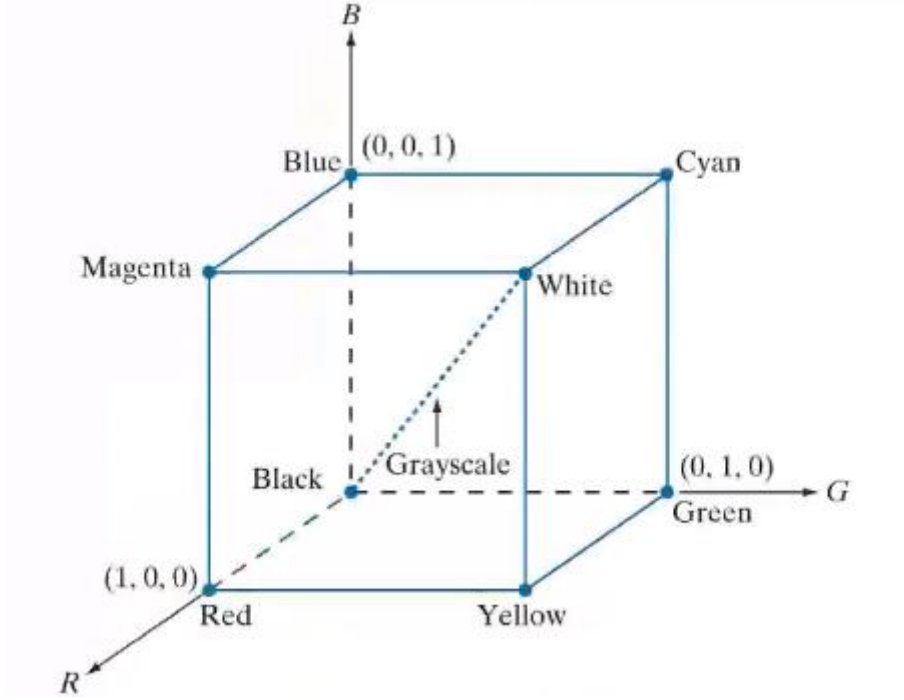
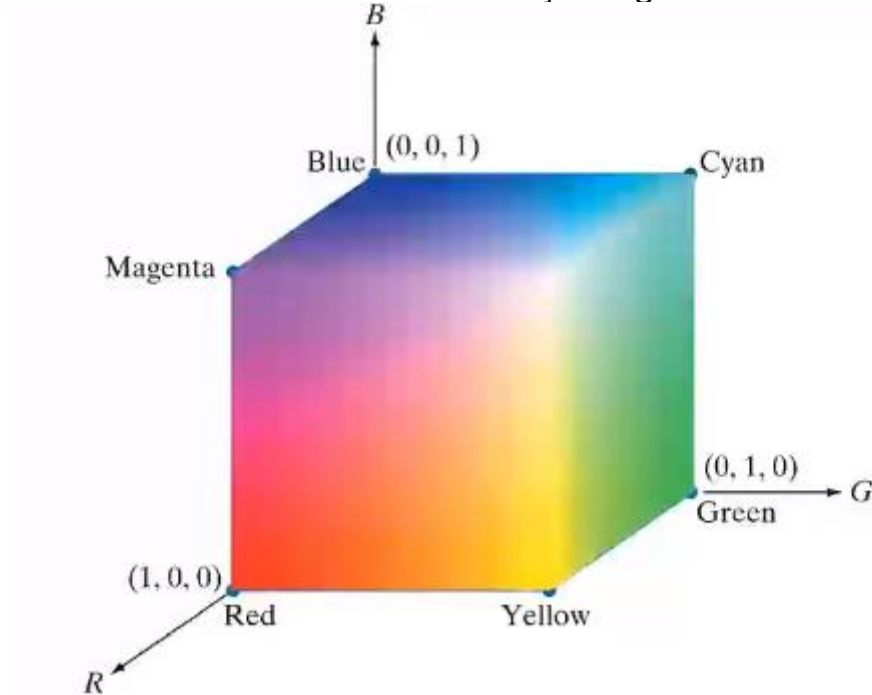


