Fruit Detection

December 12, 2021

[57]:
```
!pip3 install opencv-python
```

```
Requirement already satisfied: opencv-python in
/Library/Frameworks/Python.framework/Versions/3.8/lib/python3.8/site-packages
(4.5.3.56)
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)
WARNING: You are using pip version 20.2.1; however, version 21.3.1 is
available.
You should consider upgrading via the
'/Library/Frameworks/Python.framework/Versions/3.8/bin/python3.8 -m pip install
--upgrade pip' command.
```

[5]:
```
# 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
```

[6]:
```
import cv2
import glob
import numpy as np
import os
dim = 100
from sklearn.svm import SVC
from sklearn import metrics
from sklearn.tree import DecisionTreeClassifier
```

[7]:
```
# Get image and Labels
def Preprecessing(fruitlist):
    # 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('---------------------------------')
    return X_train,X_test,y_train,y_test
```

[8]:
```
def Binary_Fruit_Dectetion(fruitlist):
    X_train,X_test,y_train,y_test = Preprecessing(fruitlist)
    # Import SVM model from sklearn package
    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))
```

[9]:
```
# choose two fruit, run the detection function to predict and get␣
 ↪detection_accuracy
fruitlist=['Pineapple','Mango']
Binary_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%
```

[10]:
```
fruitlist=['Grape White','Grape Pink']
Binary_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%
```

[11]:
```
fruitlist=['Potato Red','Potato Sweet']
Binary_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%
```

[12]:
```
# use Decison_Tree to do multiple fruit detection/classification.
def multiple_Fruit_Detect(fruitlist):
    X_train,X_test,y_train,y_test = Preprecessing(fruitlist)
    tree = DecisionTreeClassifier()
    tree = tree.fit(X_train,y_train)
    y_pred = tree.predict(X_test)
    precision=metrics.accuracy_score(y_pred,y_test)*100
    print("Accuracy to detect"+str(fruitlist)+':'+'{0:.2f}%'.format(precision))
```

[13]:
```
multi_fruitlist = ['Peach','Potato Red','Mango','Pineapple']
multiple_Fruit_Detect(multi_fruitlist)
```

```
TRAIN PEACH : 492
TRAIN POTATO RED : 450
```

```
TRAIN MANGO : 490
TRAIN PINEAPPLE : 490
TEST PEACH : 164
TEST POTATO RED : 150
TEST MANGO : 166
TEST PINEAPPLE : 166
--------------------------------
shape of the training data: (1922, 30000)
shape of the testing data: (646, 30000)
--------------------------------
Accuracy to detect['Peach', 'Potato Red', 'Mango', 'Pineapple']:96.59%
```

[14]:
```
multi_fruitlist = ['Kiwi','Potato Red','Potato Sweet','Pineapple']
multiple_Fruit_Detect(multi_fruitlist)
```

```
TRAIN KIWI : 466
TRAIN POTATO RED : 450
TRAIN POTATO SWEET : 450
TRAIN PINEAPPLE : 490
TEST KIWI : 156
TEST POTATO RED : 150
TEST POTATO SWEET : 150
TEST PINEAPPLE : 166
--------------------------------
shape of the training data: (1856, 30000)
shape of the testing data: (622, 30000)
--------------------------------
Accuracy to detect['Kiwi', 'Potato Red', 'Potato Sweet', 'Pineapple']:79.90%
```

[15]:
```
multi_fruitlist = ['Pepper Green','Pepper Red','Pepper Orange','Pepper Yellow']
multiple_Fruit_Detect(multi_fruitlist)
```

```
TRAIN PEPPER GREEN : 444
TRAIN PEPPER RED : 666
TRAIN PEPPER ORANGE : 702
TRAIN PEPPER YELLOW : 666
TEST PEPPER GREEN : 148
TEST PEPPER RED : 222
TEST PEPPER ORANGE : 234
TEST PEPPER YELLOW : 222
--------------------------------
shape of the training data: (2478, 30000)
shape of the testing data: (826, 30000)
--------------------------------
Accuracy to detect['Pepper Green', 'Pepper Red', 'Pepper Orange', 'Pepper
Yellow']:85.71%
```

[ ]:
```
```

[ ]: