



## GÖRÜNTÜ İŞLEME PROJE ÖDEVİ RAPORU

Okul:

Bolu İzzet Baysal Üniversitesi

Proje Üyeleri:

Bekir ORUK

Cevahir ATICI

Hediye Zeynep ÖZDER

Projenin Adı:

Kırmızı, Yeşil, Mavi Renk Tanıyıcı

Projenin Amacı:

Kırmızı, yeşil ve mavi renklerini anlayan bir uygulama yapmak.

Projede Kullanılan Kodun Açıklaması:

Gerekli kütüphaneler çağrılır.

```
import numpy as np
import cv2

cap = cv2.VideoCapture(0)

while(1):
```

Aşağıdaki kod satırı ile kameradan alınan görüntü okunur.

```
_, imageFrame = cap.read()
```

Aşağıdaki kod satırı, bir görüntünün renk uzayını BGR (RGB renk uzayı) ile HSV (ton-doygunluk-değer) renk uzayına dönüştürüyor.

```
hsvFrame = cv2.cvtColor(imageFrame, cv2.COLOR_BGR2HSV)
```

Aşağıdaki kod satırları belirli bir renk aralığına sahip pikselleri ayırmak ve bu pikselleri işlemek için kullanılan bir renk tabanlı maske oluşturur.

```
red_lower = np.array([136, 87, 111], np.uint8)
red_upper = np.array([180, 255, 255], np.uint8)
red_mask = cv2.inRange(hsvFrame, red_lower, red_upper)

green_lower = np.array([25, 52, 72], np.uint8)
green_upper = np.array([102, 255, 255], np.uint8)
green_mask = cv2.inRange(hsvFrame, green_lower, green_upper)

blue_lower = np.array([94, 80, 2], np.uint8)
blue_upper = np.array([120, 255, 255], np.uint8)
blue_mask = cv2.inRange(hsvFrame, blue_lower, blue_upper)
```

Aşağıdaki kod satırları belirli bir renge ait pikselleri tespit etmek için renk tabanlı bir maske oluşturur ve bu maskeyi kullanarak görüntü çerçevesinde yalnızca o renge ait pikselleri belirler.

```
kernal = np.ones((5, 5), "uint8")
```

Kırmızı renk için:

```
red_mask = cv2.dilate(red_mask, kernal)
res_red = cv2.bitwise_and(imageFrame, imageFrame, mask = red_mask)
```

Yeşil renk için:

```
green_mask = cv2.dilate(green_mask, kernal)
res_green = cv2.bitwise_and(imageFrame, imageFrame, mask = green_mask)
```

Mavi renk için:

```
blue_mask = cv2.dilate(blue_mask, kernal)
res_blue = cv2.bitwise_and(imageFrame, imageFrame, mask = blue_mask)
```

Aşağıdaki kod satırları kırmızı, yeşil ve mavi alanlarını takip etmek için kontur oluşturur ve tespit edilen alanları dikdörtgen ve metinle işaretler.

Kırmızı renk için:

```
contours, hierarchy = cv2.findContours(red_mask, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)

for pic, contour in enumerate(contours):
    area = cv2.contourArea(contour)
    if(area > 300):
        x, y, w, h = cv2.boundingRect(contour)
```

```
        imageFrame = cv2.rectangle(imageFrame, (x, y), (x + w, y + h), (0,
0, 255), 2)

        cv2.putText(imageFrame, "Red Colour", (x, y),
cv2.FONT_HERSHEY_SIMPLEX, 1.0, (0, 0, 255))
```

Yeşil renk için:

```
    contours, hierarchy = cv2.findContours(green_mask, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)

    for pic, contour in enumerate(contours):
        area = cv2.contourArea(contour)
        if(area > 300):
            x, y, w, h = cv2.boundingRect(contour)
            imageFrame = cv2.rectangle(imageFrame, (x, y), (x + w, y + h), (0,
255, 0), 2)

            cv2.putText(imageFrame, "Green Colour", (x, y),
cv2.FONT_HERSHEY_SIMPLEX, 1.0, (0, 255, 0))
```

Mavi renk için:

```
    contours, hierarchy =
cv2.findContours(blue_mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)

    for pic, contour in enumerate(contours):
        area = cv2.contourArea(contour)

        if(area > 300):
            x, y, w, h = cv2.boundingRect(contour)
            imageFrame = cv2.rectangle(imageFrame, (x, y),(x + w, y + h),
(255, 0, 0), 2)

            cv2.putText(imageFrame, "Blue Colour", (x, y),
cv2.FONT_HERSHEY_SIMPLEX,1.0, (255, 0, 0))
```

Programı sonlandırma:

```
cv2.imshow("Color Detection", imageFrame)
if cv2.waitKey(10) & 0xFF == ord('q'):
    cap.release()
    cv2.destroyAllWindows()
    break
```