

```
In [3]: #Importing the libraries

import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import os
%matplotlib inline
```

## Open and display the image “dog.jpeg.”

```
In [4]: image = Image.open('dogs.jpeg')
image
```

Out[4]:



## Find out the dimensions of the image and convert it to a two-dimensional array.

```
In [5]: # converting the image into 3d array
img_np = np.asanyarray(image)
```

```
In [6]: #checking the image dimensions
img_np.shape
```

Out[6]: (185, 272, 3)

```
In [7]: # converting the 3d array into 2d
pixels = img_np.reshape(img_np.shape[0]*img_np.shape[1], img_np.shape
[2])
pixels[0:5]
```

```
Out[7]: array([[240, 240, 240],
               [255, 255, 255],
               [255, 255, 255],
               [251, 251, 251],
               [255, 255, 255]], dtype=uint8)
```

```
In [8]: pixels.shape
```

```
Out[8]: (50320, 3)
```

## Use K-means clustering with k set to 3 and cluster the image.

```
In [9]: from sklearn.cluster import KMeans
```

```
In [10]: model = KMeans(n_clusters=3)
          model.fit(pixels)
```

```
Out[10]: KMeans(n_clusters=3)
```

```
In [11]: model_centroid = model.labels_
          cluster_center = model.cluster_centers_
```

```
In [12]: cluster_center
```

```
Out[12]: array([[210.90454397, 175.13378555, 144.0112782 ],
                [ 29.60878353,  21.66133173,  20.54312558],
                [251.52151489, 247.77665976, 243.31041108]])
```

## Predict the cluster label of every pixel in the image and plot it back as an image.

```
In [13]: final = np.zeros((model_centroid.shape[0],3))
```

```
In [14]: final
```

```
Out[14]: array([[0., 0., 0.],
                [0., 0., 0.],
                [0., 0., 0.],
                ...,
                [0., 0., 0.],
                [0., 0., 0.],
                [0., 0., 0.]])
```

```
In [15]: for cluster_no in range(3):
          final[model_centroid==cluster_no]=cluster_center[cluster_no]
```

```
In [16]: final[0:5]
```

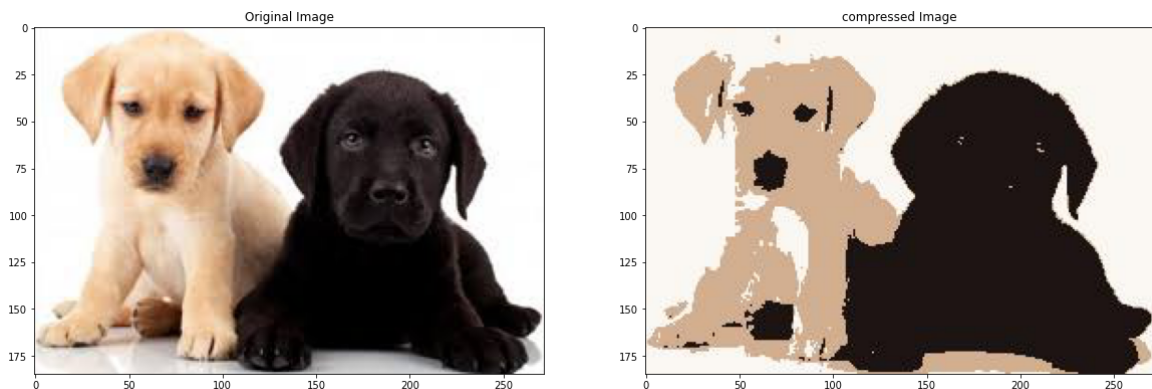
```
Out[16]: array([[251.52151489, 247.77665976, 243.31041108],
                [251.52151489, 247.77665976, 243.31041108],
                [251.52151489, 247.77665976, 243.31041108],
                [251.52151489, 247.77665976, 243.31041108],
                [251.52151489, 247.77665976, 243.31041108]])
```

```
In [17]: comp_image= final.reshape(img_np.shape[0], img_np.shape[1],3)
         comp_image.shape
```

```
Out[17]: (185, 272, 3)
```

```
In [18]: comp_image = Image.fromarray(np.uint8(comp_image))
         comp_image.save('dog_compressed.jpeg')
         img_1 = mpimg.imread('dogs.jpeg')
         img_2 = mpimg.imread('dog_compressed.jpeg')
```

```
In [19]: fig, (ax1,ax2) = plt.subplots(1,2,figsize=(20,20))
         ax1.imshow(img_1)
         ax1.set_title('Original Image')
         ax2.imshow(img_2)
         ax2.set_title('compressed Image')
         plt.show()
```



## Find out the three dominant colors in the image.

```
In [20]: cluster_center
```

```
Out[20]: array([[210.90454397, 175.13378555, 144.0112782 ],
                [ 29.60878353,  21.66133173,  20.54312558],
                [251.52151489, 247.77665976, 243.31041108]])
```

```
In [21]: for centers in cluster_center:
         print(centers.astype(int))
```

```
[210 175 144]
[ 29  21  20]
[251 247 243]
```

```
In [22]: centers = [centers.astype(int) for centers in cluster_center]
```

```
In [34]: print(centers[0])
print(centers[1])
print(centers[2])
```

```
[210 175 144]
[29 21 20]
[251 247 243]
```

```
In [26]: blob = np.ones((100,100,3))
```

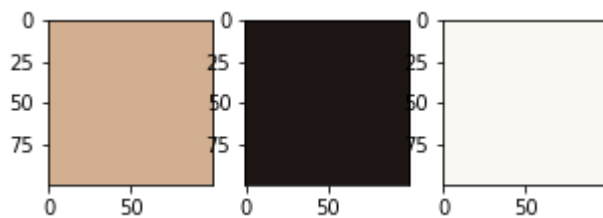
```
In [29]:
```

```
In [41]: blob1 = blob * centers[0]
blob1 = Image.fromarray(np.uint8(blob1))
blob1.save('blob1.jpeg')
```

```
In [40]: blob2 = blob * centers[1]
blob2 = Image.fromarray(np.uint8(blob2))
blob2.save('blob2.jpeg')
```

```
In [39]: blob3 = blob * centers[2]
blob3 = Image.fromarray(np.uint8(blob3))
blob3.save('blob3.jpeg')
```

```
In [38]: fig, (ax1,ax2,ax3) = plt.subplots(1,3, figsize=(5,5))
ax1.imshow(blob1)
ax2.imshow(blob2)
ax3.imshow(blob3)
plt.show()
```



```
In [ ]:
```

```
In [ ]:
```