**Color segmentation:**

Color segmentation is a method to separate out objects from the image based on its different color intensities from other objects in the image.

Color space:

RGB:

The most commonly used color space is **RGB( Red, Green, Blue )** which is also known as additive color space as these primary colors can add up with different intensities to produce different colors.

HSV:

HSV stands for **Hue, Saturation and Brightness**. It is used to define contrast in the image, hence it is more localized in nature.

Let us compare the localization of an image in both color space:

Python code:

#import required librabies

import cv2

import matplotlib.pyplot as plt

import numpy as np

from mpl\_toolkits.mplot3d import Axes3D

from matplotlib import cm

from matplotlib import colors

flower1 = cv2.imread('dahlia.jpg');

#visualization in RGB color space

b, g, r = cv2.split(flower1); #bydefault openCV reads image in bgr format

fig = plt.figure();

axis = fig.add\_subplot(1, 1, 1, projection="3d"); #for 3D projection

#normalization

color = flower1.reshape((np.shape(flower1)[0]\*np.shape(flower1)[1], 3));

norm = colors.Normalize(vmin=-1,vmax=1);

norm.autoscale(color);

color = norm(color).tolist();

axis.scatter(b.flatten(), g.flatten(), r.flatten(), facecolors=color, marker=".");

axis.set\_xlabel("Blue");

axis.set\_ylabel("Green");

axis.set\_zlabel("Red");

plt.show();

#visulization in HSV color space

flower2 = cv2.cvtColor(flower1, cv2.COLOR\_RGB2HSV);

h, s, v = cv2.split(flower2);

fig = plt.figure();

axis = fig.add\_subplot(1, 1, 1, projection="3d");

axis.scatter(h.flatten(), s.flatten(), v.flatten(), facecolors=color, marker=".");

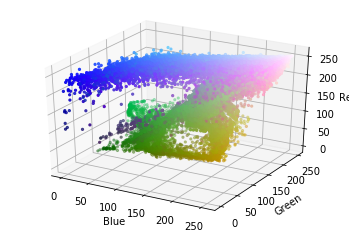
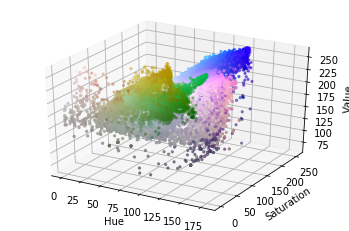
axis.set\_xlabel("Hue");

axis.set\_ylabel("Saturation");

axis.set\_zlabel("Value");

plt.show();

Output:

**Color segmentation process:**

1. Define range of color to be segmented( thresholding ).
2. Create a mask according to the threshold values.
3. Perform masking on the image using AND operation.

Python code:

import cv2

import numpy as np

flower = cv2.imread('dahlia.jpg',1);

#conversion of BGR color space to HSV color space

hsv = cv2.cvtColor(flower, cv2.COLOR\_BGR2HSV);

#step1: thresholding

lower = np.array([0, 30, 0]);

upper = np.array([200, 200, 200]);

#step2: masking

mask = cv2.inRange(hsv, lower, upper);

output\_img = cv2.bitwise\_and(flower, flower, mask = mask);

cv2.imshow('Original', flower);

cv2.imshow('Applied mask', mask);

cv2.imshow('Output', output\_img);

#wait for 10 sec

cv2.waitKey(10000);

cv2.destroyAllWindows();

Output:

