BÁO CÁO ĐỒ ÁN CUỐI KỲ

MÔN: MÁY HỌC

LỚP: CS114.K21

GIẢNG VIÊN:

PGS.TS. LÊ ĐÌNH DUY
THS. PHẠM NGUYỄN TRƯỜNG AN

TỔNG QUAN

1. Mô tả ngắn bài toán

- Bài toán đặt ra là một bài toán phân lớp (Classification).
- Đầu vào (Input): Hình ảnh một con rắn.
- Đầu ra (Output): Cho biết rắn có độc hay không.

2. Ứng dụng

- Xây dựng ứng dụng giáo dục, học tập về các loài rắn.
- Xây dựng từ điển liên kết về các loài rắn.

3. Phương pháp sử dụng: Transfer Learning

Bài toán phân loại ảnh thường được chia làm hai bước: Feature Engineering và Train a Classifier. Những năm gần đây, Deep Learning phát triển cực nhanh dựa trên lượng dữ liệu training khổng lồ và khả năng tính toán ngày càng được cải tiến của các máy tính. Các kết quả cho bài toán phân loại ảnh ngày càng được nâng cao. Bộ cơ sở dữ liệu thường được dùng nhất là ImageNet với 1.2M ảnh cho 1000 classes khác nhau. Rất nhiều các mô hình Deep Learning đã giành chiến thắng trong các cuộc thi ILSVRC (ImageNet Large Scale Visual Recognition Challenge). Có thể kể ra một vài: AlexNet, ZFNet, GoogLeNet, ResNet, VGG.

Nhìn chung, các mô hình này đều bao gồm rất nhiều layers. Các layers phía trước thường là các Convolutional layers kết hợp với các nonlinear activation functions và pooling layers (và được gọi chung là ConvNet). Layer cuối cùng là một Fully Connected Layer và thường là một Softmax Regression (Xem Hình 1). Số lượng units ở layer cuối cùng bằng với số lượng classes (với ImageNet là 1000). Vì vậy output ở layer gần cuối cùng (second to last layer) có thể được coi là feature vectors và Softmax Regression chính là Classifier được sử dụng.

Chính nhờ việc features và classifier được trained cùng nhau qua deep networks khiến cho các mô hình này đạt kết quả tốt. Tuy nhiên, những mô hình này đều là các Deep Networks với rất nhiều layers. Việc training dựa trên 1.2M bức ảnh của ImageNet cũng tốn rất nhiều thời gian (2-3 tuần).

Với các bài toàn dựa trên tập dữ liệu khác, rất ít khi người ta xây dựng và train lại toàn bộ Network từ đầu, bởi vì có rất ít các cơ sở dữ liệu có kích thước lớn. Thay vào đó, phương pháp thường được

dùng là sử dụng các mô hình (nêu phía trên) đã được trained từ trước, và sử dụng một vài kỹ thuật khác để giải quyết bài toán. Phương pháp sử dụng các mô hình có sẵn như thế này được gọi là ***Transfer Learning*.**

I. BỘ DỮ LIỆU

1. Mô tả dữ liệu

Dữ liệu được lấy từ hai nguồn:

Từ Kaggle:

Nguồn dữ liệu được xây dựng lại từ dữ liệu được cung cấp trong cuộc thi AIcrowd Snake Species Identification Challenge, là một bản rút gọn tốt hơn về chất lượng trên năm lớp là các loài rắn. Trong đó có 4 loài từ lớp 1 đến lớp 4 là không độc và lớp 5 là độc:

- Northern Watersnake

- Common Garter snake

- DeKay's Brown snake

- Black Rat snake

- Western Diamondback rattlesnake

Từ Github:

Nguồn dữ liệu được xây dựng từ các hình ảnh tải về từ Google, bao gồm các loài rắn phổ biến ở Ấn Độ được chia ra làm hai lớp độc và không độc:

- Độc:
  + Common Krait
  + King Cobra
  + Monocled Cobra
  + Russell’s Viper
  + Saw-scaled Viper
  + Spectacled Cobra
- Không độc:
  + Banded Racer
  + Checkered Keelback
  + Common Rat Snake
  + Common Sand Boa
  + Common Trinket
  + Green Tree Vine
  + Rock Python

Vì các loài rắn phổ biến tại Ấn Độ thường có xu hướng trải dài phạm vi sinh sống từ Ấn Độ đến khu vực Đông Nam Á cho nên bộ dữ liệu này không mang tính chính xác và bao phủ hoàn toàn.

Số lượng ảnh:

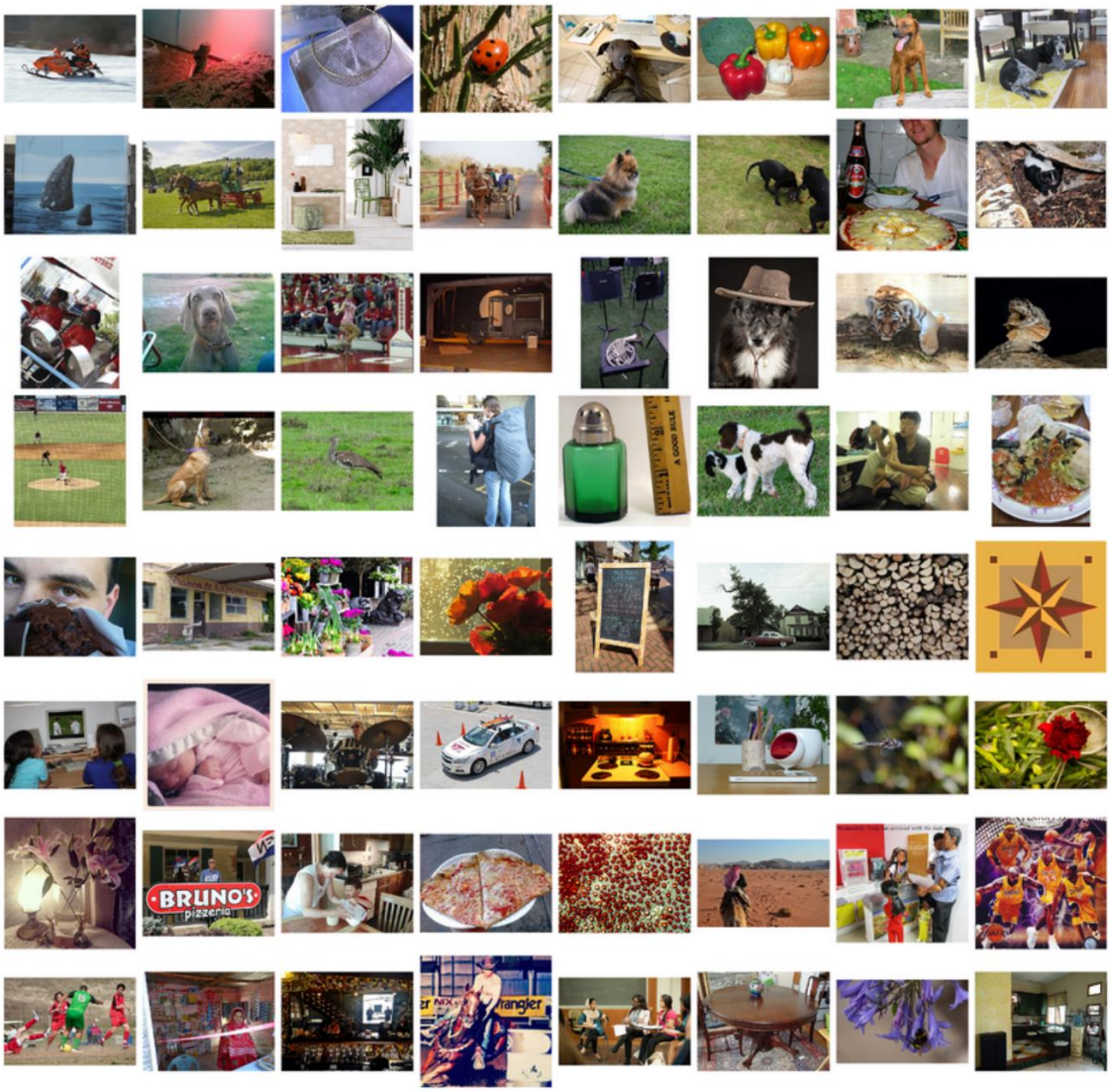
- Số lượng ảnh rắn độc là 4808.
- Số lượng ảnh rắn không độc là 4082.

