

Assignment 4

```
In [1]: import os
import warnings
warnings.filterwarnings("ignore")
dog_images=r'output folder'

from skimage import filters
from skimage import io,color
from skimage import exposure
import numpy as np

def angle(dx, dy):
    return np.mod(np.arctan2(dy, dx), np.pi)

hist_images = []
labels = []
for index,breed in enumerate(os.listdir(dog_images)):
    image_folder=os.path.join(dog_images,breed)
    for image in os.listdir(image_folder):
        img = io.imread(os.path.join(image_folder,image.strip()))
        img=color.rgb2gray(img)
        sobel_img = angle(filters.sobel_h(img),filters.sobel_v(img))
        hist,_=exposure.histogram(sobel_img, nbins=36)
        hist_images.append(hist/np.sum(hist))
        labels.append(index)
hist_images=np.array(hist_images)
labels=np.array(labels)
```

```
In [4]: from sklearn.decomposition import PCA
model=PCA(2)
pca2 = model.fit_transform(hist_images)
```

```
In [6]: from sklearn.cluster import KMeans,BisectingKMeans,SpectralClustering
from sklearn.metrics import silhouette_score,fowlkes_mallows_score

models={"KMeans Random" :KMeans(init="random",n_clusters=4),
        "KMeans KMeans++":KMeans(init="k-means++",n_clusters=4),
        "BisectingKmeans":BisectingKMeans(init="random",n_clusters=4),
        "SpectralClustering":SpectralClustering(n_clusters=4)}

for method,model in models.items():
    model=model.fit(pca2)
    print(method)
    print("fowlkes  :" +str(fowlkes_mallows_score(labels,model.labels_)))
    print("silhouette :"+str(silhouette_score(pca2,model.labels_)))
    print("\n")
```

KMeans Random
fowlkes :0.303598987692129
silhouette :0.40348069718047996

KMeans KMeans++
fowlkes :0.30350207520189104
silhouette :0.4035559982266343

BisectingKmeans
fowlkes :0.29622051404486477
silhouette :0.3923168310410845

SpectralClustering
fowlkes :0.36025060408323173
silhouette :0.0006807467905450256

```
In [13]: from sklearn.cluster import DBSCAN
db = DBSCAN(eps=0.015, min_samples=2).fit(pca2)
# Number of clusters in labels, ignoring noise if present.
n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
n_noise_ = list(labels).count(-1)

print("Estimated number of clusters: %d" % n_clusters_)
print("Estimated number of noise points: %d" % n_noise_)
```

Estimated number of clusters: 4
Estimated number of noise points: 7

eps=0.015, min_samples=2

```
In [15]: print("fowlkes :" +str(fowlkes_mallows_score(labels,db.labels_)))
print("silhoutte :"+str(silhouette_score(pca2,db.labels_)))
```

fowlkes :1.0
silhoutte :0.5867101875184342

```
In [20]: from sklearn.cluster import AgglomerativeClustering
linkage=["ward", "complete", "average", "single" ]
for link in linkage:
    clustering=AgglomerativeClustering(n_clusters=4,linkage=link)
    pred=clustering.fit(pca2)
    print(link)
    print("fowlkes :"+str(fowlkes_mallows_score(labels,pred.labels_)))
    print(" silhouette :"+str(silhouette_score(pca2,pred.labels_))+"\n")
```

```
ward
folwkes :0.6121318386466018
silhouette :0.3985069045807418
```

```
complete
folwkes :0.7228616701783567
silhouette :0.47104361558104924
```

```
average
folwkes :0.9496324516863133
silhouette :0.5056665915382704
```

```
single
folwkes :0.9883867504826501
silhouette :0.6691299342888389
```

**fowlkes score best to worst : dbscan,agglomerative
single,Average,complete,ward,spectral,random,k-
means+ +,bisecting**

**silhouette score best to worst : agglomerative
single,dbscan,average,complete,k-
means+ +,random,ward,bisecting,,spectral.**

In []: