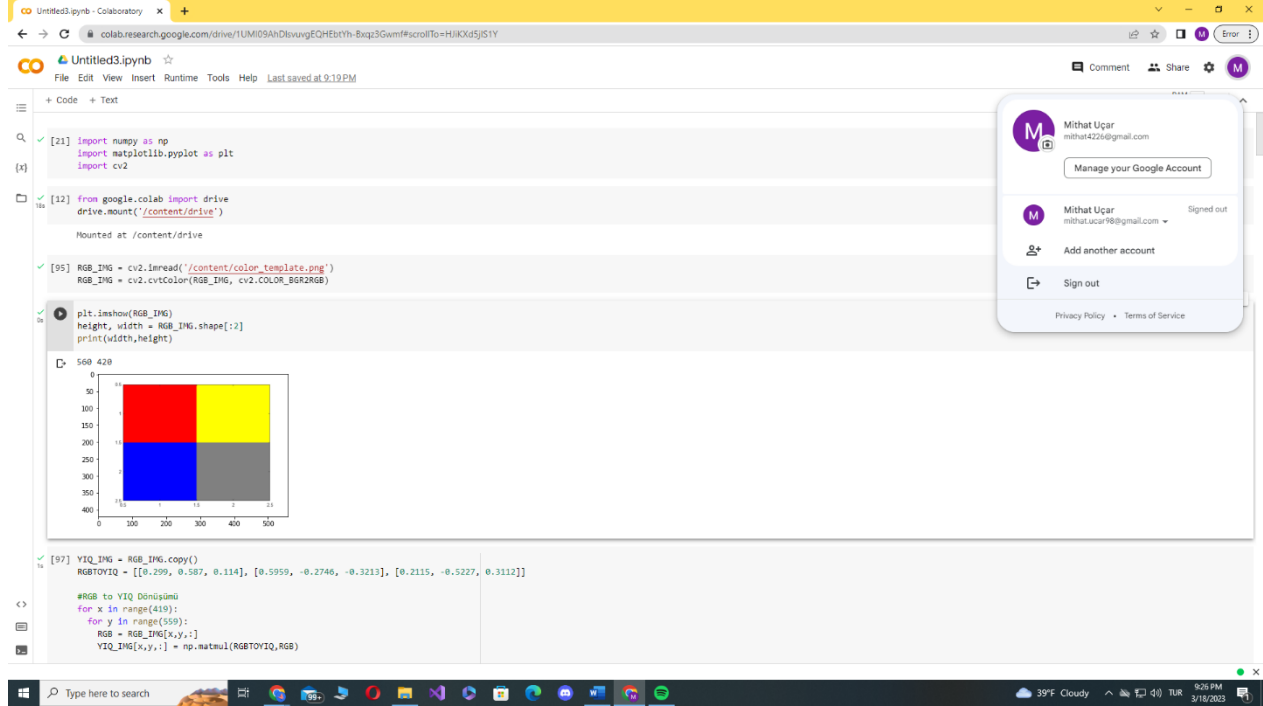


Pattern Recognition HW1



```
[21] import numpy as np
import matplotlib.pyplot as plt
import cv2

[12] from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

[95] RGB_IMG = cv2.imread('/content/color_template.png')
RGB_IMG = cv2.cvtColor(RGB_IMG, cv2.COLOR_BGR2RGB)

plt.imshow(RGB_IMG)
height, width = RGB_IMG.shape[:2]
print(width,height)

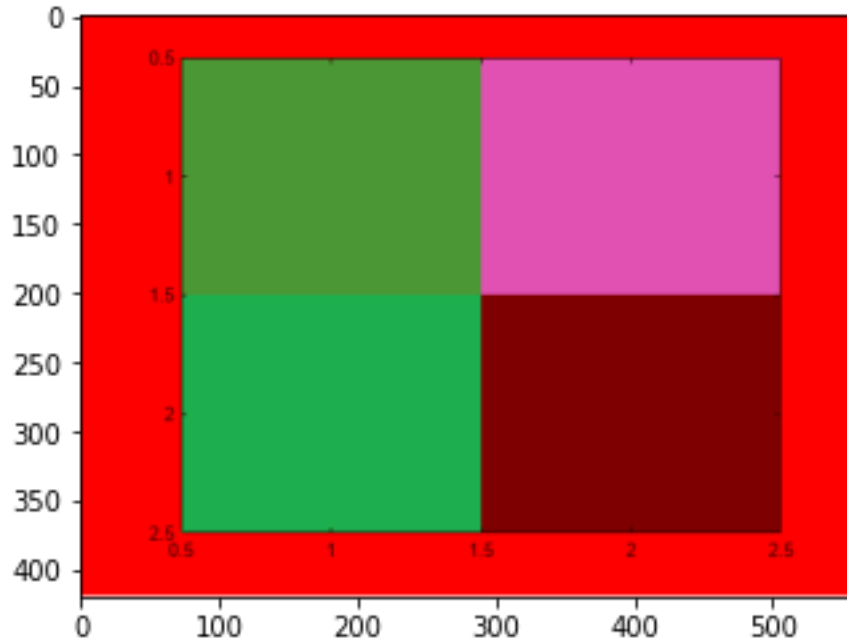
568 428

[97] YIQ_IMG = RGB_IMG.copy()
RGBTOYIQ = [[0.299, 0.587, 0.114], [0.5959, -0.2746, -0.3213], [0.2115, -0.5227, 0.3112]]

#RGB to YIQ Dönüşümü
for x in range(419):
    for y in range(559):
        RGB = RGB_IMG[x,y,:]
        YIQ_IMG[x,y,:] = np.matmul(RGBTOYIQ,RGB)
```

