IMAGE CLASSIFICATION USING KNN, SVC, ANN

CODE

```
import cv2
import os
```

```
def extract_color_histogram(image):
  hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
  bins = [10]
  h, s, v = hsv[:, :, 0], hsv[:, :, 1], hsv[:, :, 2]
  hist = cv2.calcHist([h], [0], None, bins, [0, 180])
  cv2.normalize(hist, hist)
  return hist.flatten()
```

```
train_data_path = '/content/drive/MyDrive/JISNIT/Courses/ML/LectureNotes/data/ImageData/FruitData/Training'
test_data_path = '/content/drive/MyDrive/JISNIT/Courses/ML/LectureNotes/data/ImageData/FruitData/Test'
data_dir_list = list(os.listdir(train_data_path))
print(data_dir_list)
features = []
classLabels = []
for dataset in data_dir_list:
    img_list = os.listdir(train_data_path+'/'+ dataset)
    print ('Loaded the images of dataset-'+'{}\n'.format(dataset))
    for img in img_list:
        image = cv2.imread(train_data_path + '/'+ dataset + '/'+ img )
        label = dataset
        hist = extract_color_histogram(image)
        features.append(hist)
        classLabels.append(label)
```

OUTPUT

```
['Apple', 'Banana']
Loaded the images of dataset-Apple

Loaded the images of dataset-Banana
```

CODE

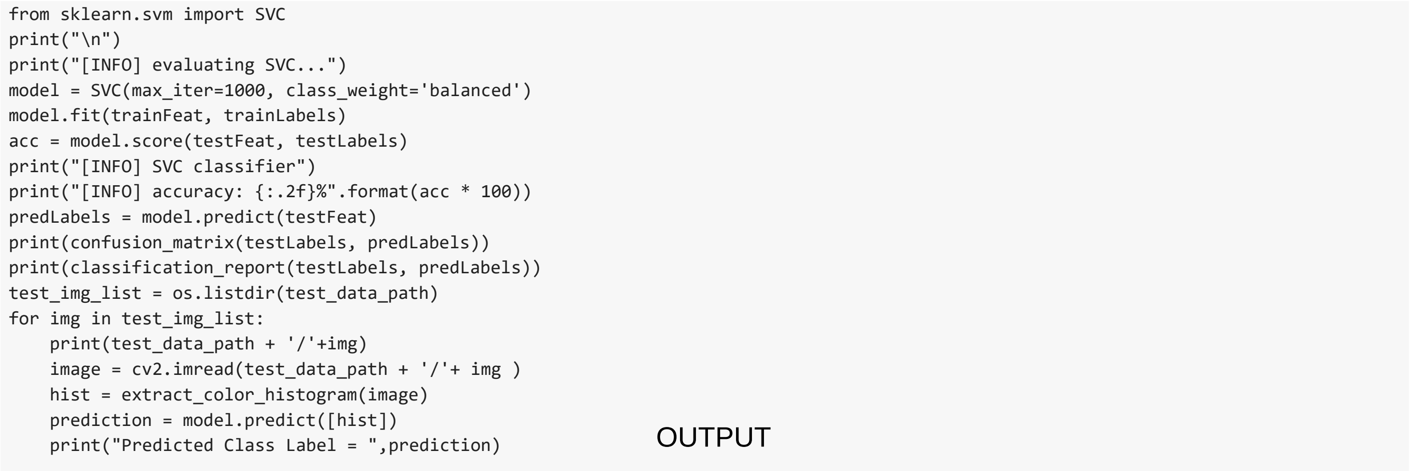
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
trainFeat, testFeat, trainLabels, testLabels =  train_test_split(features, classLabels, test_size=0.20)
```

```
from sklearn.neighbors import KNeighborsClassifier
print("\n")
print("[INFO] evaluating k-NN...")
k = 9
model = KNeighborsClassifier(n_neighbors = k)
model.fit(trainFeat, trainLabels)
acc = model.score(testFeat, testLabels)
print("[INFO] k-NN classifier: k = ", k)
print("[INFO] accuracy: {:.2f}%".format(acc * 100))
predLabels = model.predict(testFeat)
print(confusion_matrix(testLabels, predLabels))
print(classification_report(testLabels, predLabels))
test_img_list = os.listdir(test_data_path)
for img in test_img_list:
    print(test_data_path + '/'+img)
    image = cv2.imread(test_data_path + '/'+ img )
    hist = extract_color_histogram(image)
    prediction = model.predict([hist])
    print("Predicted Class Label = ",prediction)
```

OUTPUT

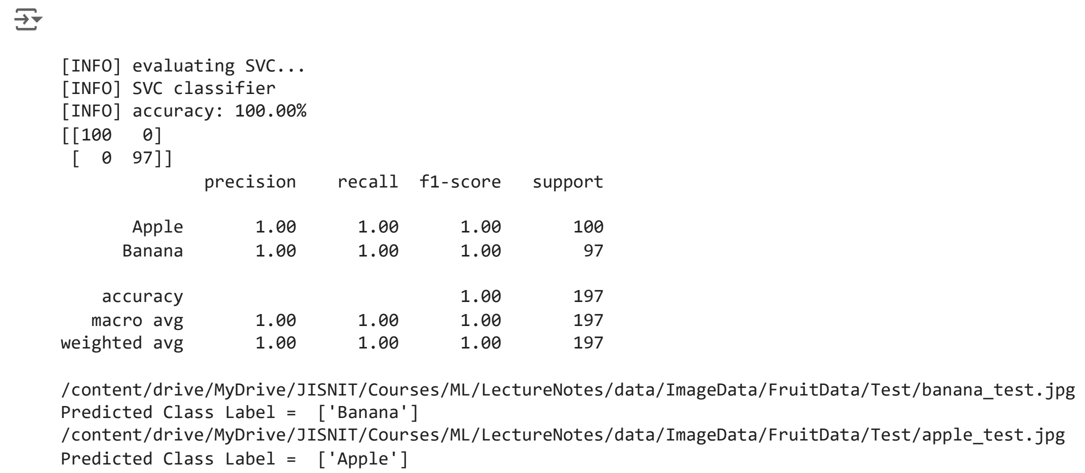
```

[INFO] evaluating k-NN...
[INFO] k-NN classifier: k =  9
[INFO] accuracy: 100.00%
[[100   0]
 [  0  97]]
              precision    recall  f1-score   support

       Apple       1.00      1.00      1.00       100
      Banana       1.00      1.00      1.00        97

    accuracy                           1.00       197
   macro avg       1.00      1.00      1.00       197
weighted avg       1.00      1.00      1.00       197

/content/drive/MyDrive/JISNIT/Courses/ML/LectureNotes/data/ImageData/FruitData/Test/banana_test.jpg
Predicted Class Label =  ['Banana']
/content/drive/MyDrive/JISNIT/Courses/ML/LectureNotes/data/ImageData/FruitData/Test/apple_test.jpg
Predicted Class Label =  ['Apple']
```

CODE



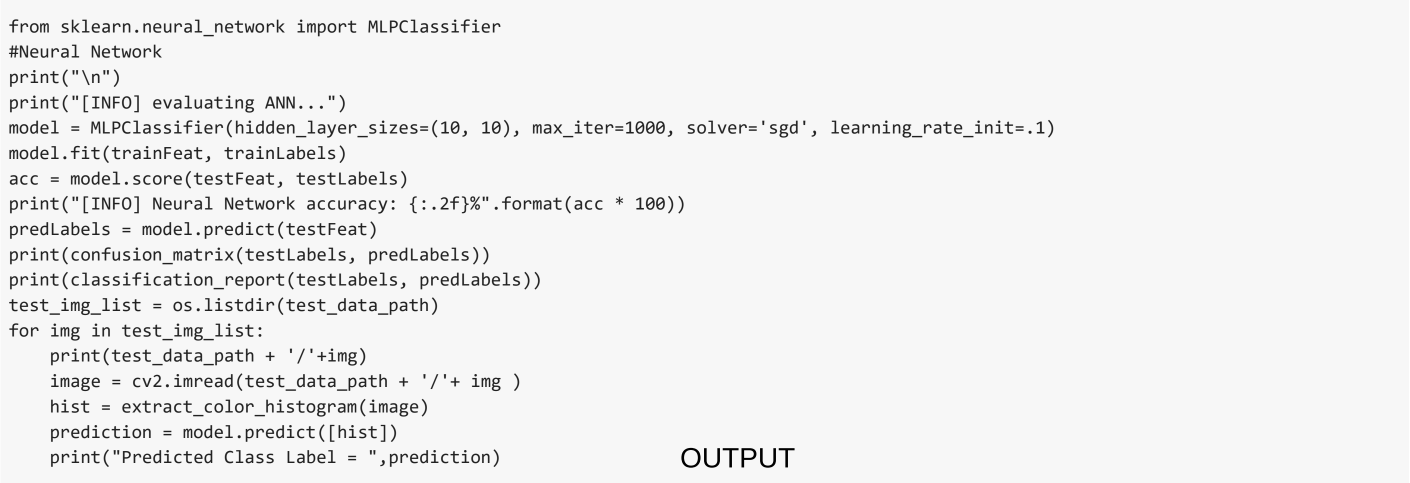
```
from sklearn.svm import SVC
print("\n")
print("[INFO] evaluating SVC...")
model = SVC(max_iter=1000, class_weight='balanced')
model.fit(trainFeat, trainLabels)
acc = model.score(testFeat, testLabels)
print("[INFO] SVC classifier")
print("[INFO] accuracy: {:.2f}%".format(acc * 100))
predLabels = model.predict(testFeat)
print(confusion_matrix(testLabels, predLabels))
print(classification_report(testLabels, predLabels))
test_img_list = os.listdir(test_data_path)
for img in test_img_list:
    print(test_data_path + '/'+img)
    image = cv2.imread(test_data_path + '/'+ img )
    hist = extract_color_histogram(image)
    prediction = model.predict([hist])
    print("Predicted Class Label = ",prediction)
```

OUTPUT



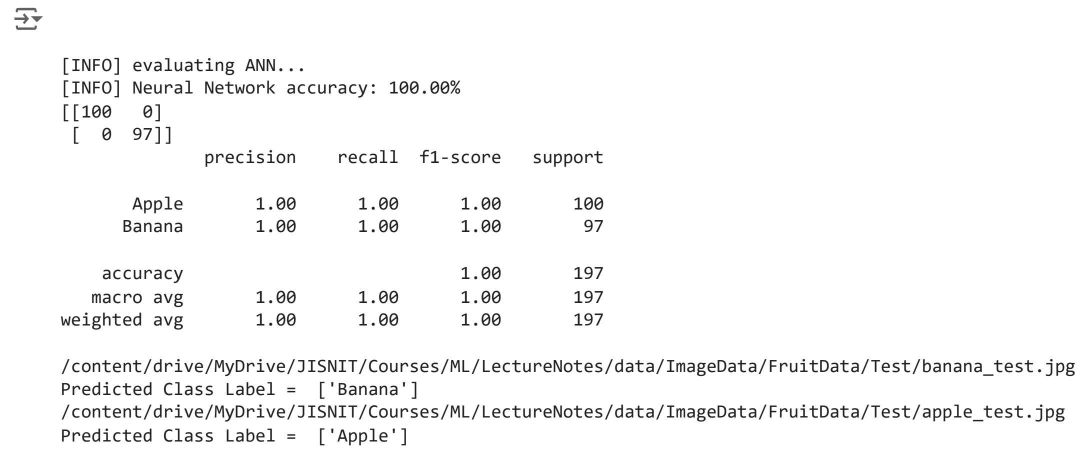
```
[INFO] evaluating SVC...
[INFO] SVC classifier
[INFO] accuracy: 100.00%
[[100   0]
 [  0  97]]
              precision    recall  f1-score   support

       Apple       1.00      1.00      1.00       100
      Banana       1.00      1.00      1.00        97

    accuracy                           1.00       197
   macro avg       1.00      1.00      1.00       197
weighted avg       1.00      1.00      1.00       197

/content/drive/MyDrive/JISNIT/Courses/ML/LectureNotes/data/ImageData/FruitData/Test/banana_test.jpg
Predicted Class Label =  ['Banana']
/content/drive/MyDrive/JISNIT/Courses/ML/LectureNotes/data/ImageData/FruitData/Test/apple_test.jpg
Predicted Class Label =  ['Apple']
```

CODE



```
from sklearn.neural_network import MLPClassifier
#Neural Network
print("\n")
print("[INFO] evaluating ANN...")
model = MLPClassifier(hidden_layer_sizes=(10, 10), max_iter=1000, solver='sgd', learning_rate_init=.1)
model.fit(trainFeat, trainLabels)
acc = model.score(testFeat, testLabels)
print("[INFO] Neural Network accuracy: {:.2f}%".format(acc * 100))
predLabels = model.predict(testFeat)
print(confusion_matrix(testLabels, predLabels))
print(classification_report(testLabels, predLabels))
test_img_list = os.listdir(test_data_path)
for img in test_img_list:
    print(test_data_path + '/'+img)
    image = cv2.imread(test_data_path + '/'+ img )
    hist = extract_color_histogram(image)
    prediction = model.predict([hist])
    print("Predicted Class Label = ",prediction)
```

OUTPUT



```
[INFO] evaluating ANN...
[INFO] Neural Network accuracy: 100.00%
[[100   0]
 [  0  97]]
              precision    recall  f1-score   support

       Apple       1.00      1.00      1.00       100
      Banana       1.00      1.00      1.00        97

    accuracy                           1.00       197
   macro avg       1.00      1.00      1.00       197
weighted avg       1.00      1.00      1.00       197

/content/drive/MyDrive/JISNIT/Courses/ML/LectureNotes/data/ImageData/FruitData/Test/banana_test.jpg
Predicted Class Label =  ['Banana']
/content/drive/MyDrive/JISNIT/Courses/ML/LectureNotes/data/ImageData/FruitData/Test/apple_test.jpg
Predicted Class Label =  ['Apple']
```