

ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



Bài tập 01

MÔN: NHẬN DẠNG

Tp Hồ Chí Minh 2022

ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

NGUYỄN THANH TÙNG

BÀI TẬP 01

Môn: Nhận dạng

MSSV: 20120617

Tên: Nguyễn Thanh Tùng

Email: tungcltt@gmail.com

GIÁO VIÊN

Thầy Lê Hoàng Thái

Tp Hồ Chí Minh 2022

Mục lục

I. Phân tích và cài đặt	2
1) Phân tích bài toán.....	2
2) Kỹ thuật HOG:	2
3) Mô hình SVM đa lớp:	4
4) Huấn luyện mô hình:	5
5) Xuất kết quả:	5

I. Phân tích và cài đặt

1) Phân tích bài toán

- Bài toán nhận diện loại bệnh trên cây là một bài toán phân loại ảnh, trong đó ta cần phân loại các loại bệnh trên cây bao gồm 4 loại: Multiple diseases, Healthy, Rust, Scab.
- Để giải quyết bài toán này, ta sử dụng kỹ thuật HOG (Histogram of Oriented Gradients) để rút trích đặc trưng từ tập dữ liệu hình ảnh và mô hình SVM (Support Vector Machine) để phân tách lớp.

2) Kỹ thuật HOG:

a. Lấy dữ liệu từ file CSV

- Tải file .zip lên google drive sau đó giải nén
- Dùng thư viện **pandas** trong python để đọc các file .csv thành các dataframe
- Đưa các dataframe về các ma trận bằng thư viện **numpy**

y_train: ma trận lưu loại bệnh từ file train.csv

```
y_train
array([[0, 0, 0, 1],
       [0, 1, 0, 0],
       [1, 0, 0, 0],
       ...,
       [1, 0, 0, 0],
       [0, 0, 1, 0],
       [0, 0, 0, 1]])
```

b. Hog

- HOG là viết tắt của Histogram of Oriented Gradient - một loại “feature descriptor”. Mục đích của “feature descriptor” là trừu tượng hóa đối tượng bằng cách trích xuất ra những đặc trưng của đối tượng đó và bỏ đi những thông tin

không hữu ích. Vì vậy, HOG được sử dụng chủ yếu để mô tả hình dạng và sự xuất hiện của một đối tượng trong ảnh.

- Code:

```
train_data = []
for i in range(len(train_df)):
    img_path = 'images/' + train_df['image_id'][i] + '.jpg'
    img = cv2.imread(img_path)
    img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    img_resized = cv2.resize(img_gray, (128, 128))
    hog = cv2.HOGDescriptor()
    hog_feature = hog.compute(img_resized)
    train_data.append(hog_feature.flatten())

X_train = np.array(train_data)
y_train = np.array(train_labels)

test_data = []
test_labels = []
for i in range(len(test_df)):
    img_path = 'images/' + test_df['image_id'][i] + '.jpg'
    img = cv2.imread(img_path)
    img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    img_resized = cv2.resize(img_gray, (128, 128))
    hog = cv2.HOGDescriptor()
    hog_feature = hog.compute(img_resized)
    test_data.append(hog_feature.flatten())

X_test = np.array(test_data)
```

- Rút trích đặc trưng từng ảnh train trong file images, lưu thành ma trận X_train

x_train

```
array([[ -1.3690434 , -0.74682075,  1.96989   , ..., -0.02048721,
         0.6403492 , -0.473394   ],
       [ -0.30164722,  2.2554123 ,  3.6020718 , ...,  0.26276514,
         0.3785587 ,  0.8068598 ],
       [ -0.15153204,  0.16100848, -0.4534161 , ...,  1.4285856 ,
         1.8695781 ,  1.8176979 ],
       ...,
       [  2.2475555 ,  0.39796522, -0.3062217 , ..., -1.0118287 ,
        -1.121625   ,  0.8710902 ],
       [ -0.85664713, -0.9480543 , -0.9861904 , ..., -0.34101415,
        -0.63359594, -0.18366267],
       [ -0.37961882,  0.5974209 ,  0.7448005 , ..., -1.2932298 ,
        -1.4319602 ,  1.8561747 ]], dtype=float32)
```

- Rút trích đặc trưng từng ảnh test trong file images, lưu thành ma trận X_test

3) Mô hình SVM đa lớp:

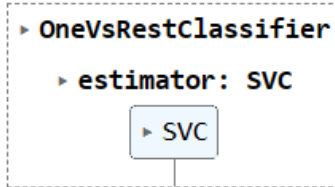
- SVM là một mô hình học máy thuộc nhóm học máy dựa trên phân loại.
- SVM xác định ranh giới phân chia tốt nhất giữa các lớp bằng cách tìm ra siêu phẳng có khoảng cách lớn nhất với các điểm dữ liệu.
- Triển khai mô hình SVM đa lớp bằng phương pháp OnevsRest (OvR): Phương pháp này sử dụng một mô hình SVM cho mỗi lớp, trong đó lớp đang xét được coi là lớp positive, các lớp còn lại được coi là lớp negative. Khi phân loại dữ liệu mới, mô hình SVM của lớp nào cho ra kết quả cao nhất sẽ được chọn để dự đoán lớp của dữ liệu mới.
- Code: huấn luyện trên tập dữ liệu X_train, y_train

```

] from sklearn.svm import SVC
  from sklearn.multiclass import OneVsRestClassifier

svm = OneVsRestClassifier(SVC(kernel='linear', C=1, probability=True))
svm.fit(X_train, y_train)

```



4) Huấn luyện mô hình:

- sử dụng mô hình SVM ở trên để dự đoán loại bệnh trên tập dữ liệu X_test

```

| from sklearn.metrics import accuracy_score, recall_score, f1_score
| from sklearn.metrics import roc_auc_score
| y_pred = svm.predict_proba(X_test)

```

5) Xuất kết quả:

- Dựa vào tập y_pred là kết quả dự đoán loại bệnh để xuất ra file submission.csv.
- File submission.csv:

submission.csv X					1 to 10 of 1821 entries		Filter	
image_id	healthy	multiple_diseases	rust	scab				
Test_0	0.29708165096473416	0.06071942478509327	0.2524138150588629	0.40088077500497477				
Test_1	0.1873915219178275	0.04649238675293331	0.4243867784424367	0.31769916809218346				
Test_2	0.17072022993507097	0.03144810612187956	0.26747697546808524	0.5636338527385046				
Test_3	0.0919907828012587	0.03401137531602634	0.39846476222136334	0.5511711624485809				
Test_4	0.1452501816303666	0.04577962416677515	0.2961438729079212	0.5718309055741166				
Test_5	0.21270017274924552	0.09165478806830324	0.22097414404045262	0.544134633854677				
Test_6	0.2752993118930075	0.053398801131754624	0.2904654922647393	0.4171982144318616				
Test_7	0.2660270787262728	0.06293123960824451	0.2472530305201207	0.5198120423201663				
Test_8	0.4219291389555612	0.09759651343952683	0.34563064087192646	0.11482955736461652				
Test_9	0.2911310731048829	0.06379973927141959	0.38697000764573114	0.12796787086050085				