RGB YIQ DÖNÜŞÜMLERİ

-Resmin kaç pikseli olduğunu shape methoduyla buldum for döngüleriyle bütün pikselleri verilen matris kullanarak yiq dönüşümü yapıldı.



-Aynı şekilde verilen matris kullanarak rgb dönüşümü yapıldı.

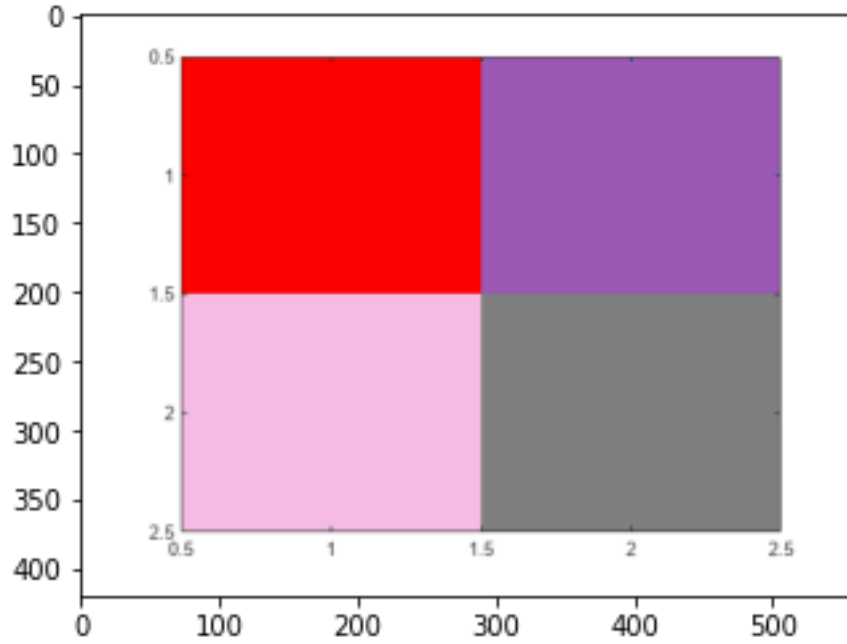




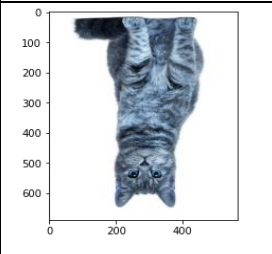
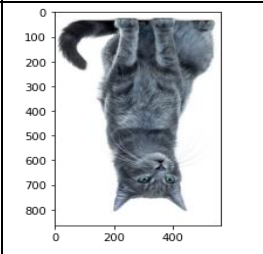
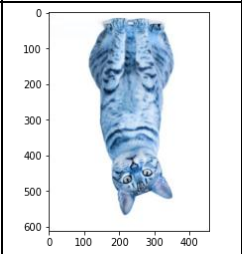
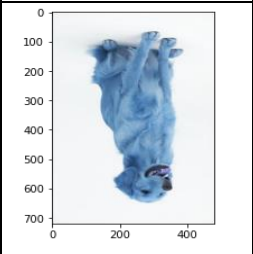
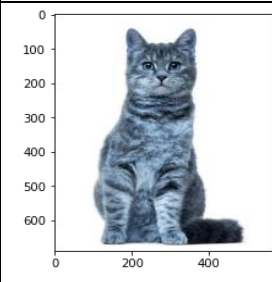
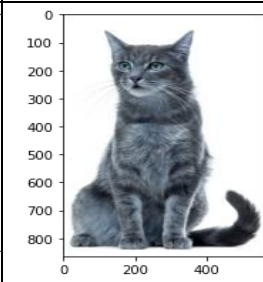
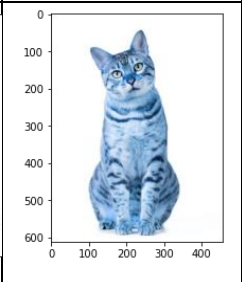
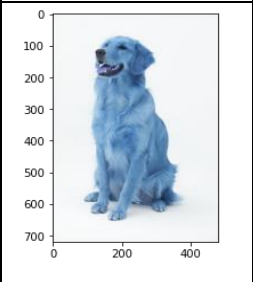
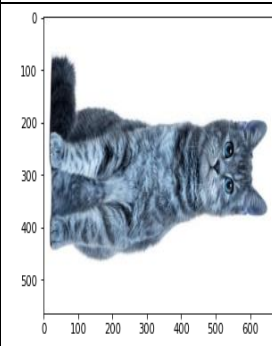
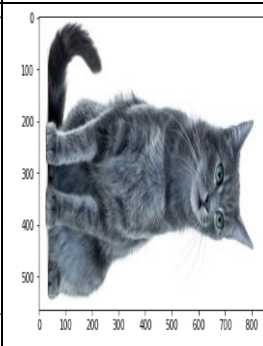
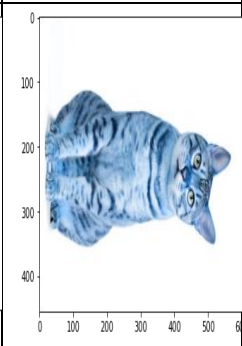
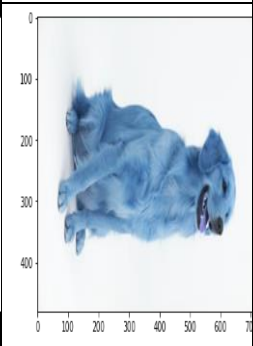
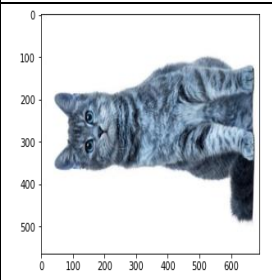
