Lâm Hàn Vương

MSSV: 14521106

Môn: Máy Học Trong Thị Giác Máy Tính

BÀI TẬP THỰC HÀNH – BÀI 1

CLUSTERING DATA

Tóm Tắt:

- Phân Tích tập dữ liệu digits, face, traffic signs.
- Dùng Phương pháp rút trích đặc trưng: Local Binary Partten, Histogram of oriented Gradient để rút tríc đặc trưng sử dụng cho quá trình cluster
  Sử dụng các phương pháp cluster: Kmeans, Spectral Clustering, DBSCAN, Agglomerative Cluster.
- Quy trình thực hiện:

I. Cài đặt thuật toán

1. Dataset Handwriten-digits

a. Dataset Digits

- Dataset Digits được hỗ trợ trên sklearn.datasets gồm 1797 ảnh với kích thước (8, 8) bao gồm số đếm có giá trị từ 0 đến 9 được khởi tạo random dựa trên các labels được đánh các label cho mỗi ảnh
- Load Dataset từ thư viện:

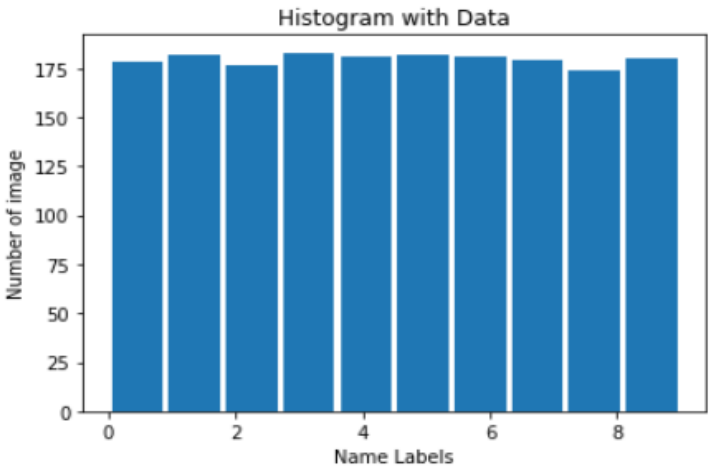
```
digits = load_digits()

image, labels = digits.data, digits.target
```

- Với image là danh sách các anh, labels là nhãn tương ứng với mỗi ảnh, tập dataset gồm:

0	1	2	3	4	5	6	7	8	9
178	182	177	183	181	182	181	179	174	180

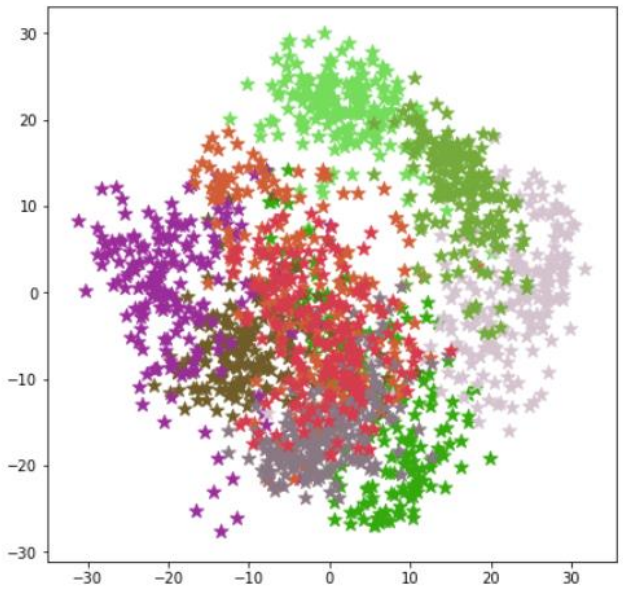
- Được biểu diễn trên lược đồ histogram:

```
import matplotlib.pyplot as plt
%matplotlib inline
hist, bins = np.histogram(labels, bins=10)
width = 0.9 * (bins[1] - bins[0])
center = (bins[:-1] + bins[1:]) / 2
bins = np.array([0, 1, 2, 3, 4 ,5 ,6 ,7 ,8, 9])
plt.bar(center, hist, align='center', width=width)
plt.title("Histogram with Data")
plt.xlabel('Name Labels')
plt.ylabel('Number of image')
plt.show()
```



