**TEMELLER**

import cv2 #OpenCV 'yi dosyasını yüklicek  
import numpy as np #Numpy dosyasını yüklicek as diyerek numpy diye çağırmak yerine np yazıp çağıracağız  
  
image=cv2.imread("I.E.1.png",0) #cv2 dosyasınından aldığımız imread fonksiyonu ile resmimizi okuyoruz  
 #resmi yazdıktan sonra 0 koyarsanız resim renksiz biçimde yazdırılacak

cv2.imshow("Denizhan Medikal",image) #cv2 dosyasından aldığımız imshow ile resmi gösteriyoruz

cv2.imwrite("I.E.1-Black.png",image) #imwrite fonksiyonu ile siyaha çevirdiğimiz resmi kaydediyoruz  
 #ilk başa resme vermek istediğimiz adı daha sonra göstermek istediğimiz resmi yazıyoruz

cv2.waitKey(0) #cv2 dosyasından aldığımız waitKey yardımıyla o resmi ekranda kalmasını sağlıyoruz  
 #Eğer waitKey fonksiyonunun içerisine 0 yazmaz iseniz resim kaybolacaktır

cv2.destroyAllWindows() #destroyAllWimdows fonksiyonu sayesinde de penceremizi kapatıyoruz

print(image) #eğer resmi printle yazdırsak her bir pixelin aldığı değeri yazdırır  
print(image.size) #resmin boyutunu  
print(image.dtype) #resmin bir pikselde ne kadar yer kapladığını  
print(image.shape) #(genişlik,yükseklik,renk)olduğunu belirtir  
print(image[(78),(200)]) #belirtiğim koordinatlardaki pikselin renk değerini bize yazdırır

**EFEKT UYGULAMA**

from cv2 import\*  
from numpy import\*  
  
image2=imread("image-2.jpg")  
#image2[:,:,0]=30 #resme komple mavi efek uygular  
#image2[:,:,1]=155 #resme komple yeşil efek uygular  
#image2[:,:,2]=200 #resme komple kırmızı efek uygular  
image2[150:300,100:500,2]=150 #image[yükseklik,genişlik,renk]=ton  
kesit=image2[150:300,100:500] #bu koordinatlardaki alanı alıp kesit değişkenine atar  
 #böylelikle resimden bir kesit almış oluruz  
image2[0:150,0:400]=kesit #kesilen kesiti resmin 0:15,0:400 aralığına yapıştırdı  
# imshow("kesit",kesit)  
imshow("manzara",image2)  
waitKey(0)  
destroyAllWindows()

**RESMİ GRİLEŞTİRME**

from cv2 import\*  
from numpy import\*  
  
image1=imread("wp4.jpg")  
imagegray=cvtColor(image1,COLOR\_BGRA2GRAY) #resmi grileştiriyoruz  
yukseklik,genislik,kanalsayisi=image1.shape #image1 değişkeninin boyutlarını değişkenlere atıyoruz  
yukseklik,genislik=imagegray.shape  
print("orjinal",yukseklik,genislik,kanalsayisi)  
print("gray",yukseklik,genislik)  
  
imshow("orji",image1)  
imshow("gray",imagegray)  
  
waitKey(0)  
destroyAllWindows()

**FİLİTRELEME**

from cv2 import\*  
from numpy import\*  
  
image1=imread("blur.jpg")  
#MEAN FİLTER  
meanFilter=blur(image1,(3,3)) #3-3 bir pixel içeriisndeki sayıların  
meanFilter2=blur(image1,(5,5))  
meanFilter3=blur(image1,(51,51))  
#medianfilter  
medianFilter=medianBlur(image1,5)  
#gaussFilter  
gausFilter=GaussianBlur(image1,(9,9),0)  
  
imshow("image",image1)  
imshow("meanFilter",meanFilter)  
imshow("meanFilter2",meanFilter2)  
imshow("meanFilter3",meanFilter3)  
imshow("Gausfilter",gausFilter)  
imshow("medianFilter",medianFilter)  
  
waitKey(0)  
destroyAllWindows()

**AĞIRLIKLI TOPLAMA**

from cv2 import\*  
from numpy import\*  
  
image5=imread("image-2.jpg")  
image6=imread("image3.jpg")  
  
print(image5[300,400]) #iki resimden de seçtiğimiz koordinatlardaki renkleri alıyoruz ve ekrana yazdırıyoruz  
print(image6[300,400])  
imshow("toplama",image5)  
imshow("toplama2",image6)  
print(image5[300,400]+image6[300,400]) #iki resimden gelen renk scalelerini toplayıp ekrana yazdııyoruz  
  
waitKey(0)  
destroyAllWindows()

**AĞIRLIKLI TOPLAMA 2**

from cv2 import\*  
from numpy import\*  
  
#iki resim aynı boyutta olmak zorunda  
image10=imread("wp4.jpg")  
image11=imread("wp2.jpg")  
  
imshow("wp1",image10)  
imshow("wp2",image11)  
  
toplam=add(image10,image11) #bir değişkene birleştirip 2 resmi atıyoruz  
toplam2=addWeighted(image10,0.9,image11,0.1,0) #2 resim değişkenlerinin birleştiklerinde ki oranı yazıyoruz  
imshow("toplam2",toplam2)  
imshow("toplama",toplam)  
waitKey(0)  
destroyAllWindows()

**AYNALAMA,UZATMA,TEKRARLAMA,ÇERÇEVELEME**

from cv2 import\*  
from numpy import\*  
  
image3=imread("image-2.jpg")  
aynalama=copyMakeBorder(image3,500,500,500,500,BORDER\_REFLECT) #aynalama efekti  
uzatma=copyMakeBorder(image3,100,100,100,100,BORDER\_REPLICATE) #uzaatma efekti  
tekrarlama=copyMakeBorder(image3,100,100,100,100,BORDER\_WRAP) #tekrarlama efekti  
cerceve=copyMakeBorder(image3,100,100,100,100,BORDER\_CONSTANT,value=(70,120,255)) #çerçeve oluşturma efekti  
#eğer son value değeri girilmez ise çerçeveyi siyah yazdırır  
imshow("aynalama",aynalama)  
imshow("uzatma",uzatma)  
imshow("tekrar",tekrarlama)  
imshow("border",cerceve)  
  
waitKey(0)  
destroyAllWindows()

**NOKTA,ÇİZGİ**

from cv2 import\*  
from numpy import\*  
  
image=imread("image-2.jpg")  
image[50,30]=[255,255,255] #seçtiğimiz kordinatlara seçtiğimiz renkte bir nokta oluşturuyor  
for i in range(100): #bir döngü içerisine veridiğimiz 100 değeri ile çizgi oluşturuyoruz  
 image[100,i] = [255, 255, 255] #  
 for i in range(100):  
 image[i, 100] = [255, 255, 255]  
imshow("Manzara",image)  
waitKey(0)  
destroyAllWindows()

**ÇİZGİ,DAİRE,METİN**

from cv2 import\*  
from numpy import\*  
  
image1=zeros((300,300,3),"uint8")  
  
line(image1,(0,0),(100,100),(25,25,255),4) #çizgi yazma  
line(image1,(300,0),(200,200),(25,25,255),4)  
line(image1,(0,300),(100,100),(25,25,255),4)  
line(image1,(300,300),(200,200),(25,25,255),4)  
circle(image1,(150,150),50,(25,25,255),4) #daire  
putText(image1,"TheDenizhan",(50,35),FONT\_HERSHEY\_COMPLEX,1,(25,25,255),1) #metin yazma  
#putText(değişken,"yazılacak yazı",(x,y)koordinatları,yazı şekli,yazı boyutu,renk,kalınlık  
imshow("image",image1)  
  
waitKey(0)  
destroyAllWindows()

**GÖRÜNTÜ PİRAMİTLERİ**

from cv2 import\*  
from numpy import\*  
  
image1=imread("wp34.jfif")  
ikikatB=pyrUp(image1) #resmi 2 kat büyütüyor(yükseklik\*2,genişlik\*2)  
ikikatK=pyrDown(image1) #resmi 2 kat küçültüyor(yükseklik/2,genişlik/2)  
  
print("orji",image1.shape) #resmin orjinal değerleri yazılıyor  
print("ikiKB",ikikatB.shape) #resmin 2 kat büyültülmüş değerleri yazılıyor  
print("ikiKK",ikikatK.shape) #resmin 2 kat küçültülmüş değerleri yazılıyor  
  
imshow("orj",image1)  
imshow("ikiKB",ikikatB)  
imshow("ikiKk",ikikatK)  
  
waitKey(0)  
destroyAllWindows()

