

Báo cáo đồ án

Môn Máy học nâng cao – CS315.J11

Thành viên

Nhóm gồm 2 sinh viên:

1. Lê Ngọc Hải - MSSV: 15520182
2. Lê Văn Hạnh - MSSV: 15520197

Thu thập Dữ liệu

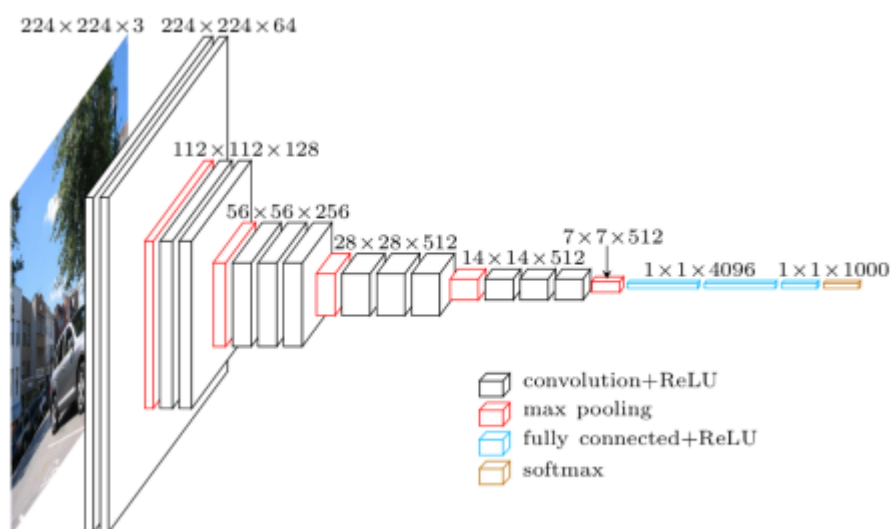
Nhóm thu thập được 30 tấm ảnh, chia đều cho hai lớp đối tượng như đã yêu cầu. Đối tượng mà nhóm chọn là: **motobike** và **non-motobike**.

Thông tin dữ liệu: Dữ liệu này gồm những ảnh được chụp thực tế ở trong khuôn viên Kí túc xá ĐHQG khu B.

Tiếp theo, là bước rút trích đặc trưng của dữ liệu.

Rút trích dữ liệu

Đặc trưng của ảnh được rút trích sử dụng mạng CNN VGG16 được đề cập ở paper *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Có 2 phiên bản phổ biến là VGG16 với 16 layer và VGG19 với 19 layer. Kiến trúc của mạng VGG16 tuy nhiều lớp nhưng dễ hình dung bao gồm 5 block convolutional với 13 lớp và 3 lớp fully-connected layer.



Chương trình sử dụng ma trận kết quả của lớp FC thứ nhất (nằm ở block thứ 6 nên được gọi là FC6) làm đặc trưng cho ảnh. Các tham số của mạng được lấy từ file của tác giả paper train trên tập ImageNet với 1000 lớp. Mạng có thể được train trên tập dữ liệu 30 ảnh tự thu thập nhưng số lượng ảnh quá ít nên không thích hợp cho một mô hình deep-learning.

Có nhiều phiên bản implement của các tác giả khác nhau. Qua thử nghiệm, phiên bản cài đặt mạng VGG16 của thư viện keras-application phù hợp với cấu hình máy tính cá nhân của nhóm nên được sử dụng ở đây. Ảnh đầu vào phải có kích thước 224x224x3 nếu sử dụng tham số được train từ tập ImageNet. Đầu ra của model là kết quả của lớp FC thứ nhất và sử dụng làm input cho mô hình SVM.

Xây dựng mô hình

Nhóm sử dụng mô hình Linear SVM của thư viện scikit-learn để phân loại hai lớp đối tượng trên tập train.

Khai báo thư viện:

```
from sklearn import svm
```

Khai báo mô hình:

```
clf = svm.LinearSVC()
```

Tham số huấn luyện là: `LinearSVC(penalty='l2', loss='squared_hinge', dual=True, tol=1e-4, C=1.0, multi_class='ovr', fit_intercept=True, intercept_scaling=1, class_weight=None, verbose=0, random_state=None, max_iter=1000)`

Học trên dữ liệu train:

```
clf.fit(X_train_fc6, y_train)
```

Lưu lại mô hình nhóm dùng thư viện pickle của Python.

Khai báo thư viện:

```
import pickle
```

Đặt tên file sẽ lưu là **model.sav**:

```
filename = 'model.sav'
```

Lưu file mô hình đã học xuống đĩa:

```
pickle.dump(clf, open(filename, 'wb'))
```

Test dữ liệu trên mô hình đã lưu

Load file mô hình đã lưu:

```
loaded_model = pickle.load(open(filename, 'rb'))
```

Test trên dữ liệu và ghi kết quả ra màn hình:

```
scores = loaded_model.score(X_test_fc6, y_test)

print("accuracy = %f" %scores)
```

Kết quả test trên dữ liệu là: **80%**

Hướng dẫn chạy chương trình:

Các thông tin về các lớp, đường dẫn lưu dataset và model được định nghĩa ở file `constants.py`.

Để train và save model, chạy file `train.py`.

```
$ python3 train.py
```

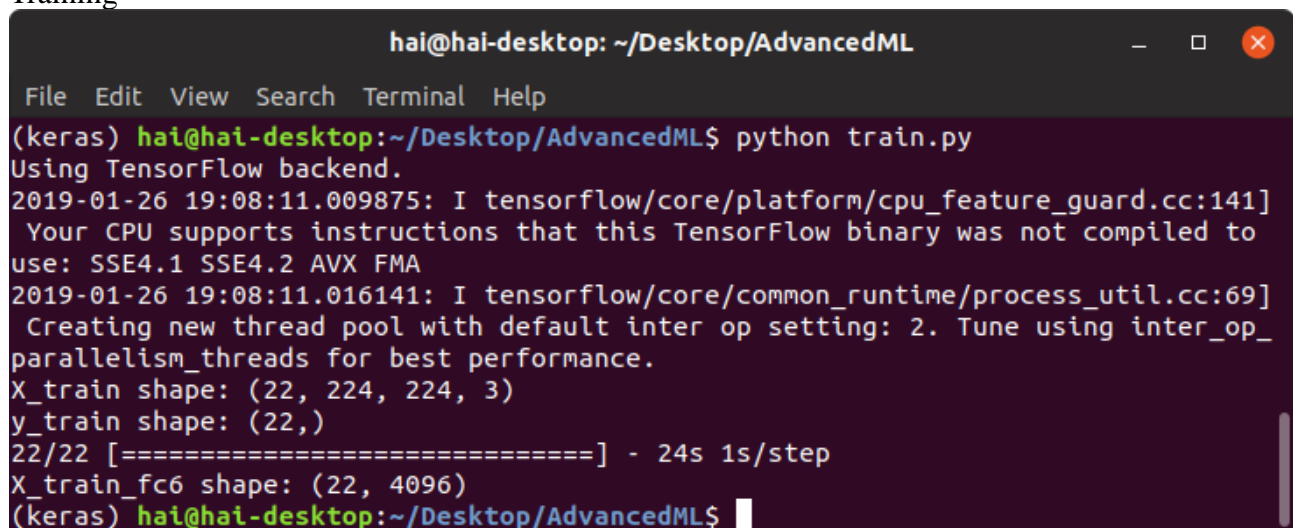
Để load và test model, chạy file `test.py` (chỉ thực hiện sau khi đã save model).

```
$ python3 test.py
```

Link github: <https://github.com/lvhanh270597/AdvancedML>

Demo chương trình

Training

A screenshot of a terminal window titled 'hai@hai-desktop: ~/Desktop/AdvancedML'. The terminal shows the execution of 'python train.py'. It displays TensorFlow backend information, CPU feature warnings, thread pool creation, and the shapes of training data (X_train: (22, 224, 224, 3), y_train: (22,)). It also shows a progress bar for 22/22 steps, taking 24 seconds per step, and the shape of the training features (X_train_fc6: (22, 4096)). The prompt '(keras) hai@hai-desktop:~/Desktop/AdvancedML\$' is visible at the bottom.

```
hai@hai-desktop: ~/Desktop/AdvancedML
File Edit View Search Terminal Help
(keras) hai@hai-desktop:~/Desktop/AdvancedML$ python train.py
Using TensorFlow backend.
2019-01-26 19:08:11.009875: I tensorflow/core/platform/cpu_feature_guard.cc:141]
  Your CPU supports instructions that this TensorFlow binary was not compiled to
  use: SSE4.1 SSE4.2 AVX FMA
2019-01-26 19:08:11.016141: I tensorflow/core/common_runtime/process_util.cc:69]
  Creating new thread pool with default inter op setting: 2. Tune using inter_op_
  parallelism_threads for best performance.
X_train shape: (22, 224, 224, 3)
y_train shape: (22,)
22/22 [=====] - 24s 1s/step
X_train_fc6 shape: (22, 4096)
(keras) hai@hai-desktop:~/Desktop/AdvancedML$
```

