

## Ex 5: Cluster Color (Bài làm thêm)

```
In [0]: # import the necessary packages
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import cv2
import numpy as np
```

```
In [0]: image_path = './dog_1.jpg'
number_cluster = 4
```

```
In [0]: def centroid_histogram(clt):
    # grab the number of different clusters and create a histogram
    # based on the number of pixels assigned to each cluster
    numLabels = np.arange(0, len(np.unique(clt.labels_)) + 1)
    (hist, _) = np.histogram(clt.labels_, bins = numLabels)

    # normalize the histogram, such that it sums to one
    hist = hist.astype("float")
    hist /= hist.sum()

    # return the histogram
    return hist

def plot_colors(hist, centroids):
    # initialize the bar chart representing the relative frequency
    # of each of the colors
    bar = np.zeros((50, 300, 3), dtype = "uint8")
    startX = 0

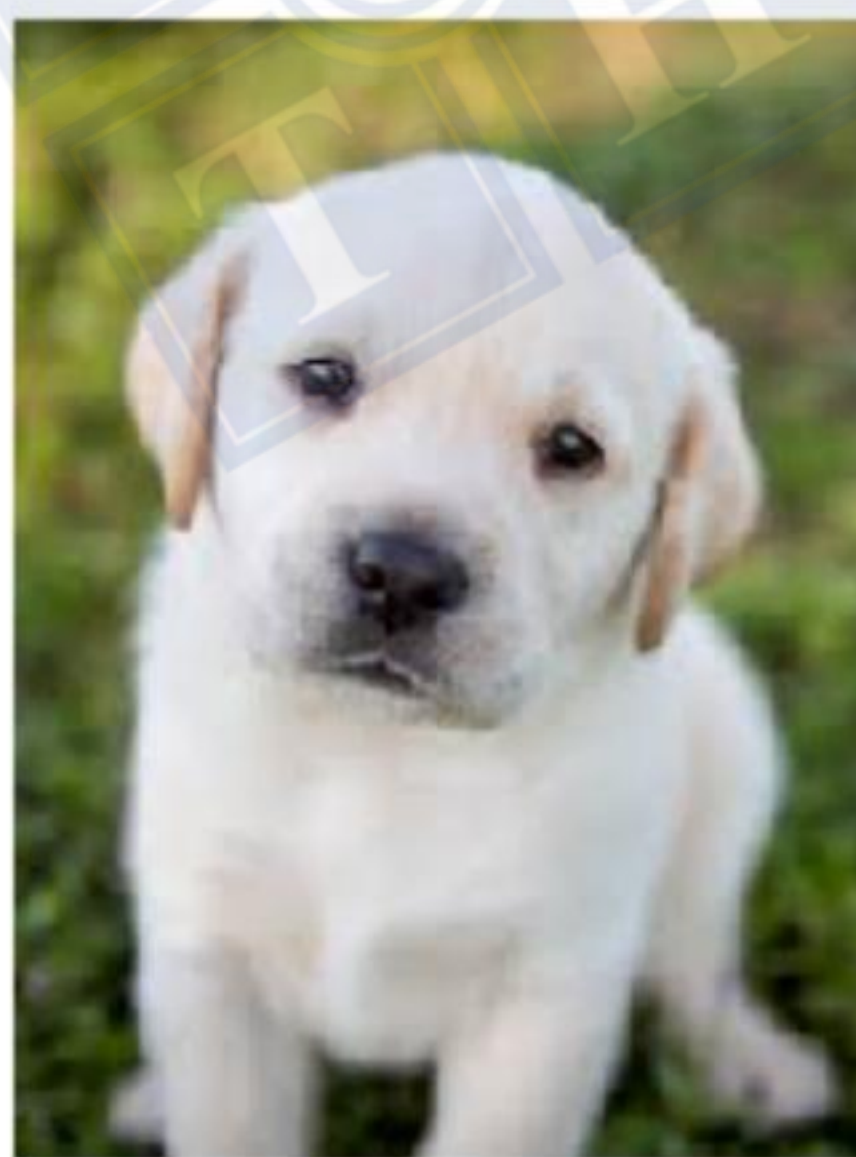
    # loop over the percentage of each cluster and the color of
    # each cluster
    for (percent, color) in zip(hist, centroids):
        # plot the relative percentage of each cluster
        endX = startX + (percent * 300)
        cv2.rectangle(bar, (int(startX), 0), (int(endX), 50),
            color.astype("uint8").tolist(), -1)
        startX = endX

    # return the bar chart
    return bar
```

```
In [0]: # Load the image and convert it from BGR to RGB so that
# we can display it with matplotlib
image = cv2.imread(image_path)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# show our image
plt.figure()
plt.axis("off")
plt.imshow(image)
```

```
Out[4]: <matplotlib.image.AxesImage at 0x11dbc45d0>
```



```
In [0]: # reshape the image to be a list of pixels
image = image.reshape((image.shape[0] * image.shape[1], 3))
# cluster the pixel intensities
clt = KMeans(n_clusters = number_cluster)
clt.fit(image)
```

```
Out[5]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
n_clusters=4, n_init=10, n_jobs=None, precompute_distances='auto',
random_state=None, tol=0.0001, verbose=0)
```



```
In [0]: # build a histogram of clusters and then create a figure
# representing the number of pixels labeled to each color
hist = centroid_histogram(clt)
bar = plot_colors(hist, clt.cluster_centers_)

# show our color bar
plt.figure()
plt.axis("off")
plt.imshow(bar)
plt.show()
```



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
TRUNG TÂM TIN HỌC