



`cv2.waitKey(0)`	