Testing

```
hai@hai-desktop: ~/Desktop/AdvancedML
File Edit View Search Terminal Help
(keras) hai@hai-desktop:~/Desktop/AdvancedML$ python test.py
Using TensorFlow backend.
2019-01-26 19:06:54.236890: I tensorflow/core/platform/cpu_feature_guard.cc:141]
  Your CPU supports instructions that this TensorFlow binary was not compiled to
use: SSE4.1 SSE4.2 AVX FMA
2019-01-26 19:06:54.243369: I tensorflow/core/common_runtime/process_util.cc:69]
  Creating new thread pool with default inter op setting: 2. Tune using inter_op_
parallelism_threads for best performance.
X_test shape: (15, 224, 224, 3)
y_test shape: (15,)
15/15 [=====] - 16s 1s/step
X_test_fc6 shape: (15, 4096)
Y test =
[0 0 0 0 0 0 0 0 1 1 1 1 1 1 1]
Y predict =
[0 0 0 0 0 0 0 0 1 1 1 0 0 0 1]
accuracy = 0.800000
false predictions:
['./images/test/non-bike/xe-dap-dien-honda-1512646699-1-5688341-1520132782.jpeg'
 './images/test/non-bike/xe-dap-dien-133-sport-46b8b8cf-ed55-4003-91e3-2f22213c7
649.jpg'
 './images/test/non-bike/z1118648032398_04b1fdbb695deec42b5b624abfb71927.md.jpg'
]
confidence scores:
[-0.56060437 -0.38431081 -0.27748296 -0.9368723 -0.82108762 -0.64797478
 -0.77540847 -1.14969156  0.60554453  0.55210539  1.22331105 -0.14420016
 -0.16466811 -0.48440333  0.41432244]
false predictions confidence:
[-0.14420016 -0.16466811 -0.48440333]
(keras) hai@hai-desktop:~/Desktop/AdvancedML$
```

Các ảnh dự đoán sai:

Ảnh	Nhãn đúng/Nhãn sai	Độ tin cậy
	Non-motobike/Motobike	-0.1442

	Non-motobike/Motobike	-0.1647
	Non-motobike/Motobike	-0.4844

Độ tin cậy được sử dụng trong model là khoảng cách từ sample đến siêu mặt phẳng học được. Do chỉ có 2 class nên nếu dấu của giá trị thể hiện phía của mặt phẳng mà sample thuộc về. So sánh với độ tin cậy của các trường hợp dự đoán đúng, ta thấy chúng không có nhiều khác biệt.

Dựa vào cảm tính, ta có thể thấy các ảnh dự đoán sai là không đáng ngạc nhiên vì mặc dù được phân vào lớp non-motobike (không phải xe máy) nhưng chúng thực tế có những đặc điểm giống với motobike (xe máy) chẳng hạn như có 2 bánh, có tay lái, có yên xe...

Một số nguyên nhân có thể dẫn đến các dự đoán sai:

- Dữ liệu chưa đủ: ta chưa cho model học cách phân biệt xe máy và xe đạp điện nên có thể dẫn đến việc hiểu xe máy là các xe 2 bánh có hình dáng tương đối giống xe máy. Ta cần cung cấp dữ liệu phân biệt 2 loại xe này nhiều hơn nếu muốn độ chính xác cao hơn.
- Model chưa tốt: việc sử dụng Linear SVM là cách tiếp cận đơn giản nhất của SVM nhưng thực tế dữ liệu có thể phân bố phức tạp hơn, khi đó ta cần thử nghiệm các kernel phức tạp hơn.
- Xử lý dữ liệu chưa tốt: đặc trưng được rút trích từ mạng VGG16 với 1000 lớp và được chứng minh là hiệu quả. Tuy nhiên không phải lúc nào dữ liệu được train từ tập ImageNet cũng phù hợp với bài toán như việc phân loại 2 loại xe gần giống nhau. Nếu đây là vấn đề thì ta cần tìm cách rút trích đặc trưng khác hiệu quả hơn.

Kết luận

Nhóm đã cài đặt được cho chương trình rút trích đặc trưng của ảnh từ mạng VGG16. Thử nghiệm phân lớp bằng mô hình Linear SVM với tập dữ liệu tự thu thập. Nhóm đã đánh giá mô hình qua một số tình huống khó phân biệt. Tuy nhiên mô hình Linear SVM và tập dữ liệu còn đơn giản cần cải thiện trong tương lai.

Tài liệu tham khảo

[1] VGG in TensorFlow, Davi Frossard - <https://www.cs.toronto.edu/~frossard/post/vgg16/>

[2] Support Vector Machines - <https://scikit-learn.org/stable/modules/svm.html>