Fruit Reconition

December 10, 2021

[15]:

```
!pip3 install opencv-python
```

```
Collecting opencv-python
  Downloading opencv_python-4.5.3.56-cp38-cp38-macosx_10_15_x86_64.whl (42.6 MB)
     |                  | 42.6 MB 1.5 MB/s eta 0:00:01
Requirement already satisfied: numpy>=1.17.3 in
/Library/Frameworks/Python.framework/Versions/3.8/lib/python3.8/site-packages
(from opencv-python) (1.19.2)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.5.3.56
WARNING: You are using pip version 20.2.1; however, version 21.2.4 is

available.

You should consider upgrading via the

'/Library/Frameworks/Python.framework/Versions/3.8/bin/python3.8 -m pip install

--upgrade pip' command.
```

[24]:

```python
# cv2 is for re-sizing fruits pictures
def GetFruit(fruitlist,datatype):
    images=[]
    labels=[]
    val=['train','test']
    PATH="./fruits-360/"+datatype+"/"
    for i,fruit in enumerate(fruitlist):
        location=PATH+fruit
        data_number=0
        for image_path in glob.glob(os.path.join(location,"*.jpg")):
            image=cv2.imread(image_path,cv2.IMREAD_COLOR)
            image=cv2.resize(image,(dim,dim))
            image=cv2.cvtColor(image,cv2.COLOR_RGB2BGR)
            images.append(image)
            labels.append(i)
            data_number+=1
        print(datatype.upper(),fruitlist[i].upper(),":",data_number)
    images=np.array(images)
    labels=np.array(labels)
    return images,labels
```

[29]:

```python
import cv2
import glob
import numpy as np
import os
```

[54]:

```python
def Fruit_Dectetion(fruitlist):
    # Get image and Labels
    dim = 100
    # by GetFruit function to get dataset for certain kind of fruit
    X_train_raw,y_train=GetFruit(fruitlist,'train')
    X_test_raw,y_test=GetFruit(fruitlist,'test')

    # scaling preprocessing
    from skimage import feature
    from sklearn.preprocessing import StandardScaler
    # Scale data Images
    scaler=StandardScaler()
    X_train=scaler.fit_transform([i.flatten() for i in X_train_raw])
    X_test=scaler.fit_transform([i.flatten() for i in X_test_raw])
    print('--------------------------------')
    print('shape of the training data:',X_train.shape)
    print('shape of the testing data:',X_test.shape)
    print('--------------------------------')

    # linear SVM using hog features
    from sklearn.svm import SVC
    from sklearn import metrics
    svm=SVC(gamma='auto',kernel='linear',probability=True)
    svm.fit(X_train,y_train)
    y_pred=svm.predict(X_test)

    # Evaluation
    precision=metrics.accuracy_score(y_pred,y_test)*100
    print('Accuracy to classify'+' '+fruitlist[0]+' '+'and'+' '+fruitlist[1]+':␣
 ↪{0:.2f}%'.format(precision))
```

[51]:

```python
# choose two fruit to detect and run the detection function to get the
fruitlist=['Pineapple','Mango']
Fruit_Dectetion(fruitlist)
```

```
TRAIN PINEAPPLE : 490
TRAIN MANGO : 490
TEST PINEAPPLE : 166
TEST MANGO : 166
--------------------------------
shape of the training data: (980, 30000)
shape of the testing data: (332, 30000)
```

```
--------------------------------
Accuracy to classify Pineapple and Mango: 100.00%
```

[52]:

```python
fruitlist=['Grape White','Grape Pink']
Fruit_Dectetion(fruitlist)
```

```
TRAIN GRAPE WHITE : 490
TRAIN GRAPE PINK : 492
TEST GRAPE WHITE : 166
TEST GRAPE PINK : 164
--------------------------------
shape of the training data: (982, 30000)
shape of the testing data: (330, 30000)
--------------------------------
Accuracy to classify Grape White and Grape Pink: 100.00%
```

[53]:

```python
fruitlist=['Potato Red','Potato Sweet']
Fruit_Dectetion(fruitlist)
```

```
TRAIN POTATO RED : 450
TRAIN POTATO SWEET : 450
TEST POTATO RED : 150
TEST POTATO SWEET : 150
--------------------------------
shape of the training data: (900, 30000)
shape of the testing data: (300, 30000)
--------------------------------
Accuracy to classify Potato Red and Potato Sweet: 78.67%
```

[ ]: