

Aşağıdaki işlem basamaklarını gerçekleştiren kod satırlarını yazarak uygulayınız.

- 1- Tümleşik kameradan görüntü alınız;
 - 2- Görüntünün belirli bir alanını ayırarak etrafına bir kare çizin
 - 3- İmaj parçasını uygun renk uzayına çevirerek belirli nesneler üzerinde alt ve üst eşik değerleri belirleyin.
 - 4- Bu eşik parametreleri ile imaj parçasını binarize ediniz.
 - 5- İkili imaj parçası üzerinden öznitelik hesaplayınız. (Ortalama vs)
 - 6- Hesapladığınız öznitelik ile koşullandırma yaparak imaj üzerine rengi yazdırınız.
 - 7- Nesne merkezini hesaplayarak nesneyi geometrik bir şekil içine alınız.
 - 8- Yazdığınız programı çalıştırarak ekran çıktısını bir sonraki satıra ekleyiniz.
- ** Bu uygulama için en az derste çalıştığımız uygulamaya bir ekleme yapmanızı istiyorum.**
- Örneğin birden fazla rengi seçebilirsiniz.Ya da ekrana bir scroll bar çizerek (sadece piksellerin rengini değiştirerek) kırmızı renkli kalemin scrollun üzerindeki hareketi ile alınan resmin parlaklığını parlaklığını ya da renk uzayındaki konumunu değiştirebilirsiniz.
- Yaptığınız görev ile ilgili açıklama yapmalısınız.

```
import cv2
from cv2 import VideoCapture
from cv2 import bitwise_and
from cv2 import bitwise_or
import numpy as np

cap=VideoCapture(0)
alan=1
satirW=1
sutunW=1
def merkez_hesapla(alan,satirW,sutunW,mask):
    for i in range(int(genislik)):
        for j in range(int(yukseklık)):
            if(mask[j,i]==255):
                alan+=1
                satirW+=i
                sutunW+=j

    return alan,satirW,sutunW
def kirmizi(alan,satirW,sutunW):
    lowerb=np.array([0,100,0])
    upperb=np.array([13,255,255])
    mask=cv2.inRange(hsv,lowerb,upperb)
    alan,satirW,sutunW=merkez_hesapla(alan,satirW,sutunW,mask)
    merkezX=satirW/alan
    merkezY=sutunW/alan
    if (alan>3500):
```

```

        cv2.circle(frame, (int(merkezX), int(merkezY)), 2, (50, 255, 50), 2)

cv2.putText(frame, "Kirmizi", (int(merkezX), int(merkezY)), 0, 1, (255, 0, 0),
2)

    return mask
def mavi(alan, satirW, sutunW):
    lowerb=np.array([80, 53, 53])
    upperb=np.array([110, 255, 255])
    mask=cv2.inRange(hsv, lowerb, upperb)
    alan, satirW, sutunW=merkez_hesapla(alan, satirW, sutunW, mask)
    merkezX=satirW/alan
    merkezY=sutunW/alan
    if(alan>3500):
        cv2.circle(frame, (int(merkezX), int(merkezY)), 2, (0, 0, 255), 2)

cv2.putText(frame, "Mavi", (int(merkezX), int(merkezY)), 0, 1, (255, 0, 0), 2)
    return mask
def yesil(alan, satirW, sutunW):
    lowerb=np.array([38, 34, 0])
    upperb=np.array([74, 255, 255])
    mask=cv2.inRange(hsv, lowerb, upperb)
    alan, satirW, sutunW=merkez_hesapla(alan, satirW, sutunW, mask)
    merkezX=satirW/alan
    merkezY=sutunW/alan
    if(alan>3500):
        cv2.circle(frame, (int(merkezX), int(merkezY)), 2, (0, 0, 255), 2)

cv2.putText(frame, "Yesil", (int(merkezX), int(merkezY)), 0, 1, (255, 0, 0), 2)
    return mask
def kesilmisBolge(maske):
    kesilmis=maske[50:200, 50:200]
    cv2.rectangle(frame, (50, 50), (200, 200), (255, 255, 255), 2)
    alan=merkez_hesapla(1, 1, 1, kesilmis)
    if(alan[0]>=5000):
        cv2.putText(frame, "Renk var", (210, 125), 0, 1, (0, 0, 255), 2)
    else:
        cv2.putText(frame, "Renk yok", (210, 125), 0, 1, (0, 0, 255), 2)

    return kesilmis

while(1):
    genislik=cap.get(cv2.CAP_PROP_FRAME_WIDTH)

```

```

yukseklık=cap.get(cv2.CAP_PROP_FRAME_HEIGHT)
ret,frame=cap.read()
frame=cv2.flip(frame,1)
hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)

mask1=kirmizi(Alan,satırW,sutunW)
mask2=yesil(Alan,satırW,sutunW)
mask3=mavi(Alan,satırW,sutunW)
toplamlaske=mask1+mask2+mask3
veMaske=bitwise_and(frame,frame,mask=toplamlaske)
genislik=150
yukseklık=150
kesilmis=kesilmisBolge(toplamlaske)
cv2.imshow("kesilmis",kesilmis)
cv2.imshow("RGBve",veMaske)
cv2.imshow("Kamera",frame)
if(cv2.waitKey(1) & 0xFF==ord("z")):
    break

```

istediğim renk aralığı için ilk önce farklı bir projede trackbar ile hsv uzayındaki minimum ve maksimum değer aralığını belirledim daha sonra inRange komutu ile bu aralıklarla hsvye çevirdiğim görüntüyü binarize edip maske oluşturdum.

ilk önce kameranın piksel sayısını öğrendim daha sonra merkez_hesapla fonksiyonu ile $\sum_{i=1}^i \sum_{j=1}^j$

$i^a x j^b x f(i, j)$ (a=1 b=0 satır,a=0 b=1 sütun) formülünü uygulayıp binarize ettiğim çıktının içerisindeki toplam beyaz piksel sayısını, satırW ve sütunW ile satır-sütundaki renk ağırlığını belirledim.

bu değerleri toplam beyaz piksel sayısına böldüğümde X ve Y eksenindeki merkezi buldum ve cv2 da daha önceden fonksiyon haline getirilmiş çember fonksiyonu ile işaretledim. bunu her renk işlemi için tekrarladım. ayrılan bölgeyi ise oluşturduğum maskelerin toplamındaki beyaz piksel sayısına göre renk varlığını belirledim.

cv2 kütüphanesindeki hazır fonksiyonlar ile oluşturduğum merkez belirleme ve objeyi dikdörtgen içine alma için: [Goruntu_isleme/Odev2 at main · sidoniusblon/Goruntu_isleme \(github.com\)](https://github.com/sidoniusblon/Goruntu_isleme)

