

```
In [1]: # https://scikit-learn.org/stable/user_guide.html
# Importing required libraries
import os
import numpy as np
from skimage import io, color, exposure, filters
from sklearn.decomposition import PCA
import pandas as pd
from sklearn.cluster import KMeans, BisectingKMeans, SpectralClustering, AgglomerativeClustering
from sklearn.metrics import silhouette_score, fowlkes_mallows_score

import warnings
warnings.filterwarnings("ignore")
```

```
In [2]: crop_folder=r'Cropped'
```

```
In [3]: def edge_histogram(selected_images):
    ims=[]
    dogs=[]
    for idx, path in enumerate(selected_images):
        img_path = os.path.join(crop_folder, path)
        for im in os.listdir(img_path):
            src = os.path.join(img_path, im)
            img = io.imread(src)
            gray_img = color.rgb2gray(img)
            dx, dy = filters.sobel_h(gray_img), filters.sobel_v(gray_img)
            angle_sobel = np.mod(np.arctan2(dy, dx), np.pi)
            hist, bins = exposure.histogram(angle_sobel, nbins=36)
            ims.append(hist/np.sum(hist))
            dogs.append(idx)
    return ims, dogs
ims, dogs=edge_histogram(os.listdir(crop_folder))
```

```
In [4]: ims=np.array(ims)
dogs=np.array(dogs)
```

```
In [5]: model = PCA(2)
ims= model.fit_transform(ims)
```

```
In [6]: Algorithms=[]
fowlkes_score=[]
sil_score =[]
```

```
In [7]: Algorithms=['random', 'k-means++', 'bisectingkmeans', 'Spectralclustering', 'DBSCAN', 'AgglomerativeClustering', 'KMeans']
```

```
In [8]: kmeans = KMeans(n_clusters=4, random_state=42, init="random").fit(ims).labels_  
fowlkes_score.append(fowlkes_mallows_score(dogs,kmeans))  
sil_score.append(silhouette_score(ims,kmeans))
```

```
In [9]: kmeans = KMeans(n_clusters=4, random_state=42, init="k-means++").fit(ims).labe  
fowlkes_score.append(fowlkes_mallows_score(dogs,kmeans))  
sil_score.append(silhouette_score(ims,kmeans))
```

```
In [10]: bisecting = BisectingKMeans(n_clusters=4, random_state=42, init="random").fit(  
fowlkes_score.append(fowlkes_mallows_score(dogs,bisecting))  
sil_score.append(silhouette_score(ims,bisecting))
```

```
In [11]: spectralclust = SpectralClustering(n_clusters=4, random_state=42).fit(ims).lab  
fowlkes_score.append(fowlkes_mallows_score(dogs,spectralclust))  
sil_score.append(silhouette_score(ims,spectralclust))
```

```
In [12]: dbscan = DBSCAN(eps=0.02, min_samples=3).fit(ims).labels_  
fowlkes_score.append(fowlkes_mallows_score(dogs,dbscan))  
sil_score.append(silhouette_score(ims,dbscan))
```

```
In [13]: s_link=AgglomerativeClustering(n_clusters=4, linkage='single').fit(ims).labels_  
fowlkes_score.append(fowlkes_mallows_score(dogs,s_link))  
sil_score.append(silhouette_score(ims,s_link))
```

```
In [14]: c_link=AgglomerativeClustering(n_clusters=4, linkage='complete').fit(ims).labe  
fowlkes_score.append(fowlkes_mallows_score(dogs,c_link))  
sil_score.append(silhouette_score(ims,c_link))
```

```
In [15]: a_link=AgglomerativeClustering(n_clusters=4, linkage='average').fit(ims).label  
fowlkes_score.append(fowlkes_mallows_score(dogs,a_link))  
sil_score.append(silhouette_score(ims,a_link))
```

```
In [16]: w_link=AgglomerativeClustering(n_clusters=4, linkage='ward').fit(ims).labels_  
fowlkes_score.append(fowlkes_mallows_score(dogs,w_link))  
sil_score.append(silhouette_score(ims,w_link))
```

**Eps = 0.02 min samples = 3 to get 4 clusters**

```
In [19]: scores_data = pd.DataFrame({ 'Clustering_Algorithms': Algorithms, 'Fowlkes': f
```

```
In [20]: ## best to worst by fowlkes scores  
scores_data.sort_values(by='Fowlkes',ascending=False)
```

Out[20]:

	Clustering_Algorithms	Fowlkes	Silhouette
5	singlelink	0.492479	0.687195
7	averagelink	0.491603	0.495271
4	DBSCAN	0.488825	0.684086
6	completelink	0.409600	0.437274
3	Spectralclustering	0.352886	0.097402
2	bisectingkmeans	0.315862	0.377617
8	ward	0.312422	0.391462
0	random	0.308022	0.378779
1	k-means++	0.308022	0.378779

```
In [21]: ## best to worst by silhouette scores  
scores_data.sort_values(by='Silhouette',ascending=False)
```

Out[21]:

	Clustering_Algorithms	Fowlkes	Silhouette
5	singlelink	0.492479	0.687195
4	DBSCAN	0.488825	0.684086
7	averagelink	0.491603	0.495271
6	completelink	0.409600	0.437274
8	ward	0.312422	0.391462
0	random	0.308022	0.378779
1	k-means++	0.308022	0.378779
2	bisectingkmeans	0.315862	0.377617
3	Spectralclustering	0.352886	0.097402

In [ ]: