

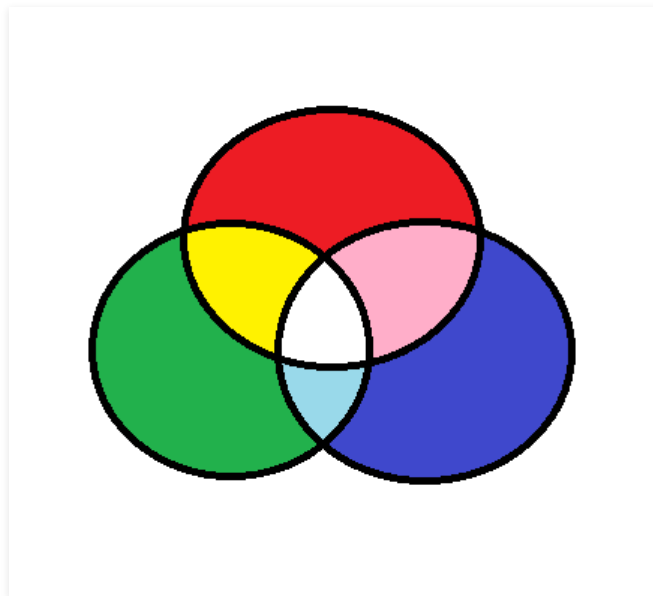
Color Spaces in OpenCV | Python

Difficulty Level : Medium • Last Updated : 06 May, 2019

Color spaces are a way to represent the color channels present in the image that gives the image that particular hue. There are several different color spaces and each has its own significance.

Some of the popular color spaces are *RGB* (Red, Green, Blue), *CMYK* (Cyan, Magenta, Yellow, Black), *HSV* (Hue, Saturation, Value), etc.

BGR color space: OpenCV's default color space is RGB. However, it actually stores color in the BGR format. It is an additive color model where the different intensities of Blue, Green and Red give different shades of color.



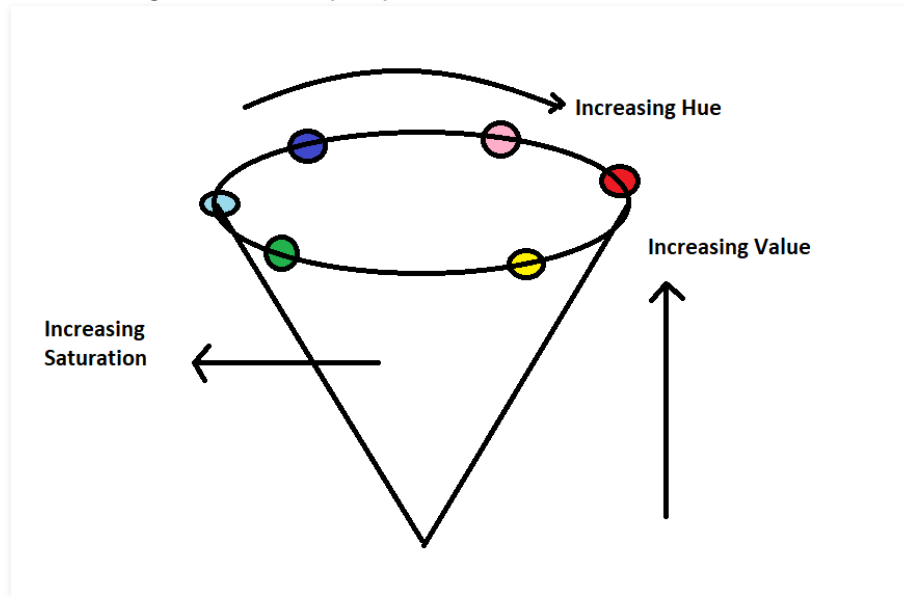
Attention geek! Strengthen your foundations with the [Python Programming Foundation](#) Course and learn the basics.



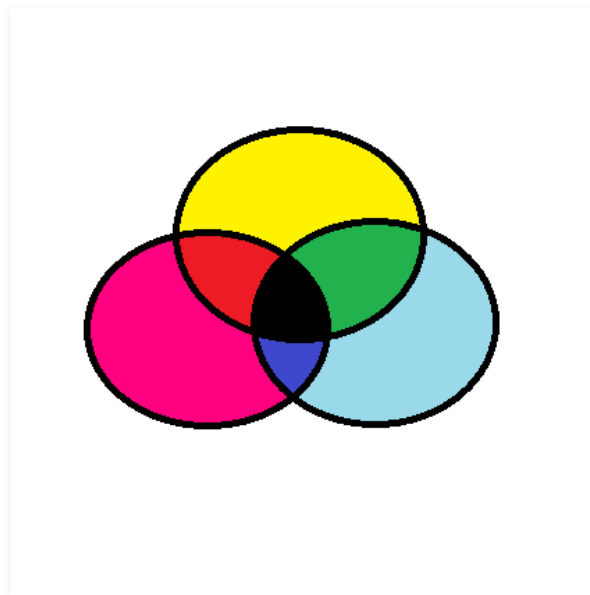
To begin with, your interview preparations Enhance your Data Structures concepts with the [Python DS](#) Course. And to begin with your Machine Learning Journey, join the

Machine Learning - Basic Level Course

HSV color space: It stores color information in a cylindrical representation of RGB color points. It attempts to depict the colors as perceived by the human eye. Hue value varies from 0-179, Saturation value varies from 0-255 and Value value varies from 0-255. It is mostly used for color segmentation purpose.



CMYK color space: Unlike, RGB it is a subtractive color space. The CMYK model works by partially or entirely masking colors on a lighter, usually white, background. The ink reduces the light that would otherwise be reflected. Such a model is called subtractive because inks "subtract" the colors red, green and blue from white light. White light minus red leaves cyan, white light minus green leaves magenta, and white light minus blue leaves yellow.



Visualizing the different color channels of an RGB image.

```
import cv2

image = cv2.imread('C://Users//Gfg//rgb.png')
B, G, R = cv2.split(image)
# Corresponding channels are seperated

cv2.imshow("original", image)
cv2.waitKey(0)

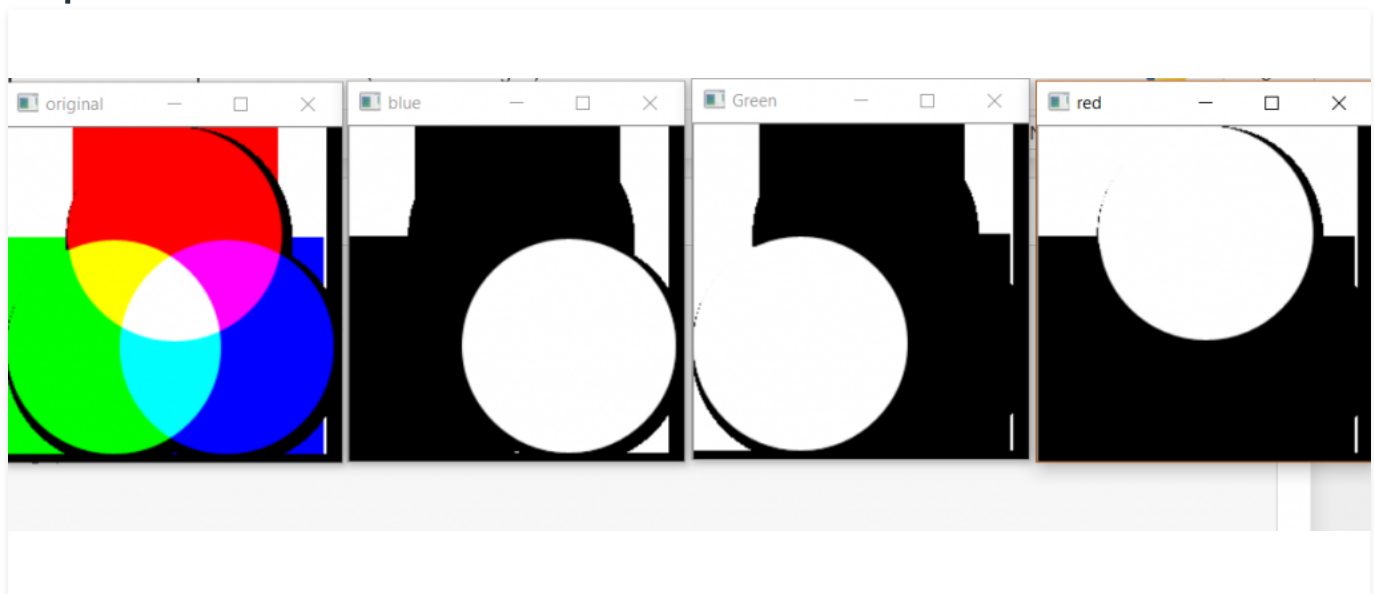
cv2.imshow("blue", B)
cv2.waitKey(0)

cv2.imshow("Green", G)
cv2.waitKey(0)

cv2.imshow("red", R)
cv2.waitKey(0)

cv2.destroyAllWindows()
```

Output:



Like 9

Next