Thị giác máy tính nâng cao

CS331

Bài tập 1

Nội dung

- Sử dụng dataset Caltech101 để thực hiện phân loại
- Tỷ lệ chia tập train:val: test = 60%:20%:20%

Nội dung

- Yêu cầu 1
- Áp dụng thuật toán KNN, tính Accuracy trên tập val và testvới các giá trị k lần lượt:1,2,3,4,5

k	1	2	3	4	5
Acc_val					
Acc_test					

Nội dung

- Yêu cầu 2
- Áp dụng thuật toán SVC (với tham số C=1),
- Tính Accuracy cho tập test
- Tính Precision của từng lớp (class)

	precision	recall	f1-score	support
0	1.00	0.95	0.98	83
1	0.18	0.96	0.30	83
2	0.65	1.00	0.79	52
3	0.65	0.88	0.75	176
4	0.33	0.50	0.40	8
5	0.32	0.93	0.48	137
6	0.00	0.00	0.00	7
7	0.00	0.00	0.00	11
8	0.00	0.00	0.00	8
9	0.00	0.00	0.00	6

Nội dung

- Yêu cầu 3
- Áp dụng thuật toán SVC (với tham số C=1),
  - Sử dụng chiến lược 1 vs 1:
    - Tính Accuracy cho tập test
    - Tính số support vector
    - Tính và hiển thị ma trận Confusion

Nội dung

- Yêu cầu 4
- Áp dụng thuật toán SVC (với tham số C=1),
  - Sử dụng chiến lược 1 vs rest:
    - Tính Accuracy cho tập test
    - Tính số support vector
    - Tính và hiển thị ma trận Confusion

Một số gợi ý

- Import các thư viện
  - import torch
  - import torchvision
  - from torchvision import datasets
  - from imutils import paths
  - from tqdm import tqdm
  - from sklearn.model_selection import train_test_split
  - from sklearn.preprocessing import LabelEncoder

  - import os
  - import cv2
  - import numpy as np

Một số gợi ý

- Download dataset

```
data_path='./'
dataset =  datasets.Caltech101(data_path, download=True)
```

https://notebook.community/tensorflow/docs-l10n/site/en-snapshot/datasets/overview

Một số gợi ý

- Extract Feature histogram

```
image_paths = list(paths.list_images('./caltech101'))

data = []
labels = []
for img_path in tqdm(image_paths):
    label = img_path.split(os.path.sep)[-2]
    if label == "BACKGROUND_Google":
        continue
    img = cv2.imread(img_path)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    hist = cv2.calcHist([img], [0], None, [256], [0, 256])
    size = img.shape[0]*img.shape[1]
    hist = hist/size

    data.append(hist)
    labels.append(label)

data = np.array(data)
labels = np.array(labels)
```

Một số gợi ý

- Encode label

```
lb = LabelEncoder()
labels = lb.fit_transform(labels)
print(f"Total Number of Classes: {len(lb.classes_)}")
```

Một số gợi ý

- Split train, val, test

```
train_ratio = 0.6
validation_ratio = 0.2
test_ratio = 0.2

x_train, x_test, y_train, y_test = train_test_split(data, labels, test_size=1 - train_ratio)
x_val, x_test, y_val, y_test = train_test_split(x_test, y_test,
test_size=test_ratio/(test_ratio + validation_ratio))

x_train = x_train.reshape(len(x_train), 256)
x_val = x_val.reshape(len(x_val), 256)
x_test = x_test.reshape(len(x_test), 256)
print(len(x_train))
print(len(x_val))
print(len(x_test))
```

Một số gợi ý

- KNN

```
best_score = 0
best_k = 0
for i in range(1,6):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(x_train, y_train)
    score = knn.score(x_val, y_val)
    if score > best_score:
        best_score = score
        best_k = i
    print(f'KNN with k={i}: {score}')
print(f"Best score = {best_score} with k = {best_k}")
```