In [1]:
```
import os
from skimage import io,color,exposure,filters
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
import warnings
warnings.filterwarnings("ignore")
def angle(dx, dy):
 return np.mod(np.arctan2(dy, dx), np.pi)
direct = "C:\\Users\\mohan\\Desktop\\Cropped"
breeds=os.listdir(direct)
images = []
dog_breed = []
for index, breed in enumerate(breeds):
    img_path = os.path.join(direct, breed)
    for image in os.listdir(img_path):
        src_path = os.path.join(img_path, image)
        img = io.imread(src_path)
        img = color.rgb2gray(img)
        img = angle(filters.sobel_h(img), filters.sobel_v(img))
        hist, _ = exposure.histogram(img, nbins=36)
        hist = hist / np.sum(hist)  # normalization added
        images.append(hist)
        dog_breed.append(index)
images = np.array(images)
dog_breed = np.array(dog_breed)
dr=PCA(2)
images=dr.fit_transform(images)
```

In [2]:
```
from sklearn.cluster import KMeans, BisectingKMeans, SpectralClustering
kmeans_variants = {
    'Random': {'init': 'random'},
    'KMeans++': {'init': 'k-means++'},
    'BisectingKMeans': {'init': 'random'},
    'SpectralClustering': {}
}
labels_dict = {}
for variant, params in kmeans_variants.items():
    if variant == 'SpectralClustering':
```

```
        clustering = SpectralClustering(n_clusters=4, random_state=42)
    elif variant == 'BisectingKMeans':
        clustering = BisectingKMeans(n_clusters=4, random_state=42)
    else:
        clustering = KMeans(n_clusters=4, random_state=42, **params)
    labels = clustering.fit_predict(images)
    labels_dict[variant] = labels
Random = labels_dict['Random']
kmeans = labels_dict['KMeans++']
bisecting = labels_dict['BisectingKMeans']
spectralclustering = labels_dict['SpectralClustering']
```

In [3]:
```
from sklearn.cluster import DBSCAN
dbscan = DBSCAN(eps=0.02, min_samples=3).fit(images).labels_
print("For DBSCAN clustering with 4 clusters:")
print("    eps = 0.02")
print("    min_samples = 3")
print("were used.")
```

```
For DBSCAN clustering with 4 clusters:
    eps = 0.02
    min_samples = 3
were used.
```

In [4]:
```
from sklearn.cluster import AgglomerativeClustering
linkage_methods = ['single', 'complete', 'average', 'ward']
results = {}
for method in linkage_methods:
    clustering = AgglomerativeClustering(n_clusters=4, linkage=method)
    labels = clustering.fit_predict(images)
    results[method] = labels
Agglomerative_single = results['single']
Agglomerative_complete = results['complete']
Agglomerative_average = results['average']
Agglomerative_ward = results['ward']
```

In [5]:
```
from sklearn.metrics import fowlkes_mallows_score
clustering_methods = {
    'Random': Random,
    'kmeans': kmeans,
```

```
    'bisecting': bisecting,
    'spectralclustering': spectralclustering,
    'dbscan': dbscan,
    'Agglomerative_single': Agglomerative_single,
    'Agglomerative_complete': Agglomerative_complete,
    'Agglomerative_average': Agglomerative_average,
    'Agglomerative_ward': Agglomerative_ward
}
for method_name, labels in clustering_methods.items():
    score = fowlkes_mallows_score(dog_breed, labels)
    print(f'{method_name}: {score}')
```

```
Random: 0.2975270506119972
kmeans: 0.29909666651045397
bisecting: 0.2972809339865332
spectralclustering: 0.31521169137383054
dbscan: 0.4874562084130845
Agglomerative_single: 0.4943827347937293
Agglomerative_complete: 0.36713456124640975
Agglomerative_average: 0.48907882180388346
Agglomerative_ward: 0.32235438804477073
```

In [6]:

```
from sklearn.metrics import silhouette_score
clustering_methods = {
    'Random': Random,
    'kmeans': kmeans,
    'bisecting': bisecting,
    'spectralclustering': spectralclustering,
    'dbscan': dbscan,
    'Agglomerative_single': Agglomerative_single,
    'Agglomerative_complete': Agglomerative_complete,
    'Agglomerative_average': Agglomerative_average,
    'Agglomerative_ward': Agglomerative_ward
}
for method_name, labels in clustering_methods.items():
    score = silhouette_score(images, labels)
    print(f'{method_name}: {score}')
```

```
Random: 0.40305783333055095
kmeans: 0.4055704996457822
bisecting: 0.37036373178557724
spectralclustering: 0.24084357806466253
dbscan: 0.7308506454338473
Agglomerative_single: 0.809639336741712
Agglomerative_complete: 0.4137738008851826
Agglomerative_average: 0.5539758547312447
Agglomerative_ward: 0.3934205143234151
```

In [ ]:
```
'''Rank the methods from the best to the worst for our dataset based on Fowlkes-Mallows index
Agglomerative_single,Agglomerative_average,dbscan,Agglomerative_complete,Agglomerative_ward,spectralclustering,random,bisectin

'''Rank the methods from the best to the worst for our dataset based on Silhouette Coefficient.
Agglomerative_single,dbscan,Agglomerative_average,Agglomerative_complete,Kmean,random,Agglomerative_ward,bisecting,spectralclu
```