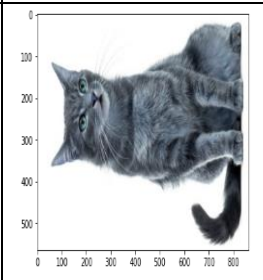
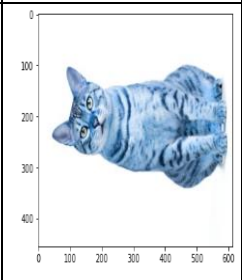
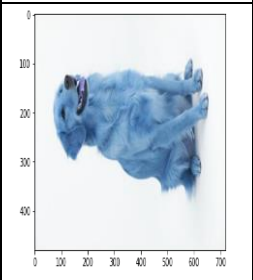


IMAGE MANIPULATIONS

-flip fonksiyonu kullanarak yatayda ve düzeyde dönüşümler yapıldı. Parametre olarak 0 dikey 1 ise yatayda çevirme işlemini yapmakta.

-rotate fonksiyonu kullanılarak saat ve saatin tersi yönlerde 90 derece dönüşümler yapıldı. Parametre olarak cv2’de tanımlanmış makrolar kullanıldı (saat yönü için ROTATE_90_CLOCKWISE tersi için ROTATE_90_COUNTERCLOCKWISE).

-resize işlemi için öncelikle resize boyutuna göre istenen width ve height piksel sayıları hesaplandı resize fonksiyonuyla aspect ratio bozmadan dönüşüm yapıldı.

				
Flip vertically				
Flip horizontal ly				
Rotate90 clockwise				
Rotate90 counter clockwise				
Resize	