Kích thước ảnh:

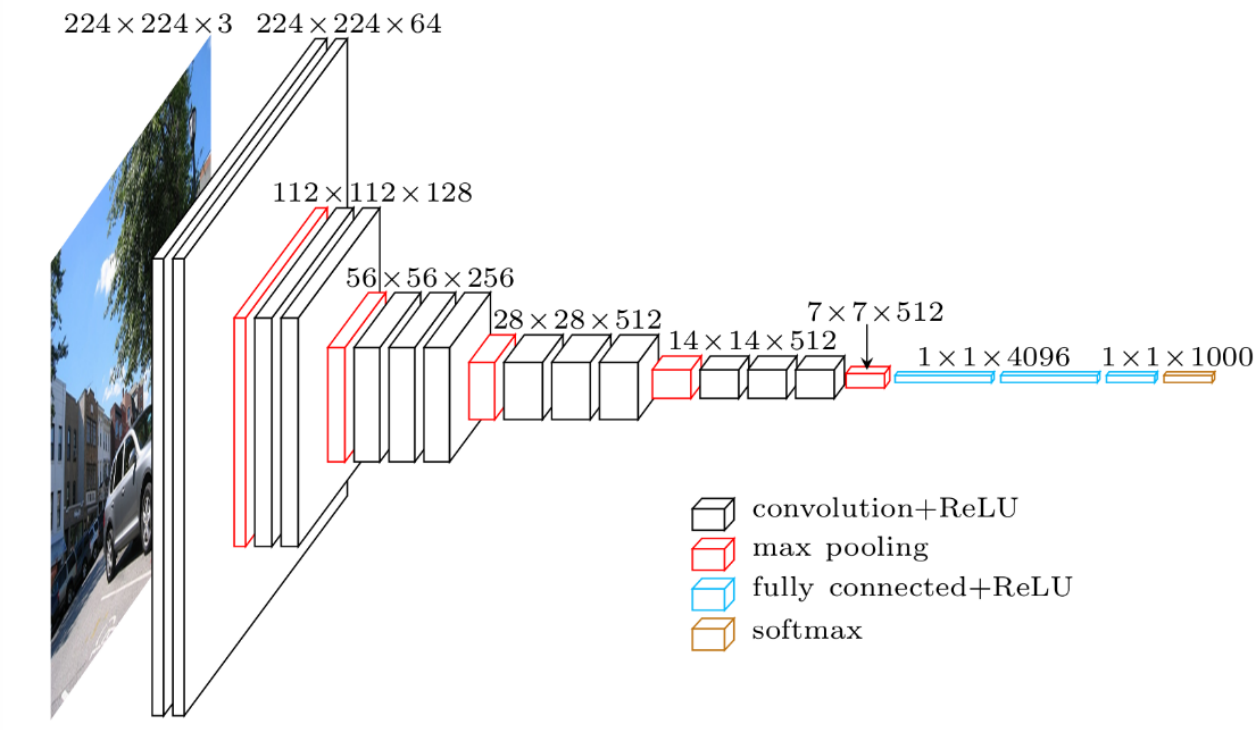
- Không đồng nhất ở bộ dữ liệu lấy từ Github.
- Trong bộ dữ liệu từ Kaggle là 384x384.

2. Xử lý dữ liệu

ImageNet là một cơ sở dữ liệu hình ảnh quy mô lớn được thiết kế để sử dụng trong nghiên cứu phần mềm nhận dạng đối tượng trực quan. Trên 14 triệu URL của hình ảnh đã được gán nhãn bằng tay bởi ImageNet để chỉ ra đối tượng nào có trong những bức hình. ImageNet chứa hơn 20 nghìn danh mục, một danh mục điển hình, chẳng hạn như "quả bóng" hoặc "dâu tây", chứa hàng trăm hình ảnh. Cơ sở dữ liệu nhãn hình ảnh của URL từ bên thứ ba cũng có sẵn miễn phí từ ImageNet; tuy nhiên, các hình ảnh thực tế không thuộc sở hữu của ImageNet. Từ năm 2010, dự án ImageNet thực hiện một cuộc thi phần mềm hàng năm, Thử thách nhận diện trực quan quy mô lớn của ImageNet (ImageNet Large Scale Visual Recognition Challenge – ILSVRC), nơi các phần mềm cạnh tranh để phân loại và phát hiện các đối tượng và cảnh vật một cách chính xác. ILSVRC sử dụng danh sách đã được "cắt xén" của một nghìn phân lớp không chồng chéo lên nhau.



VGG-16 là một mô hình CNN được đề xuất bởi K.Simonyan và A.Zisserman từ Đại học Oxford trên bài báo “Very Deep Convolutional Networks for Large-Scale Image Recognition”. Mô hình đạt 92.7% top 5 độ chính xác trên ImageNet (một bộ dữ liệu của hơn 14 triệu ảnh thuộc 1000 lớp).



Hình 2.. Mô hình của mạng VGG-16

Layer input lấy hình ảnh có kích thước (224x224x3), layer output cho ra dự đoán của layer softmax trên 1000 lớp. Từ layer input tới layer max pooling cuối cùng được xem là quá trình trích xuất đặc trưng, phần còn lại của mạng được xem là quá trình phân lớp của mô hình.

Sau khi xác định được mô hình cần tải ảnh đầu vào có kích thước mà mô hình mong muốn

Các thư viện sử dụng:

Thư viện	Vai trò
numpy	Xử lý ma trận và danh sách các ảnh
sklearn	Lấy các mô hình để phân lớp: SVM, Logistic Regression,… Hàm chia dữ liệu thành hai phần train và test
keras	Lấy mô hình VGG-16, các hàm xử lý trên ảnh để tạo input phù hợp cho

Đầu tiên tải mô hình VGG-16 từ thư viện keras với weights = 'imagenet' (pretrained weights cho bộ dữ liệu Imagenet) và include_top = False (không bao gồm 3 layer fully-connected ở đầu mạng).

```
model = VGG16(weights='imagenet', include_top=False)
```

Xử lý các ảnh và tạo danh sách ảnh:

- Lấy đường dẫn của ảnh với hàm load_image() và thay đổi kích thước ảnh thành kích thước yêu cầu (224, 224).
- Chuyển ảnh sang mảng numpy các điểm ảnh bằng hàm img_to_array().
- Tăng chiều của mảng từ 3 chiều lên 4 chiều bằng hàm expand_dims() với các chiều [samples, rows, columns, channels].
- Các ảnh cần được chuẩn bị như dữ liệu training của Imagenet, keras cung cấp cho ta hàm preprocesss_input() để chuẩn bị input cho mạng.
- Thêm ảnh đã qua xử lý vào danh sách các ảnh.

```
list_image = []
for (j, imagePath) in enumerate(image_path):
    image = load_img(imagePath, target_size=(224, 224))
    image = img_to_array(image)

    image = np.expand_dims(image, 0)
    image = imagenet_utils.preprocess_input(image)

    list_image.append(image)

list_image = np.vstack(list_image)
```

Sau khi xử lý các ảnh đầu vào thành một danh sách phù hợp với mạng VGG-16, sử dụng hàm predict() để lấy các đặc trưng của ảnh.

```
#featuring
features = model.predict(list_image)
```

Chia bộ dữ liệu làm hai tập train và test với tỷ lệ 80% (train) và 20% (test) với hàm train_test_split(). Trong đó, features là các đặc trưng có được từ bước trước và labels là nhãn của các ảnh tương ứng.

```
X_train, X_test, y_train, y_test = train_test_split(features, labels, test
_size=0.2, random_state=42)
```

II. LỰA CHỌN BỘ PHÂN LỚP

Vì mạng VGG-16 là một mô hình đã được huấn luyện tốt và được công nhận nên những gì lấy được từ quá trình trích xuất đặc trưng trên là đáng tin cậy. Các ouput từ quá trình trên được sử dụng làm input cho các mô hình phân lớp quen thuộc. Trong đề tài này, hai classifier đã lựa chọn là Linear SVM, Logistic Regression và Random Forest Classifier.

Huấn luyện các Classifier đã chọn:

SVM (Support Vector Machine)

```
#train with svm
svm_linear = SVC(kernel='linear')
svm_linear.fit(X_train, y_train)
```

Logistic Regression

```
#train with Logistic Regression
reg = LogisticRegression()
reg.fit(X_train,y_train)
```

Random Forest Classifier

```
rfc = RandomForestClassifier()
rfc = rfc.fit(X_train, y_train)
```

Lưu lại các mô hình đã huấn luyện:

```
#save model
joblib.dump(svm_linear, '/content/drive/My Drive/hk2 2019-2020/machine learni
ng/venomous snake classification/model/svm_linear1.pkl')
joblib.dump(reg, '/content/drive/My Drive/hk2 2019-2020/machine learning/veno
mous snake classification/model/logistic_regression1.pkl')
```

III. THỰC NGHIỆM

Trước tiên cần load các mô hình đã lưu để sử dụng:

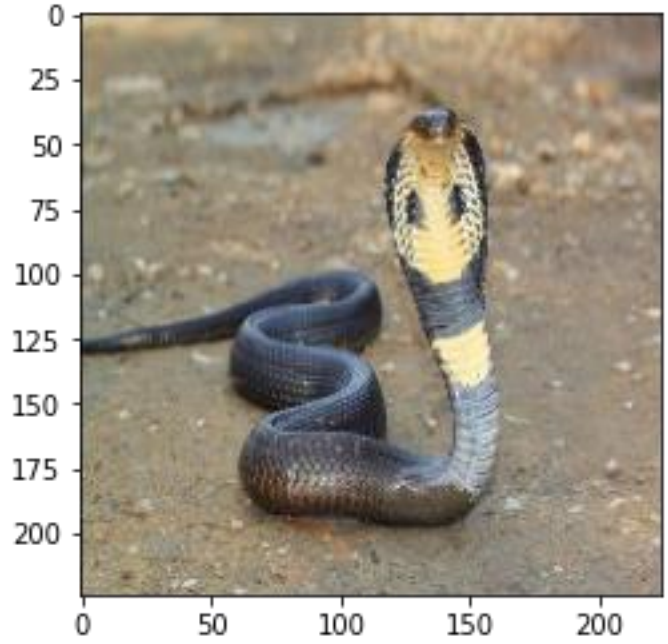
```
#load model
svm_linear=joblib.load('/content/drive/My Drive/hk2 2019-2020/machine lear
ning/venomous snake classification/model/svm_linear1.pkl')
logistic_regression =joblib.load('/content/drive/My Drive/hk2 2019-2020/ma
chine learning/venomous snake classification/model/logistic_regression1.pk
l')
```

Kết quả đạt được:

Mô hình	Độ chính xác
SVM	0.81
Logistic Regression	0.83
Random Forest Classifier	0.80

Một số ảnh thử và kết quả:

Image 1 :
Name of image: ho_mang_chua_doc.jpg



Support vector machine result: Có độc
Logistic Regression: Có độc
Random Forest Classifier: Có độc

Image 2 :
Name of image: ran_bongsung_khongdoc.jpg

Support vector machine result: Không có độc

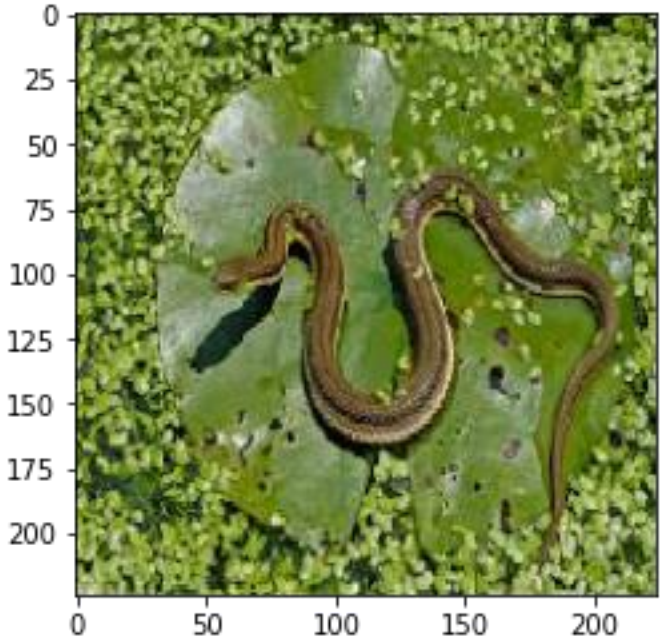
Logistic Regression: Không có độc
Random Forest Classifier: Không có độc

Image 3 :
Name of image: ran_cap_nia_doc.jpg

Support vector machine result: Không có độc
Logistic Regression: Không có độc
Random Forest Classifier: Có độc
Image 5 :
Name of image: ran_cap_nong_doc.jpg

Support vector machine result: Có độc
Logistic Regression: Có độc
Random Forest Classifier: Có độc

Image 11 :
Name of image: ran_nuoc_khongdoc_1.jpg



Support vector machine result: Không có độc
Logistic Regression: Không có độc
Random Forest Classifier: Không có độc

Đánh giá:

- Mô hình đạt độ chính xác không quá cao trên kết quả huấn luyện.
- Mô hình thể hiện không tốt trên các ảnh ngoài bộ dữ liệu.
- Đối với các loài rắn có những đặc điểm giống nhau hoặc giống với môi trường mô hình có tỷ lệ cao dự đoán sai.

Những khó khăn gặp phải:

- Về bộ dữ liệu, việc xây dựng bộ dữ liệu từ thực tế là không khả thi, các hình ảnh lấy từ Google hình ảnh có kích thước và tính chất không đồng nhất, nhiều hình ảnh có thể bị trùng hoặc mờ, các bộ dữ liệu được xây dựng sẵn có xu hướng gò bó trong một số loài nhất định.
- Về đặc trưng, các loài rắn vốn đã khó phân biệt với con người bởi các đặc trưng về vẻ ngoài có thể rất giống nhau nên khi phân lớp dễ gây nhầm lẫn, không chính xác, một số loài rắn có màu sắc và khả năng ngụy trang rất giống với môi trường xung quanh nên rất khó để nhận biết.
- Về thực tiễn, ứng dụng không quá hữu ích khi gặp sự cố bắt gặp rắn hay bị rắn cắn, những tình huống này thường sẽ nhanh chóng đến trung tâm y tế, các trường hợp biến chuyển xấu hơn thường đến từ những người vốn không quen thuộc với công nghệ

KẾT LUẬN

Sau khi xây dựng và thực nghiệm, các mô hình không thể hiện được quá tốt, tồn tại nhiều khó khăn và hạn chế trong khâu chuẩn bị bộ dữ liệu.

Mô hình thể hiện tốt với các loài rắn có đặc điểm bề ngoài đặc trưng và phân biệt với những loài khác nhưng lại phản ứng không tốt với các loài có đặc điểm bề ngoài có nhiều tương đồng và khác nhau nhỏ hay các loài có khả năng ngụy trang, hòa mình với môi trường xung quanh.

Đề tài này là một bài toán ngắn và đơn giản, tính chất thực tiễn không quá cao nhưng có thể phát triển cao hơn thành những ứng dụng có thể áp dụng vào thực tiễn.

TÀI LIỆU THAM KHẢO

https://github.com/Nguyennd507/MachineLearning/blob/master/FlowerSVM.ipynb

https://medium.com/@franky07724_57962/using-keras-pre-trained-models-for-feature-extraction-in-image-clustering-a142c6cdf5b1

https://towardsdatascience.com/extract-features-visualize-filters-and-feature-maps-in-vgg16-and-vgg19-cnn-models-d2da6333edd0#:~:text=Extract%20Features%20with%20VGG16&text=Then%20the%20VGG16%20model%20is,final%20dense%20layers%20or%20not.

https://machinelearningcoban.com/2017/07/02/tl/

https://en.wikipedia.org/wiki/ImageNet