ROBOTIK KOL

Not:

> bazı maddeler prosedür gereği yazılmaktadır. sizin için gereksiz olan kısımları kendi raporunuzda kullanırken çıkarabilirsiniz. gereğinden fazla uzun olduğunu düşündüğünüz raporları iki parçaya bölerek kullanmanız tavsiye edilir.

ELEKTRONIK

- raspberry pi'nin kurulması:
    - standart rpi kurulumu yapıldı:
        - rasbian işletim sistemi kuruldu
        - sistem güncellendi
        - ssh, vnc, gpio pinleri, raspberry camera ayarları raspi-config kısmından aktif edildi.
        - opencv kütüphanesi pip kullanılarak yüklendi.
        - yükleme kontrol edildi.

      sonuc: raspberry pi 3B+ kullanıma hazır

YAZILIM

- Renk Tanıma Araştırmaları:
    - OpenCV kütüphanesi kullanılarak renk tanıma işlemlerinin temel mantığı araştırıldı.
        - bulunan kısa eğitimler:
            - OpenCV and Python Color Detection
            - Black and white image colorization with OpenCV and Deep Learning
            - Automatic color correction with OpenCV and Python
            - Determining object color with OpenCV
    - Eğitimlerdeki anlatılanlara göre basit renk algılama projesi yapıldı başarılı şekilde çıktı alındı:
    - **Projede nasıl bir yol izlendi ?**
        - Renk üzerine kurulu bir sistem olduğundan dolayı projede HSV (Hue, Saturation, Value) adı verilen renk ayrıştırma kullanıldı.
        - Bilgisayarın tam olarak renklerin ayrımını yapabilmesi (kırmızı ve turuncu tonlarına karıştırılmaması) için kısıtlamalar tanımlamamız gerekiyor. Aşağıdaki kod parçasında ilk satırdaki tanımlamaya bakacak olursak eğer;

```
# BGR olarak alınır opencv resimleri numpy dizileri türünde
sıralarken ters olarak çıktı verir
([17, 15, 100], [50, 56, 200])
```

bu tanımlama satırında;

- R için; 100 <= R <= 200,
- G için; 15 <= G <= 56,
- B için; 17 <= B <= 50

değerlerinin sınır olarak belirlendiğini görebiliriz.

sonra bu değerleri tek tek opencv'nin içinde gelen *inRange* fonksiyonu ile opencv'ye tanıttık ve hedef resim dosyası üzerinde işlem yaptırdık

- kodlar:

```
# gerekli paketlerin çağrılması
import numpy as np
import cv2

# opencv ile resmi yükle
image = cv2.imread("img.png")


# kısıtlamalar
boundaries = [
    ([17, 15, 100], [50, 56, 200]),
    ([86, 31, 4], [220, 88, 50]),
    ([25, 146, 190], [62, 174, 250]),
    ([103, 86, 65], [145, 133, 128])
  ]

# loop over the boundaries
for (lower, upper) in boundaries:
  # create NumPy arrays from the boundaries
  lower = np.array(lower, dtype = "uint8")
  upper = np.array(upper, dtype = "uint8")
  # find the colors within the specified boundaries and
apply
  # the mask
  mask = cv2.inRange(image, lower, upper)
  output = cv2.bitwise_and(image, image, mask = mask)
  # show the images
  cv2.imshow("images", np.hstack([image, output]))
  cv2.waitKey(0)
```

- çıktısı: