

Menampilkan gambar dengan matplotlib

Contoh 1

```
import cv2
import matplotlib.pyplot as plt

def main():
    # Memuat gambar
    img = cv2.imread('img/parrot.jpg')

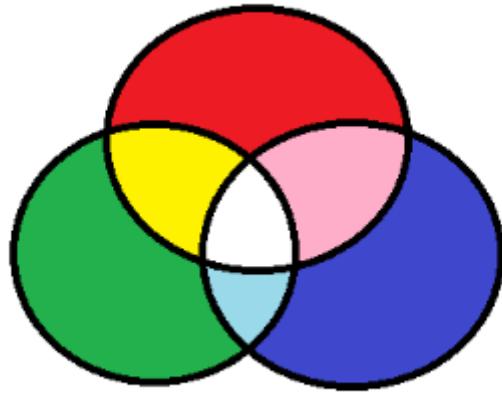
    # cv2 use BGR , convert the color from BGR to RGB for displaying using matplotlib
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    plt.imshow(img)
    plt.colorbar()
    plt.title('Parrot Image')
    plt.show()

if __name__ == "__main__":
    main()
```

1. RGB (Merah, Hijau, Biru)

Model warna RGB merepresentasikan warna menggunakan tiga warna primer: merah, hijau, dan biru. Setiap piksel dalam gambar merupakan kombinasi ketiga warna ini dengan intensitas berbeda, menciptakan beragam kemungkinan warna. Ini adalah ruang warna standar yang digunakan untuk menampilkan gambar di layar dan mengolah gambar digital.



2. BGR (Biru, Hijau, Merah)

OpenCV menggunakan BGR, bukan RGB, sebagai ruang warna default-nya, tempat kanal merah dan biru dipertukarkan. Meskipun perbedaan ini mungkin tampak kecil, penting untuk diingat saat menggunakan OpenCV agar tidak terjadi ketidakcocokan warna.

3. HSV (Hue, Saturation, Value)

HSV adalah ruang warna lain yang digunakan dalam OpenCV. HSV didasarkan pada persepsi manusia terhadap warna dan membagi informasi warna menjadi tiga komponen:

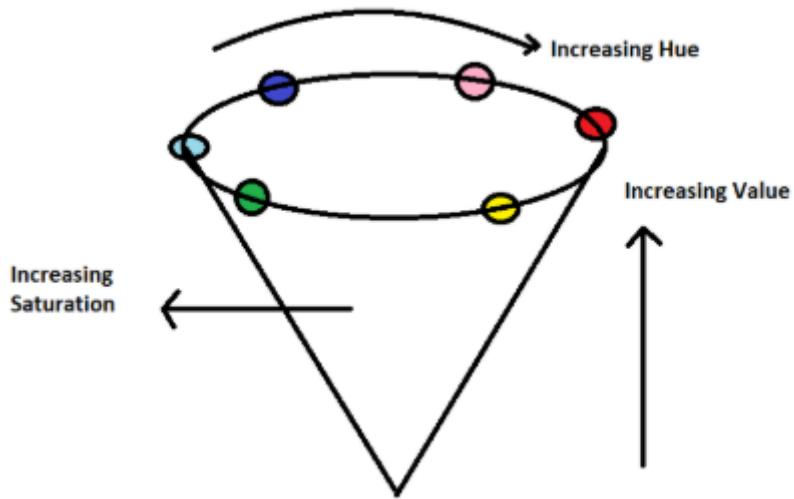
Hue: Mewakili warna itu sendiri (misalnya merah, biru, hijau).

Saturation: Menunjukkan kecerahan warna.

Value: Menentukan kecerahan warna.

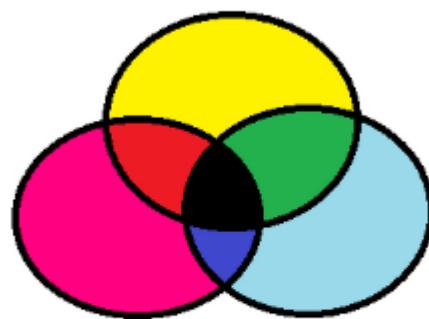
Nilai hue berkisar antara 0 hingga 179, sedangkan saturation dan value berkisar antara 0 hingga 255.

HSV berguna untuk tugas-tugas seperti segmentasi warna dan deteksi objek. Misalnya, HSV memungkinkan kita untuk dengan mudah mengisolasi warna tertentu (seperti merah atau biru) dalam sebuah gambar.



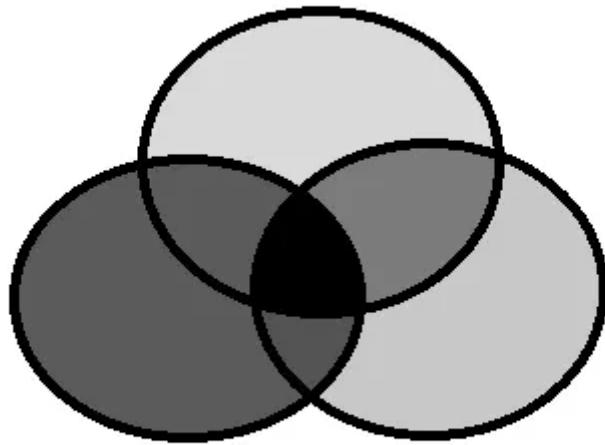
4. CMYK (Cyan, Magenta, Yellow, Black)

Model warna CMYK digunakan untuk pencetakan warna. Model ini bersifat subtraktif, artinya warna dihasilkan dengan mengurangi cahaya menggunakan berbagai kombinasi tinta cyan, magenta, yellow, dan hitam. Dalam model ini, warna menjadi lebih gelap seiring bertambahnya tinta. Model ini digunakan dalam desain cetak dan grafis, tetapi tidak sering digunakan pada OpenCV untuk pemrosesan gambar.



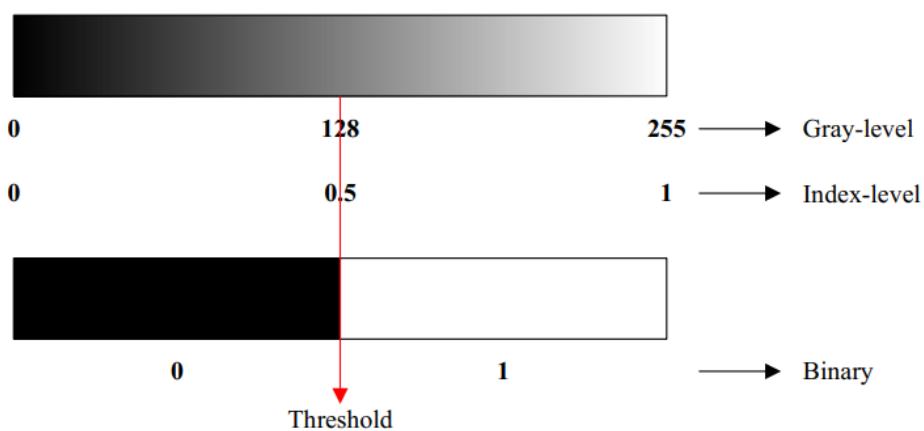
5. Grayscale (Skala Abu-abu)

Gambar skala abu-abu hanya berisi nuansa abu-abu, dengan setiap piksel mewakili tingkat intensitas yang berbeda, dari hitam hingga putih. Gambar ini digunakan ketika informasi warna tidak diperlukan, seperti dalam deteksi tepi atau tugas ambang batas gambar.



6. Black & White (Binary)

Model warna black and white (hitam putih) adalah representasi gambar yang hanya menggunakan informasi intensitas cahaya tanpa informasi warna.



Konversi Antar Ruang Warna di OpenCV

OpenCV memudahkan konversi antar ruang warna yang berbeda. Hal ini berguna ketika kita ingin melakukan tugas seperti deteksi objek berdasarkan warna atau mengekstrak fitur menggunakan informasi warna tertentu.

Mari kita lihat cara melakukan konversi ruang warna umum menggunakan fungsi cv2.cvtColor() OpenCV. Di sini, kita menggunakan gambar sampel acak yang dapat Anda

Contoh Konversi

```
import cv2
import matplotlib.pyplot as plt

def main():
    # Memuat gambar
    image = cv2.imread('img/parrot.jpg')

    # convert the color from BGR to RGB for displaying using matplotlib
    image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    image_hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    # Pixels above 127 become 255 (white), others become 0 (black)
    i_, image_binary = cv2.threshold(image_gray, 127, 255, cv2.THRESH_BINARY)

    plt.figure(figsize=(15, 10))

    plt.subplot(231), plt.imshow(image), plt.title('Original Image')
    plt.subplot(232), plt.imshow(image_rgb), plt.title('RGB Image')
    plt.subplot(233), plt.imshow(image_hsv), plt.title('HSV Image')
    plt.subplot(234), plt.imshow(image_gray, cmap='gray'), plt.title('Gray Image')
    plt.subplot(235), plt.imshow(image_binary, cmap='binary'), plt.title('Binary Image')

    plt.tight_layout()
    plt.show()

if __name__ == "__main__":
    main()
```

Mendeteksi Warna Merah pada Gambar Menggunakan HSV

Di sini kita akan melihat cara menggunakan ruang warna HSV untuk mendeteksi dan mengisolasi warna tertentu, dalam hal ini, merah, dalam suatu gambar. Ruang warna HSV seringkali lebih disukai daripada ruang warna RGB untuk tugas-tugas seperti deteksi warna karena memisahkan warna (hue) dari intensitas (nilai). Hal ini memudahkan penentuan rentang warna tertentu dan melakukan operasi seperti segmentasi warna.

mask = cv2.inRange(image_hsv, lower_red, upper_red): cv2.inRange() membuat masker yang mengisolasi area merah pada gambar berdasarkan rentang yang ditentukan.

result = cv2.bitwise_and(image, image, mask=mask): Masker diterapkan ke gambar asli menggunakan cv2.bitwise_and() untuk mempertahankan hanya area merah.

Contoh Deteksi Warna

```
import cv2
import matplotlib.pyplot as plt
import numpy as np

def main():
    # Memuat gambar
    image = cv2.imread('img/parrot.jpg')

    # convert the color from BGR to RGB for displaying using matplotlib
    image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    image_hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    lower_red = np.array([0, 120, 70])
    upper_red = np.array([10, 255, 255])
    lower_green = np.array([40, 40, 40])
    upper_green = np.array([80, 255, 255])
    lower_blue = np.array([100, 50, 50])
    upper_blue = np.array([130, 255, 255])
    mask = cv2.inRange(image_hsv, lower_green, upper_red)
    result = cv2.bitwise_and(image, image, mask=mask)
    image_red = cv2.cvtColor(result, cv2.COLOR_BGR2RGB)
    plt.figure(figsize=(15, 10))

    plt.subplot(131), plt.imshow(image_rgb), plt.title('RGB Image')
    plt.subplot(132), plt.imshow(image_hsv), plt.title('HSV Image')
    plt.subplot(133), plt.imshow(image_red), plt.title('Red Color Image')

    plt.tight_layout()
    plt.show()

if __name__ == "__main__":
    main()
```