

The Sparks Foundation-GRIP-Data science and Business Analytics-AUG21

Task by:Deepti Taram

Task 4

Domain:Computer Vision and IoT - Color Identification in Images

Problem Statement: To identify the colours in the Image

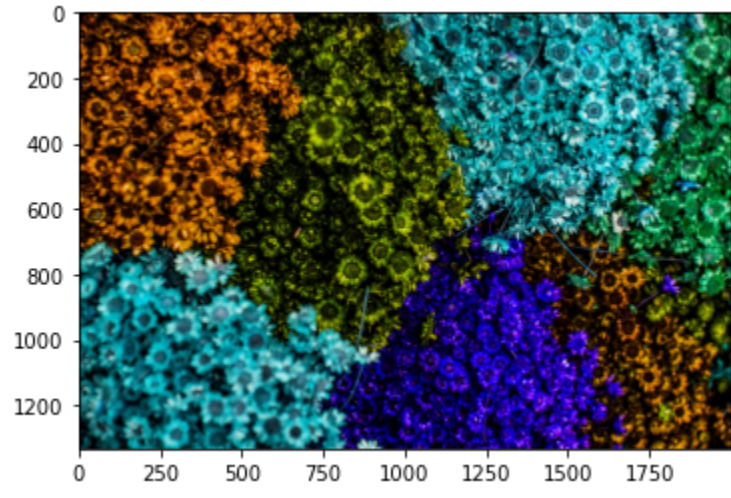
```
In [1]: !pip install opencv-python
```

```
In [2]: ## Importing relevant libraries
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import numpy as np
import cv2
from collections import Counter
from skimage.color import rgb2lab, deltaE_cie76
import os

%matplotlib inline
```

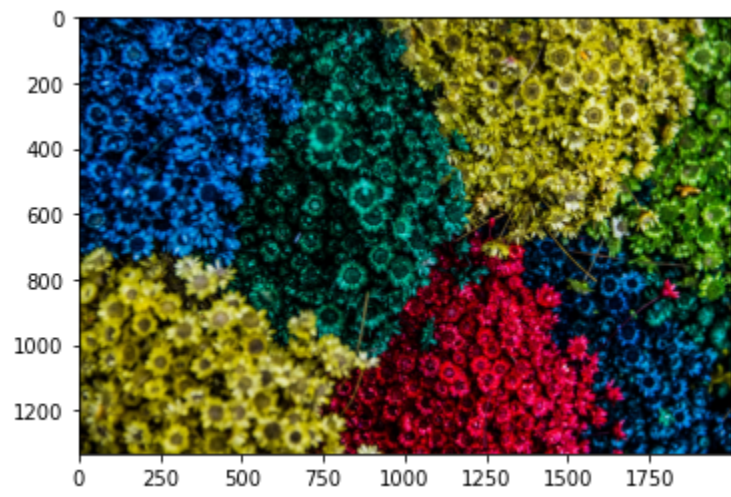
```
In [3]: image = cv2.imread("C:/Users/geetesh taram/Downloads/basis of open cv/unsplash.jpg")
type(image)
print(type(image))
print(image.shape)
plt.imshow(image)
```

```
Out[3]: <matplotlib.image.AxesImage at 0x23f8e98da90>
```



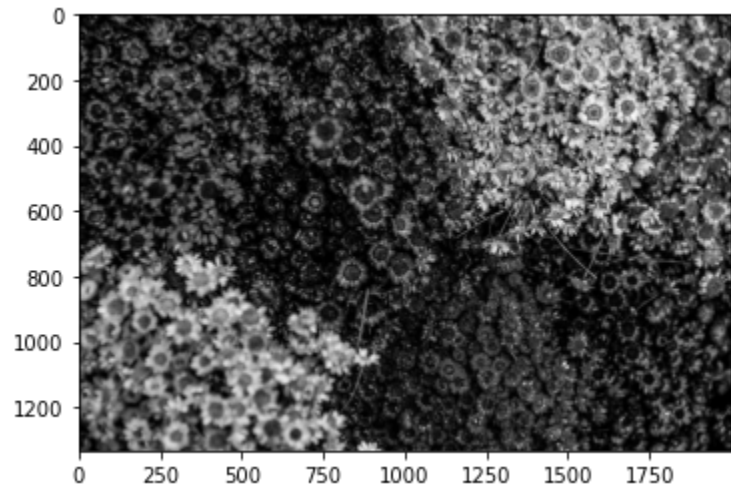
As we can see the color of image is looking odd. It is because OpenCV library reads the image in BGR format

```
In [4]: #Converting the image format to RGB
img= cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
plt.imshow(img)
plt.show()
```



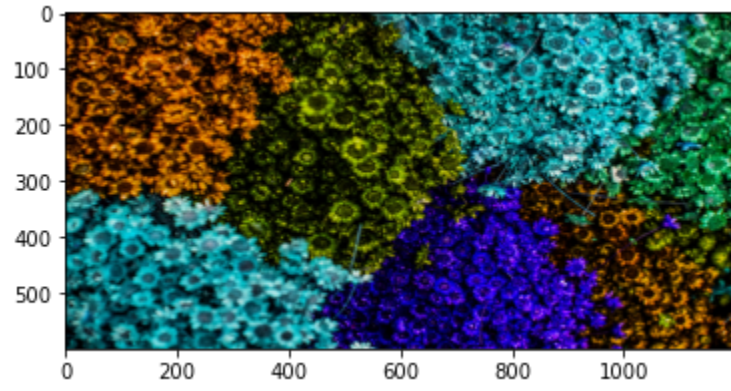
```
In [5]: gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
plt.imshow(gray_image, cmap='gray')
```

```
Out[5]: <matplotlib.image.AxesImage at 0x23f8ed6ed30>
```



```
In [6]: resized_image = cv2.resize(image, (1200, 600))
plt.imshow(resized_image)
```

```
Out[6]: <matplotlib.image.AxesImage at 0x23f8f065430>
```



Color Identification

RGB to Hex Conversion

We'd first define a function that will convert RGB to hex so that we can use them as labels for our pie chart.

```
In [7]: #Importing relevant libraries to convert hex code colors, rgb, color names, etc. and also the metrics
from webcolors import rgb_to_name, hex_to_name, hex_to_rgb
import webcolors
from sklearn.metrics import mean_squared_error
```

Read image in RGB color space

```
In [8]: def get_image(path):
    img= cv2.imread(path)
    img= cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    return img
```

```
In [9]: #Extracting HEX code from RGB format
def rgb_to_hex(color):
    return "#{:02x}{:02x}{:02x}".format(int(color[0]), int(color[1]), int(color[2]))
```

```
In [10]: #Converting the HEX code into name of color
def hex2name(color):
    h_color= rgb_to_hex(color)
    try:
        nm = hex_to_name(h_color, spec='css3')
    except ValueError as v_error:
        #print("{}".format(v_error))
        rms_lst = []
        for img_clr, img_hex in webcolors.CSS2_NAMES_TO_HEX.items():
            cur_clr = hex_to_rgb(img_hex)
            rmse = np.sqrt(mean_squared_error(color, cur_clr))
            rms_lst.append(rmse)

        closest_color = rms_lst.index(min(rms_lst))

        nm = list(webcolors.CSS2_NAMES_TO_HEX.items())[closest_color][0]
    return nm
```

We supply the path of the image as the argument . First, we read the file using imread and then change its color space before returning it.

Get colors from an image

```
In [11]: #Function to cluster the colors present in the image and plotting a pie chart
def colors(image, no_of_colors, show_chart):

    modified_image = cv2.resize(image, (220, 220), interpolation = cv2.INTER_AREA)
    modified_image = modified_image.reshape(modified_image.shape[0]*modified_image.shape[1], 3)
    clf = KMeans(n_clusters = no_of_colors)
    labels = clf.fit_predict(modified_image)
    counts = Counter(labels)

    center_colors = clf.cluster_centers_
    # We get ordered colors by iterating through the keys
    ordered_colors = [center_colors[i] for i in counts.keys()]
    hex_colors = [rgb_to_hex(ordered_colors[i]) for i in counts.keys()]
    rgb_colors = [(ordered_colors[i]) for i in counts.keys()]
    names= [(hex2name(ordered_colors[i]), rgb_to_hex(ordered_colors[i])) for i in counts.keys()]

    if (show_chart):
        plt.figure(figsize=(12,15))
        plt.subplot(131)
        plt.imshow(image)
        plt.subplot(133)
        plt.pie(counts.values(), labels = names, colors = hex_colors)
        plt.show()

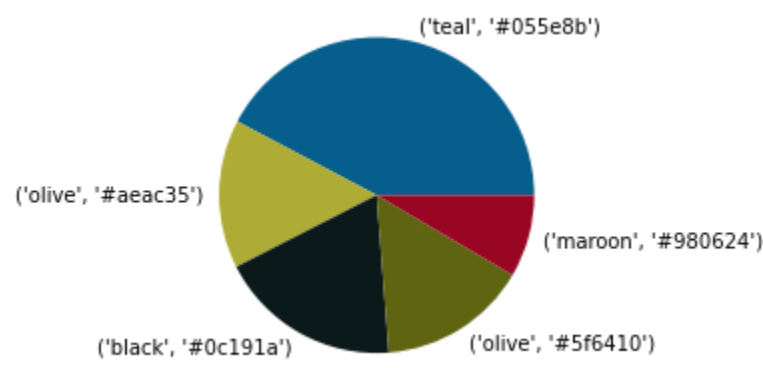
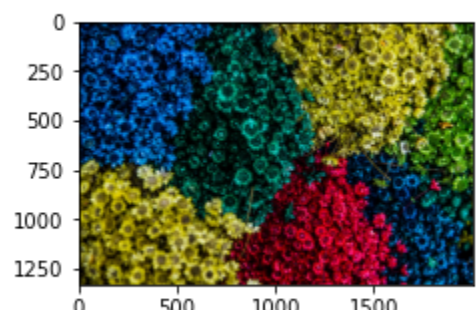
    return rgb_colors
```

Prediction on a images

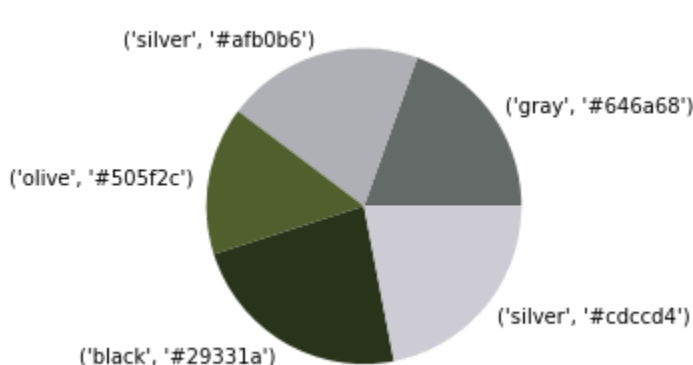
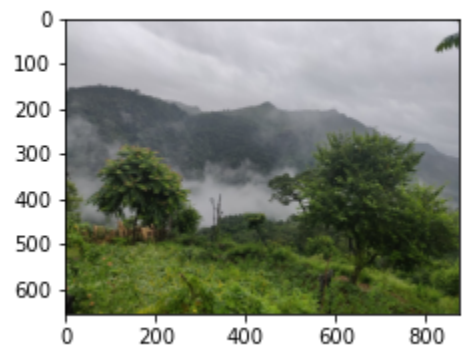
```
In [12]: img1= 'C:/Users/geetesh taram/Downloads/basis of open cv/unsplash.jpg'
img2= 'C:/Users/geetesh taram/Downloads/basis of open cv/nature.jpeg'

images= [img1, img2]
for i in images:
    print(i)
    colors(get_image(i), 5, True)
```

C:/Users/geetesh taram/Downloads/basis of open cv/unsplash.jpg



C:/Users/geetesh taram/Downloads/basis of open cv/nature.jpeg



```
In [ ]:
```