**GÖRÜNTÜ PİRAMİTLERİ 2**

from cv2 import\*  
from numpy import\*  
  
image1=zeros((300,300,3),"uint8") #zeros fonksiyonu kullanarak siyah bir resim oluşturuyoruz  
 #zeros((yükseklik,genişlik,kanal),veritipi)  
print(image1) #yazdırıyoruz  
waitKey(0)  
destroyAllWindows()

**KAREYE ALMA**

from cv2 import\*  
from numpy import\*  
  
image4=imread("image-2.jpg")  
rectangle(image4,(100,200),(600,300),[0,0,255],9)  
 #rectangle fonksiyonu ile se.tiğimiz 2 nokta içerisine seçtiğimiz renk ve kalınlıkta bir dörtgen oluşturuyor  
imshow("kare",image4)

waitKey(0)  
destroyAllWindows()

**KAMERA DAN GÖRÜNTÜ ALMA**

from cv2 import\*  
from numpy import\*  
  
camera=VideoCapture(0) #parametre 0 ise kendi bilgisayarın kendi kamerasından görüntü alır  
 #parametre 1 dersek usb yada ek olarak taktığımız kaynaktan görüntü alır  
 #parametre 2 dersek dosya içersine yüklediğimiz bir videodan görüntü alır  
while True:  
 ret,goruntu=camera.read() #ret yazdığımız kameranın çalışıp çalışmadığını kontrol ediyor  
 #görüntü kameradan alınan görüntülerin birleşmiş hali olucak  
 #ve son olarak camera değişkeninden aldığımız fotoları görüntü değişkenine atıyoruz  
 imshow("cam",goruntu) #imshow ile goruntu gösteriliyor  
  
 if waitKey(10) & 0xFF == ord('q'): #waitKey yardımıyla içine girdiğimiz parametre her bir alınan görüntünün kaç milisaniye olacağı seçiliyor  
 break #daha sonra 0xFF ile çıkış yapmak için bir tuşa atıyoruz  
camera.release() #daha sonra kamerayı kullanmayı bırakmasını sağlıyoruz  
destroyAllWindows()

KAMERADAN ALINAN GÖRÜNTÜNÜN KAYIT EDİLMESİ

from cv2 import \*  
from numpy import \*  
#Kameradan alınan canlı görüntünün kayıt edilmesi  
camera=VideoCapture(0)  
width = int(camera.get(CAP\_PROP\_FRAME\_WIDTH))  
height = int(camera.get(CAP\_PROP\_FRAME\_HEIGHT))  
print(width,height)  
fourcc=VideoWriter\_fourcc(\*'MP4V') #Kameradan alınacak görüntünün hangi türde kayıt edileceğini belirler  
writer=VideoWriter("kayit.mp4",fourcc,20,(width,height))  
#sayı küçüldükçe video hızı azalır,sayı büyükçe video hızı artar  
while True:  
 ret,frame=camera.read()  
 writer.write(frame)  
 imshow("Kamera",frame)  
 if waitKey(10) & 0xFF == ord('q'):  
 break  
  
camera.release()  
writer.release()  
destroyAllWindows()

**KAMERADAN ALINAN GÖRÜNTÜYE İŞLEM**

from cv2 import\*  
from numpy import\*  
  
kamera=VideoCapture(0)  
  
while True:  
 ret,goruntu=kamera.read()  
  
 rectangle(goruntu,(200,100),(462,450),(0,0,255),3)  
 putText(goruntu, "TheDenizhan", (160,85), FONT\_HERSHEY\_SCRIPT\_COMPLEX, 2, (25, 25, 255), 2)  
  
 imshow("goruntu",goruntu)  
  
 if waitKey(1) & 0xFF == ord('q'):  
 break  
  
kamera.release()  
destroyAllWindows()

**THRESHOLDİNG**

from cv2 import\*  
from numpy import\*  
  
image=imread("wp2.jpg",0)  
#image=blur(image,(5,5))  
#simple Thresholding  
ret,thresh1=threshold(image,127,255,THRESH\_BINARY) #127 altındaysa 255 yap  
#ret,thresh2=threshold(image,127,255,THRESH\_BINARY\_INV) #  
#ret,thresh3=threshold(image,127,255,THRESH\_TRUNC)  
#ret,thresh4=threshold(image,127,255,THRESH\_TOZERO)  
#ret,thresh5=threshold(image,127,255,THRESH\_TOZERO\_INV)  
"""  
#adapted Thresholding  
thresh6=adaptiveThreshold(image,255,ADAPTIVE\_THRESH\_MEAN\_C,\  
 THRESH\_BINARY,11,2)  
thresh7=adaptiveThreshold(image,255,ADAPTIVE\_THRESH\_GAUSSIAN\_C,\  
 THRESH\_BINARY,11,2)  
"""

#OTSU THRESHOLDİNG  
ret2,thresh8=threshold(image,0,255,THRESH\_BINARY+THRESH\_OTSU)  
  
  
imshow("orji",image)  
imshow("thresh1",thresh1)  
#imshow("thresh2",thresh2)  
#imshow("thresh3",thresh3)  
#imshow("thresh4",thresh4)  
#imshow("thresh5",thresh5)  
#imshow("thresh6",thresh6)  
#imshow("thresh7",thresh7)  
imshow("Otsu",thresh8)  
  
waitKey(0)  
destroyAllWindows()

**MORFOLOJİK İŞLEMLER**

from cv2 import\*  
from numpy import\*  
  
image=imread("Morfolojjik.png")  
  
kernel=ones((3,3),uint8) #kernel...  
  
dilation=dilate(image,kernel,iterations=1) #resimdeki beyazları genişletme  
erosion=erode(image,kernel,iterations=1) #resimdeki beyazları daraltma  
dilation2=dilate(erosion,kernel,iterations=1) #resimdeki beyazları ilk önce daraltıp daha sonra genişletme  
opening=morphologyEx(image,MORPH\_OPEN,kernel) #resimdeki beyazları ilk önce daraltıp daha sonra genişletme  
closing=morphologyEx(image,MORPH\_CLOSE,kernel) #resimdeki beyazları ilk önce genişletip daha sonra daraltma  
gradyan=morphologyEx(image,MORPH\_GRADIENT,kernel) #genişlettiğimiz resimden daraltığımız resmi çıkartma  
tophat=morphologyEx(image,MORPH\_TOPHAT,kernel) #orjinal resimden openning işlemini çıkartma  
blackhat=morphologyEx(image,MORPH\_BLACKHAT,kernel) #orjinal işlemden closing işlemini çıkartma  
imshow("orji",image)  
imshow("dilations",dilation)  
imshow("erossion",erosion)  
imshow("dilations2",dilation2)  
imshow("openning",opening)  
imshow("closing",closing)  
imshow("gradyan",gradyan)  
imshow("tophat",tophat)  
imshow("blackhat",blackhat)  
  
  
waitKey(0)  
destroyAllWindows()

**CANNY,AUTOCANNY**

from cv2 import \*  
from numpy import \*  
"""  
def auto\_canny(blur, sigma=0.33):  
  
 v = median(blur)  
 lower = int(max(0, (1.0 - sigma) \* v))  
 upper = int(min(255, (1.0 + sigma) \* v))  
 edged = Canny(blur, lower, upper)  
  
  
 return edged  
"""  
image1 = imread("groot.jpg")  
imagegray = cvtColor(image1, COLOR\_BGR2GRAY)  
blur = GaussianBlur(imagegray, (3, 3), 0)  
  
  
canny=Canny(blur,100,1)  
  
  
  
  
wide = Canny(blur, 10, 220)  
tight = Canny(blur, 200, 230)  
#auto = auto\_canny(blur)  
  
imshow("groot", image1)  
imshow("blur", blur)  
# imshow("canny",canny)  
  
imshow("hepsi", hstack([wide, tight,canny,canny]))  
  
waitKey(0)  
destroyAllWindows()

**BİTWİSE OPERATÖRLERİ**

from cv2 import \*  
from numpy import \*  
  
image1=imread("Bitwise1.png")  
image2=imread("bitwise2.png")  
  
bit\_and=bitwise\_and(image1,image2) #İki görüntüye and işlemi yapar:ikisi 1 ise 1 döner yoksa her koşul 0  
bit\_or=bitwise\_or(image1,image2) #iki görüntüye or işlemi yapar:işlemde 1 varsa 1 dir yoksa 0  
bit\_not=bitwise\_not(image1) #görüntüye not işlemi yapar:1 i 0 ,0 ı 1 yapar  
bit\_xor=bitwise\_xor(image1,image2) #iki görüntüye xor işlemi yapar:0,0ve 1,1 0dır, 0,1 ve 1,0 1 dir  
  
imshow("image1",image1)  
imshow("image2",image2)  
imshow("bitAnd",bit\_and)  
imshow("bitOr",bit\_or)  
imshow("bitNot",bit\_not)  
imshow("bitXor",bit\_xor)  
  
waitKey(0)  
destroyAllWindows()