- Biểu diễn trên ma trận 2D – giảm chiều ảnh sử dụng phương thức PCA:

```
plt.show()
plt.figure(figsize=(7, 7))
for i in range(0,9):
    plt.scatter(train_reduced[labels==i,0], train_reduced[labels ==i,1], s=100, c=colors[i], marker='*', label='cluster 1')
```



b. Clustering data

- Khởi tạo các phương thức từ sklearn:

```
km = KMeans(n_clusters=max(labels)+1,init='random',n_init=100,max_iter=200,tol=1e-04,random_state=0)
sc = SpectralClustering(n_clusters=max(labels)+1,eigen_solver='arpack',affinity="nearest_neighbors", assign_labels= 'discretize')
ac = AgglomerativeClustering(n_clusters=max(labels)+1, affinity='euclidean')
db = DBSCAN(eps=20, min_samples=2)
```

- Clustering data với y_km, y_sc, y_ac lần lược là giá trị labels tương ứng với sử dụng thuật giải kmeans, SpectralClustering, AgglomerativeClustering, DBSCAN:

```
y_km=km.fit_predict(image)
y_sc=sc.fit_predict(image)
y_ac = ac.fit_predict(image)
y_db = db.fit_predict(image)
```

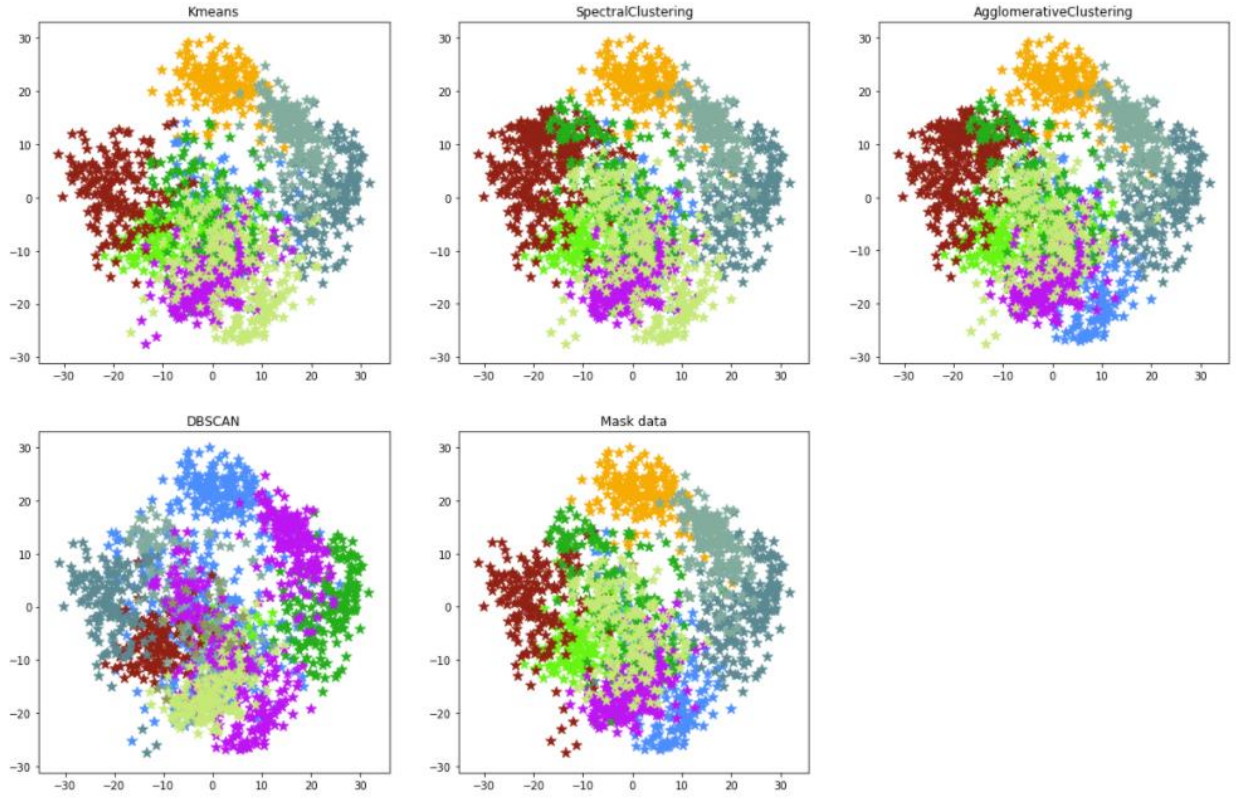
```
        indexvalue = np.where(parameter==i)
        a = indexvalue
        value = itemgetter(*indexvalue)(labels)
        value= collections.Counter(value).most_common(1)[0][0]
        index_list_km[indexvalue] = value
    return index_list_km
```

```
y_km = ValidValue(y_km)
y_sc = ValidValue(y_sc)
y_ac = ValidValue(y_ac)
```

c. Visualize kết quả

- Kết quả so với giữa 4 thuật toán và labels:

```
plt.show()
plt.figure(figsize=(20,20))
for i in range(0,max(labels)):
    plt.subplot(331)
    plt.title('Kmeans')
    plt.scatter(train_reduced[y_km==i,0], train_reduced[y_km ==i,1], s=100, c=colors[i], marker='*', label='cluster 1')
    plt.subplot(332)
    plt.title('SpectralClustering')
    plt.scatter(train_reduced[y_sc==i,0], train_reduced[y_sc ==i,1], s=100, c=colors[i], marker='*', label='cluster 1')
    plt.subplot(333)
    plt.title('AgglomerativeClustering')
    plt.scatter(train_reduced[y_ac==i,0], train_reduced[y_ac ==i,1], s=100, c=colors[i], marker='*', label='cluster 1')
    plt.subplot(335)
    plt.title('Mask data')
    plt.scatter(train_reduced[labels==i,0], train_reduced[labels ==i,1], s=100, c=colors[i], marker='*', label='cluster 1')
colors = []
for i in range(-1,max(y_db)):
    colors.append('#%06X' % randint(0, 0xFFFFFF))
for i in range(-1,max(y_db)):
    plt.subplot(334)
    plt.title('DBSCAN')
    plt.scatter(train_reduced[y_db==i,0], train_reduced[y_db ==i,1], s=100, c=colors[i+1], marker='*')
```



- Tính độ chính xác thuật toán theo metrics:

```
from sklearn import metrics
print("K-mean Accurary: {0:.9f}".format(metrics.accuracy_score(labels, y_km)))
print("SpectralClustering Accurary: {0:.9f}".format(metrics.accuracy_score(labels, y_sc)))
print("AgglomerativeClustering Accurary: {0:.9f}".format(metrics.accuracy_score(labels, y_ac)))
print("DBSCAN: {0:.9f}".format(metrics.accuracy_score(labels, y_db)))
```

```
K-mean Accurary: 0.792431831
SpectralClustering Accurary: 0.820812465
AgglomerativeClustering Accurary: 0.861992209
DBSCAN: 0.728436283
```

- Tính độ chính xác thuật toán theo Recall, Precision, f1-score, Support

```
from sklearn import metrics
print("Classification Report")
print("Kmeans: ")
print(metrics.classification_report(labels,y_km, labels=[0,1,2,3,4,5,6,7,8,9]))
print("SpectralClustering: ")
print(metrics.classification_report(labels,y_sc, labels=[0,1,2,3,4,5,6,7,8,9]))
print("AgglomerativeClustering: ")
print(metrics.classification_report(labels,y_ac, labels=[0,1,2,3,4,5,6,7,8,9]))
```

```
Kmeans:
             precision    recall  f1-score   support

          0       0.99      0.99      0.99       178
          1       0.59      0.30      0.40       182
          2       0.85      0.84      0.84       177
          3       0.88      0.84      0.86       183
          4       0.98      0.90      0.94       181
          5       0.91      0.75      0.82       182
          6       0.97      0.98      0.98       181
          7       0.85      0.98      0.91       179
          8       0.45      0.58      0.51       174
          9       0.56      0.77      0.65       180

avg / total       0.80      0.79      0.79      1797

SpectralClustering:
             precision    recall  f1-score   support

          0       1.00      1.00      1.00       178
          1       0.71      0.46      0.56       182
          2       0.99      0.94      0.97       177
          3       0.53      0.93      0.67       183
          4       1.00      0.98      0.99       181
          5       0.98      0.98      0.98       182
          6       0.99      0.99      0.99       181
          7       0.94      1.00      0.97       179
          8       0.58      0.93      0.71       174
          9       0.00      0.00      0.00       180

avg / total       0.77      0.82      0.78      1797
```

AgglomerativeClustering:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	178
1	0.87	0.85	0.86	182
2	0.84	0.94	0.89	177
3	0.53	0.92	0.68	183
4	1.00	0.98	0.99	181
5	0.99	0.98	0.99	182
6	0.99	0.99	0.99	181
7	0.91	1.00	0.95	179
8	0.86	0.95	0.90	174
9	0.00	0.00	0.00	180
avg / total	0.80	0.86	0.83	1797

DB:

	precision	recall	f1-score	support
0	0.40	1.00	0.58	178
1	1.00	0.14	0.25	182
2	1.00	0.89	0.94	177
3	1.00	0.83	0.90	183
4	1.00	0.92	0.96	181
5	1.00	0.80	0.89	182
6	0.43	0.96	0.60	181
7	1.00	0.84	0.91	179
8	1.00	0.13	0.22	174
9	1.00	0.77	0.87	180
avg / total	0.88	0.73	0.71	1797

2. Dataset Face

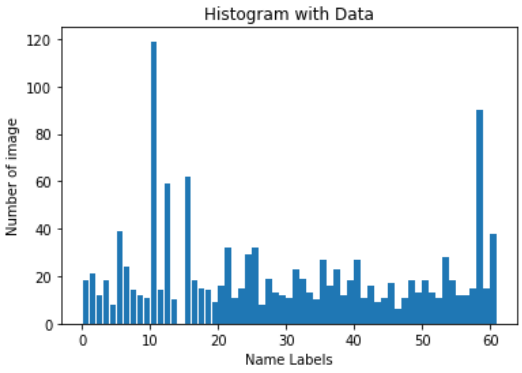
a. Dataset fetch_lfw-people

- Dataset Fetch_lfw-people gồm: 34799 ảnh được chia thành 13233 labels, để việc cluster data đạc hiệu quả nên loại bỏ các labels có tập hợp ảnh nhỏ hơn 20 faces:

```
digits = fetch_lfw_people(data_home=None, funneled=True, resize=1, min_faces_per_person=20 , color=False, slice_=(slice(70, 195, None), slice(78, 172, None)), download_if_missing=True)

image, labels = digits.data, digits.target
```

- Dữ liệu được hiển thị trên Histogram:

```
import matplotlib.pyplot as plt
%matplotlib inline
hist, bins = np.histogram(labels, bins=max(labels))
width = 0.9 * (bins[1] - bins[0])
center = (bins[:-1] + bins[1:]) / 2
plt.bar(center, hist, align='center', width=width)
plt.title("Histogram with Data")
plt.xlabel('Name Labels')
plt.ylabel('Number of image')
plt.show()
```



b. Cluster data

- Khởi tạo phương thức cluster

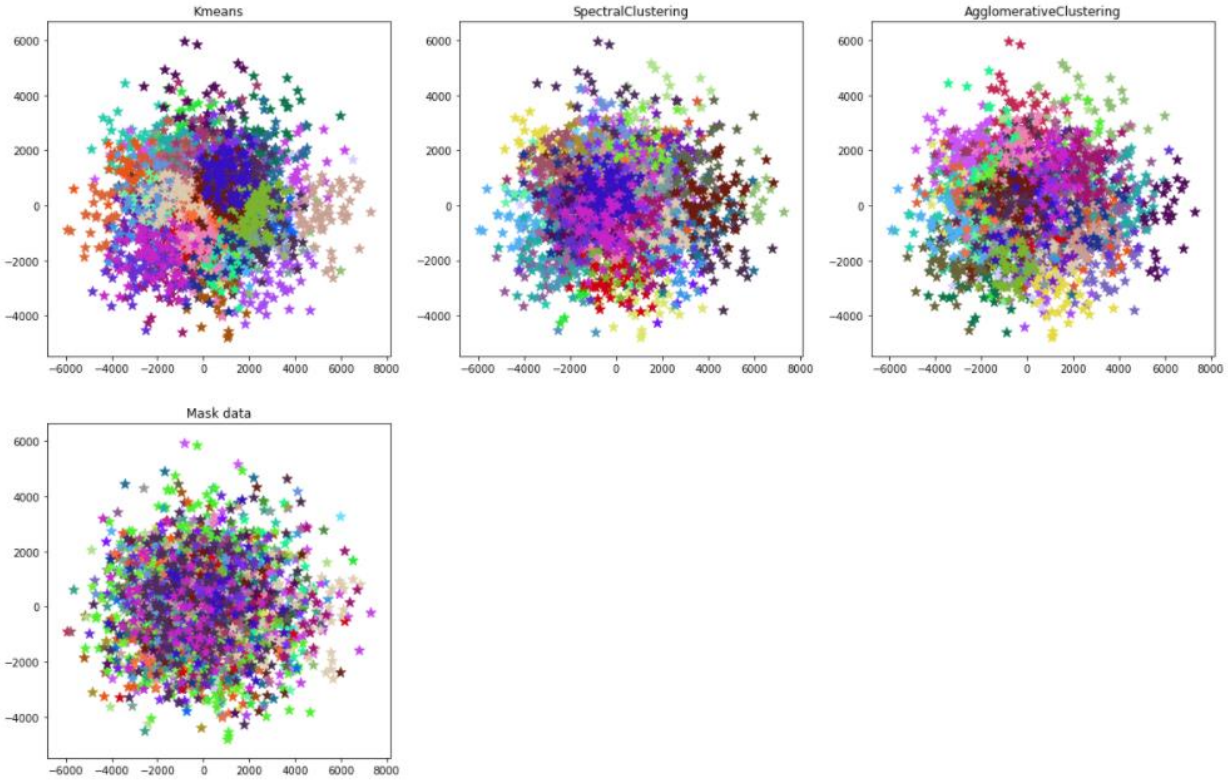
```
km = KMeans(n_clusters=max(labels)+1,init='random',n_init=100,max_iter=200,tol=1e-04,random_state=0)
sc = SpectralClustering(n_clusters=max(labels)+1,eigen_solver='arpack',affinity="nearest_neighbors", assign_labels= 'discretize')
ac = AgglomerativeClustering(n_clusters=max(labels)+1, affinity='euclidean')
```

Clustering data

```
y_km=km.fit_predict(image)
y_sc=sc.fit_predict(image)
y_ac = ac.fit_predict(image)
```

```
ds = DBSCAN(eps=0.1, min_samples=8)
y_ds = ds.fit_predict(image)
max(y_ds)
```

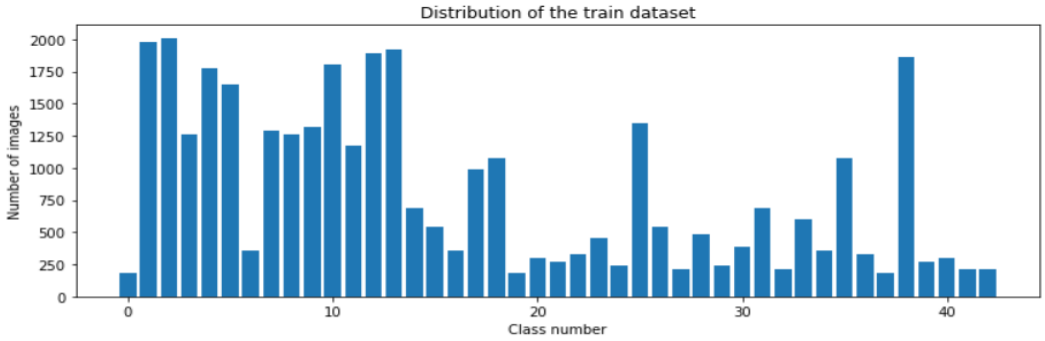
c. Visualize kết quả



3. Dataset GTSRB

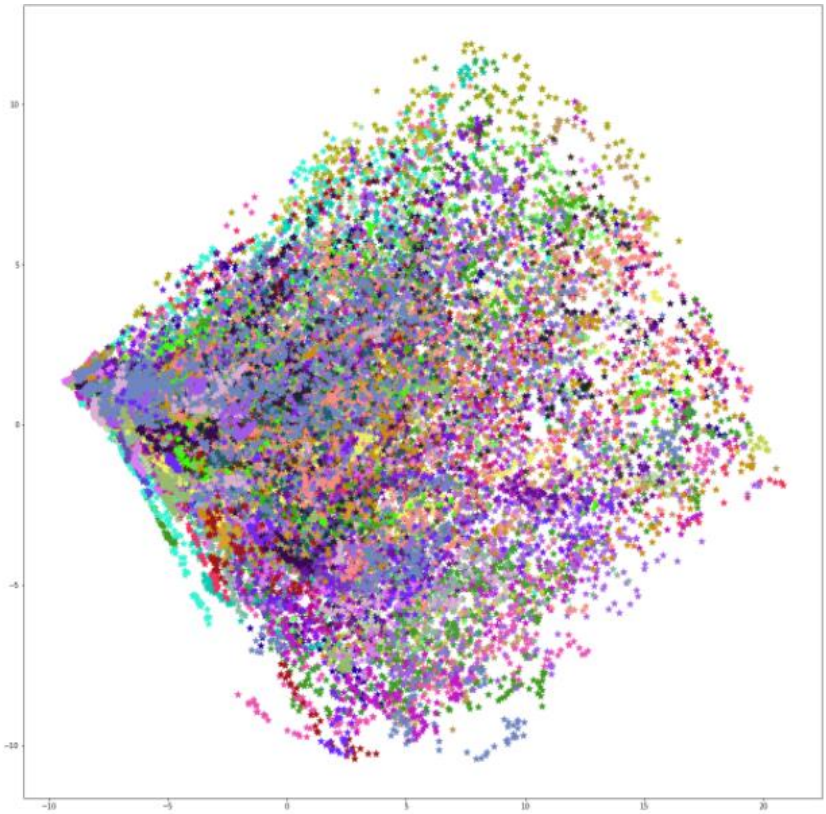
a. GTSRB

- Dataset gồm 34799 samples chia thành 43 labels. Tương ứng trung bình mỗi labels sẽ có 3 sample, hiển trị trên lược đồ histogram




```
Min number of images per class = 180
Max number of images per class = 2010
```

- Được biểu diễn trên đồ thị 2 chiều:



- Sử dụng Histogram of oriented gradient (HoG) để rút trích đặc trưng cho dữ liệu:

```
hogs=[]
for i in range(0,len(y_train)):
    hogs.append(hog(dataset[i], orientations=3, pixels_per_cell=(16,16),cells_per_block=(1, 1)))
```

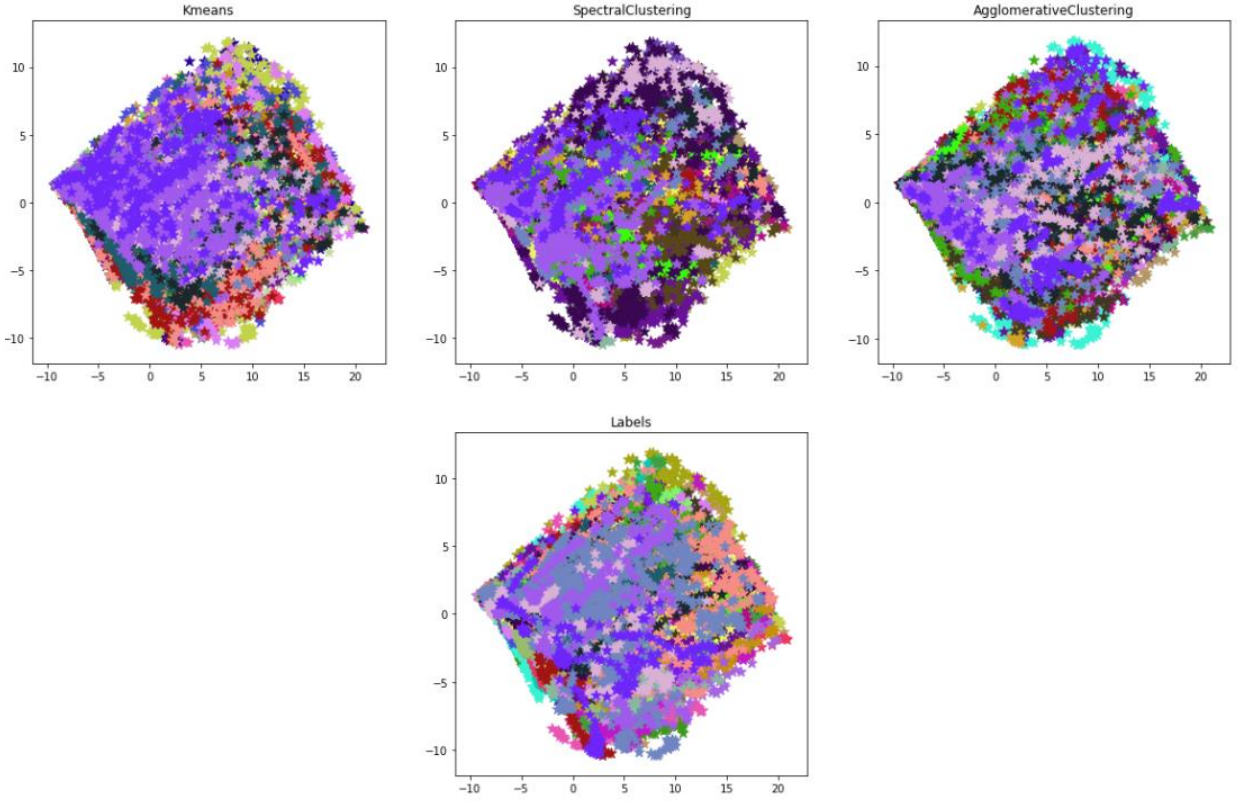
b. Visualize kết quả

- Phương thức rút trích đặc trưng

```
km = KMeans(n_clusters=max(y_train),init='random',n_init=100,max_iter=200,tol=1e-04,random_state=0)
sc = SpectralClustering(n_clusters=max(y_train), eigen_solver='arpack', affinity="nearest_neighbors")
ac = AgglomerativeClustering(n_clusters=max(y_train), affinity='euclidean')
db = DBSCAN(eps=17.5, min_samples=1)
```

```
y_km=km.fit_predict(hogs)
y_sc=sc.fit_predict(hogs)
y_ac = ac.fit_predict(hogs)
y_db = db.fit_predict(hogs)
```

- Kết quả cluster:



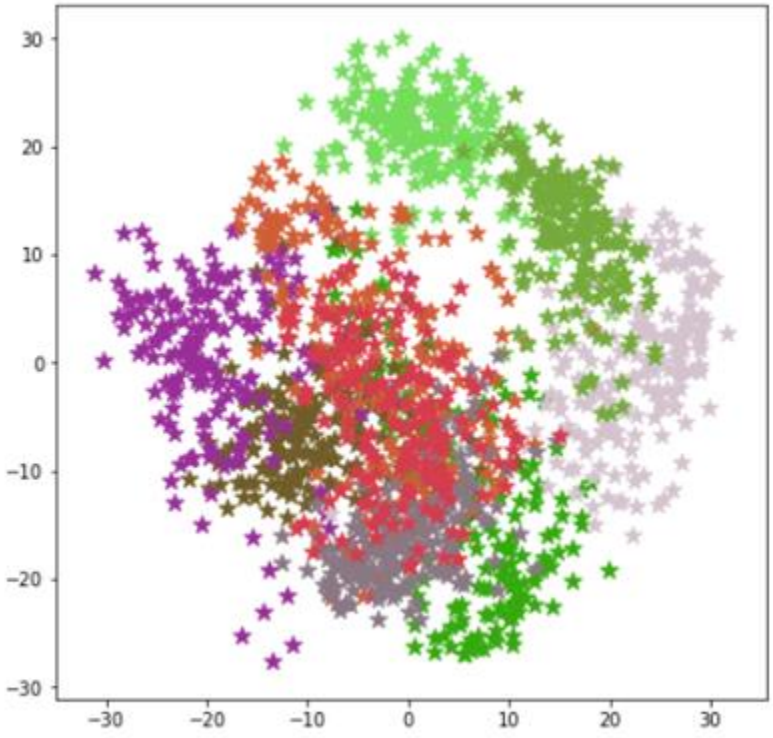
- Độ chính xác so với labels:

```
from sklearn import metrics
print(metrics.adjusted_rand_score(y_km,y_train))
print(metrics.adjusted_rand_score(y_sc,y_train))
print(metrics.adjusted_rand_score(y_ac,y_train))
print(metrics.adjusted_rand_score(y_db,y_train))
```

```
0.104785471657
0.072487435982
0.102930250129
0.0
```

II. Dánh giá thuật toán

- Các thuật toán cluster: kmeans, spectralCluster, AggolativeCluster phân lớp phụ thuộc vào số cluster đầu vào để phân lớp, và phục thuộc vào phương thức đo của dữ liệu.
- Quá trình phân lớp dựa trên đặc trưng của dữ liệu, dữ liệu có tính đặc trưng riêng rỏ rang thì kết quả phân lớp có khả năng chính xác cao và ngược lại.
- Xét dữ liệu Digits, dữ liệu mẫu có các đặc trưng rỏ ràng và ít nhầm lẫn vào nhau:



- Kết quả quá trình phân lớp có khả năng chính xác cao hơn tập dữ liệu face, và traffic signs, các thuật toán rút trích đặc trưng đưa ra các đặc trưng không có tính có ràng nên thuật toán kmeans, spectral, agglomerative đưa ra kết quả không chính xác so với labels.