RENK UZAYI

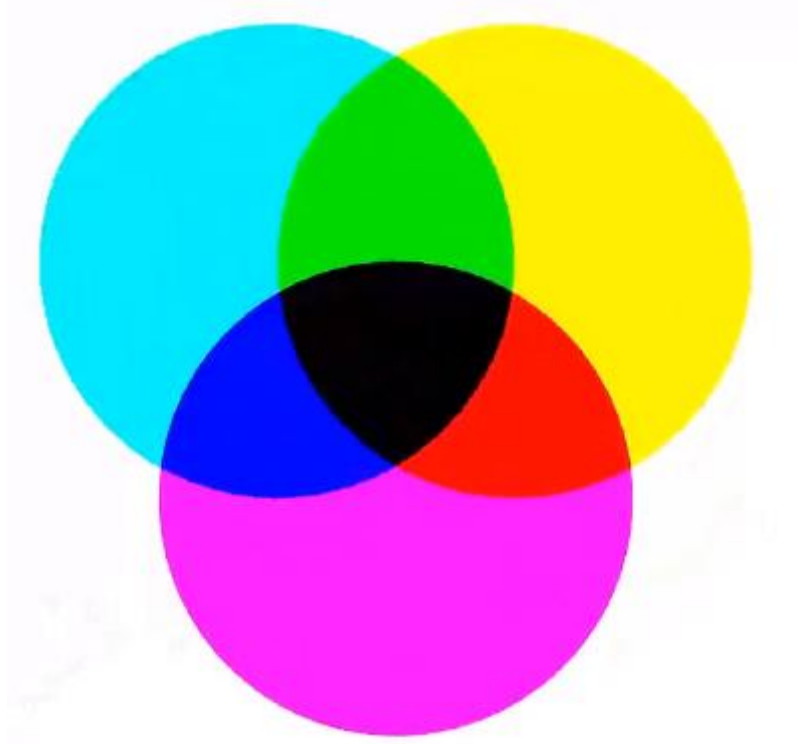
Görüntü işleme açısından en yaygın kullanılan renk modelleri, monitörler için RGB yani ana renkler, renkli yazıcılar için CMY veya CMYK bunlar da ara renkler, son olarak insanların rengi algılama şekline en yakın olan HSV modeli. RGB, CMY ve HSV modelleri daha kullanışlı olduğu için bunlar ele alındı. RGB modeli Kartezyen koordinat sistemine dayanmaktadır.



Ana renklerin ve ara renklerin noktalarını birleştirildiğinde RGB renk küpü elde ediliyor.



CMY modeli: Camgöbeği, magenta ve sarı ara renklerinden oluşur. Bu renklerin birleştiğinde ortaya net bir siyah rengi oluşmuyor.

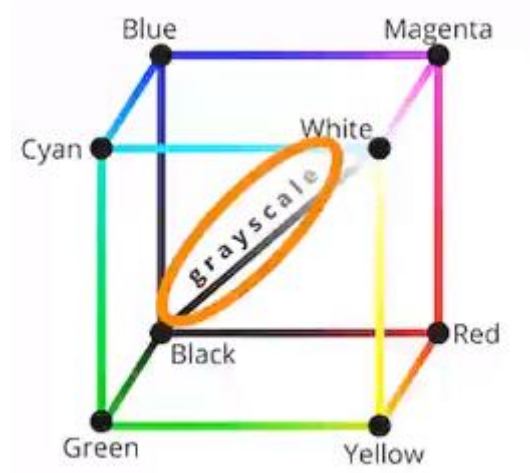


Renkli görüntüyü bir yazıcıdan çıkardığımızda RGB modelinden CMY modeline dönüşümü gerçekleştiriyor. Fakat resimde siyah bir renk varsa çok iyi bir şekilde basılmayacak. Yazıcılarda siyah rengini daha düzgün üretmek için CMYK modeli kullanılır. Burada dördüncü renk olan K, siyahtır.

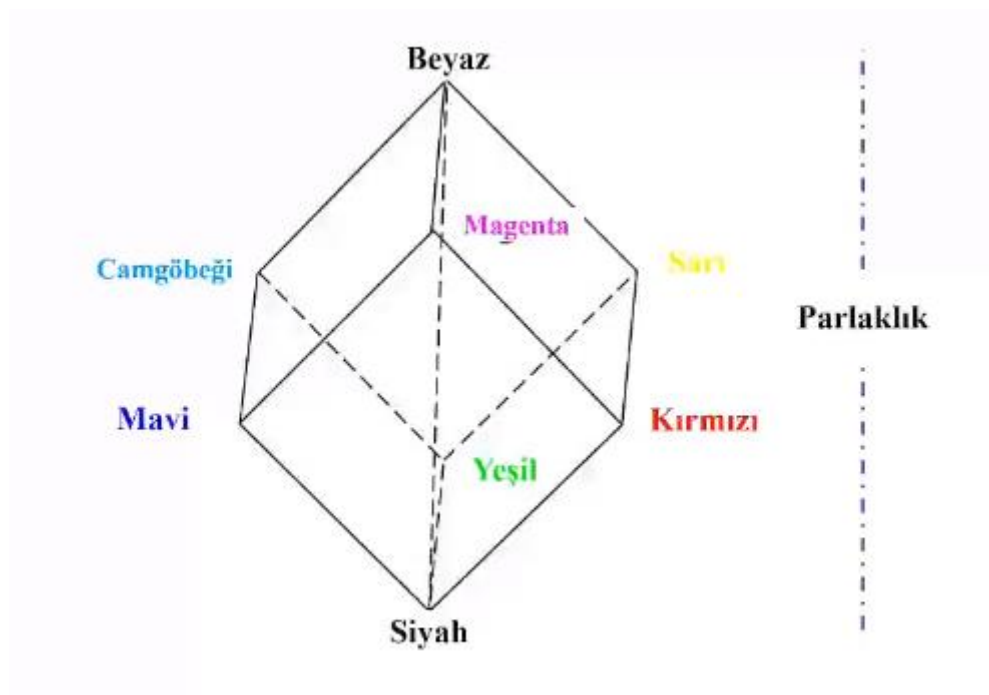
HSV modeli ton, doygunluk ve parlaklık değeri bileşenlerinden oluşur.

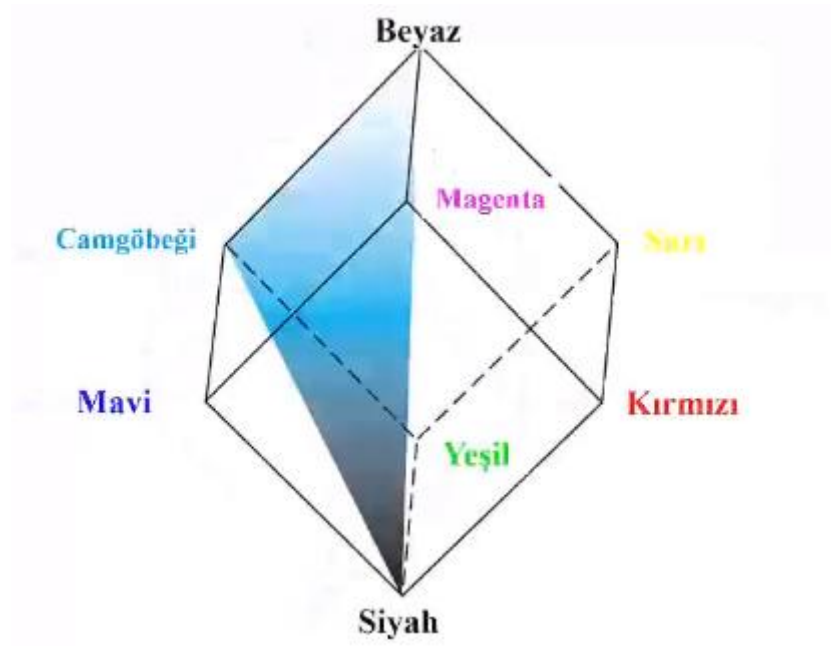


RGB renk küpünde de hem siyah hem de beyaz renkleri var.



Bu iki renk arasındaki mesafeye parlaklık referansı olarak alınabilir. Böylece RGB renk modelinde kolaylıkla HSV modeline dönüşüm sağlanır.





RENK UZAYI UYGULAMASI

OpenCV kütüphanesini kullanarak modeller arasında dönüşüm yapmak da çok kolaydır.

Kütüphaneleri ekleyip, resmi okuma:

```
[1] import matplotlib.pyplot as plt
    import cv2

[2] img = cv2.imread('/content/cube.png')

[3] rgb_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

plt.figure(figsize=(15,20))

# OpenCv BGR
plt.subplot(1,2,1)
plt.imshow(img)
plt.xticks([], plt.yticks([]))

# Matplotlib RGB (orijinal hali)
plt.subplot(1,2,2)
plt.imshow(rgb_img)
plt.xticks([], plt.yticks([]))

plt.show()
```



Sadece tek bir bileşenden görüntüye bakılabilir:

red → [x, y, RGB]

R = 0 G = 1 B = 2

```
▶ red = rgb_img.copy()
  red[:, :, 1] = 0
  red[:, :, 2] = 0

  green = rgb_img.copy()
  green[:, :, 0] = 0
  green[:, :, 2] = 0

  blue = rgb_img.copy()
  blue[:, :, 0] = 0
  blue[:, :, 1] = 0
```

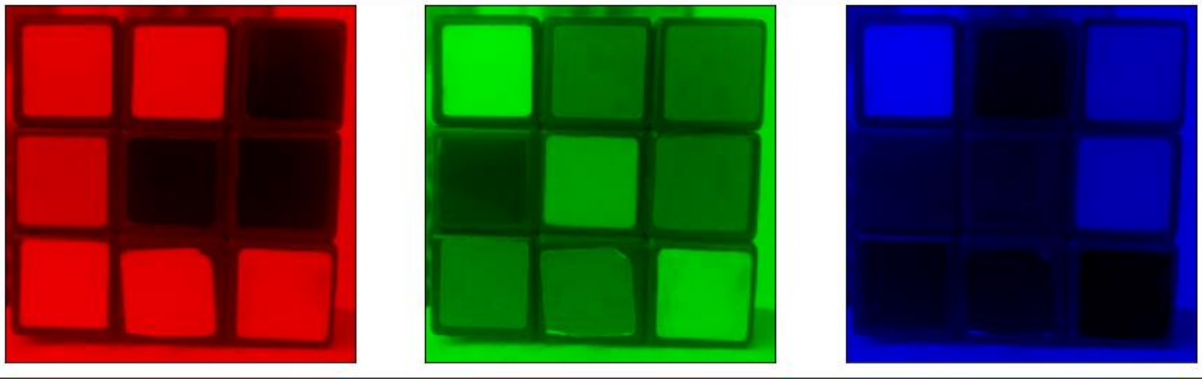
```
▶ plt.figure(figsize=(15,20))

# red
plt.subplot(1,3,1)
plt.imshow(red)
plt.xticks([], plt.yticks([]))

# green
plt.subplot(1,3,2)
plt.imshow(green)
plt.xticks([], plt.yticks([]))

# blue
plt.subplot(1,3,3)
plt.imshow(blue)
plt.xticks([], plt.yticks([]))

plt.show()
```



HSV renk modeli:

Fotoğrafı okuma:

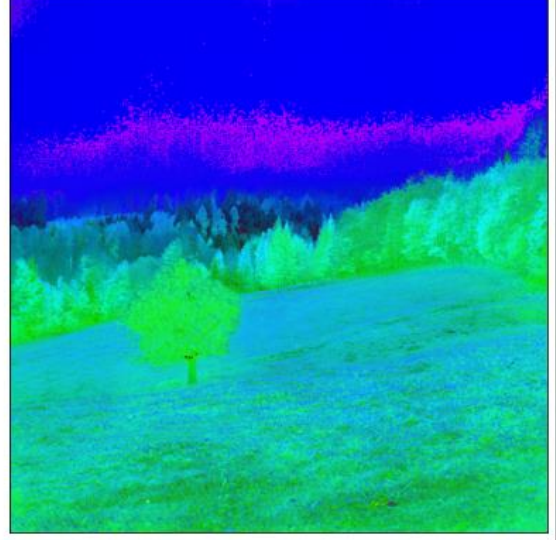
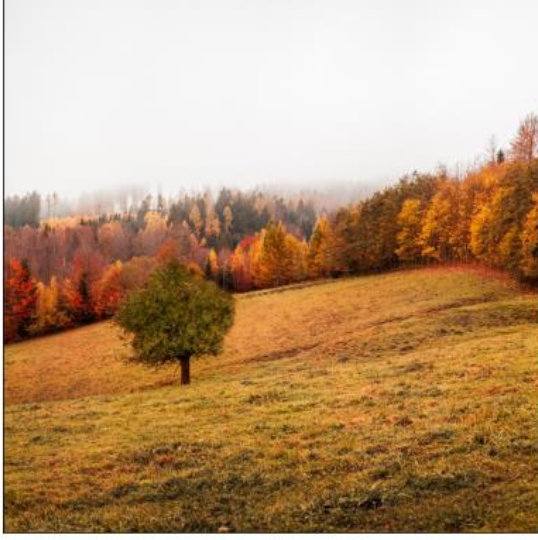
```
manzara_img = cv2.imread('/content/manzara.jpg')
```

Orijinal fotoğrafı görebilmemiz için BGR modelinden RGB modeline ve HSV modeline dönüştürme:

```
rgb_manzara_img = cv2.cvtColor(manzara_img, cv2.COLOR_BGR2RGB)  
hsv_manzara_img = cv2.cvtColor(manzara_img, cv2.COLOR_BGR2HSV)
```

Ekrana basma:

```
plt.figure(figsize=(15,20))  
  
# orijinal resim  
plt.subplot(1,2,1)  
plt.imshow(rgb_manzara_img)  
plt.xticks([], plt.yticks([]))  
  
# hsv  
plt.subplot(1,2,2)  
plt.imshow(hsv_manzara_img)  
plt.xticks([], plt.yticks([]))  
  
plt.show()
```



KAYNAKÇA

Bilgeiř “Herkes iin Yapay Zekâ II” eęitimi.

KODLUYORUZ
geleceęi kodluyoruz >_

 **EMpower**
Enriching young lives in emerging markets