**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN**

CS315.J11

**KẾT HỢP MÔ HÌNH SVM VÀ CNN ĐỂ GIẢI QUYẾT BÀI TOÁN PHÂN LỚP ẢNH**

**GVHD: NGUYỄN VINH TIỆP**

**SV THỰC HIỆN:**

**- Nguyễn Hoàng Huy 15520303**

**- Nguyễn Văn Kiệt 15520401**

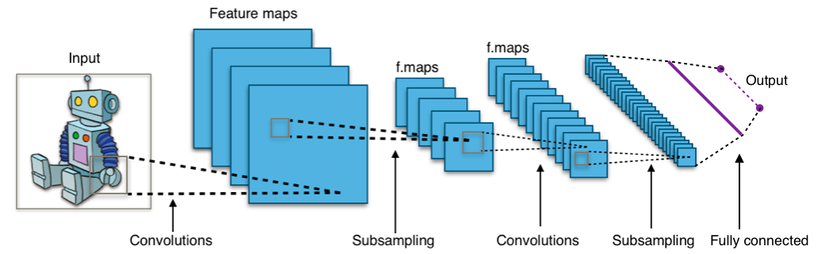
**TP. HỒ CHÍ MINH – 01/2019**

1. **Mô tả bài toán:**

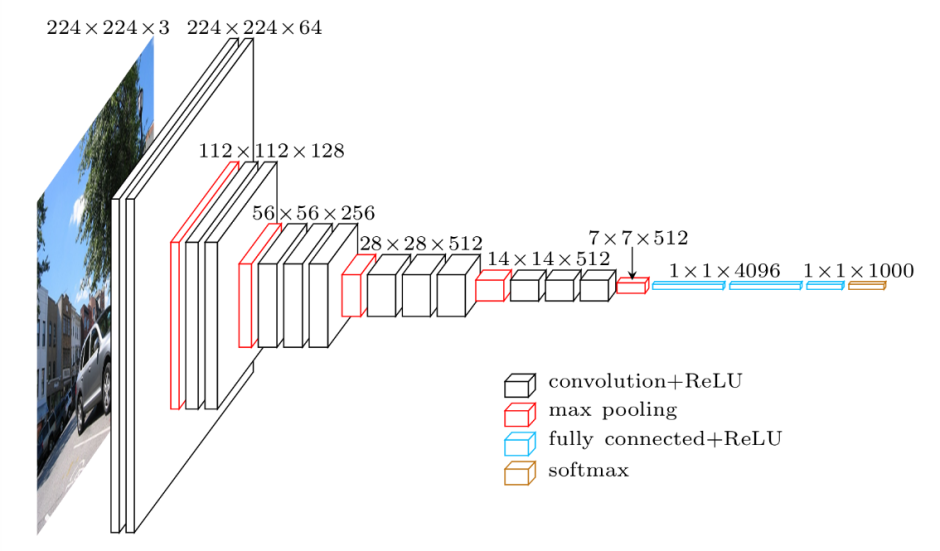
Cho tập dữ liệu ảnh gồm hai lớp đối tượng (ví dụ: motobike và non-motobike, human và non-human). Sử dụng mạng VGG16 để rút trích đặc trưng từ dữ liệu ảnh, sau đó kết hợp mô hình SVM classifier của thư viện scikit\_learn để phân loại hai lớp đối tượng trên tập train. Dùng model đã huấn luyện để dự đoán kết quả phân lớp ảnh trên tập ảnh test và đánh giá độ chính xác.

1. **Cơ sở lý thuyết:**

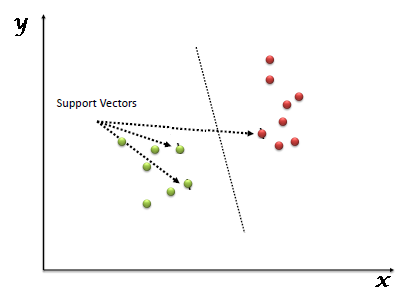
- Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay như hệ thống xử lý ảnh lớn như Facebook, Google hay Amazon đã đưa vào sản phẩm của mình những chức năng thông minh như nhận diện khuôn mặt người dùng, phát triển xe hơi tự lái hay drone giao hàng tự động. CNN được sử dụng nhiều trong các bài toán nhận dạng các object trong ảnh. CNN là một deep neural network artritecture. Hiểu đơn giản, nó cũng chính là một dạnh Artificial Neural Network, một Multiplayer Perceptron nhưng mang thêm 1 vài cải tiến, đó là Convolution và Pooling.



Mạng VGG16: Được phát triển năm 2014, là một biến thể sâu hơn nhưng lại đơn giản hơn so với kiến trúc convolution (từ gốc: convolutional structure) thường thấy ở CNN Kiến trúc như hình dưới, có thể thấy số mặc dù các tầng cao hơn được đơn giản hóa so với LeNet, AlexNet … thu gọn về kích thước nhưng số lượng lại lớn hơn, sâu hơn.

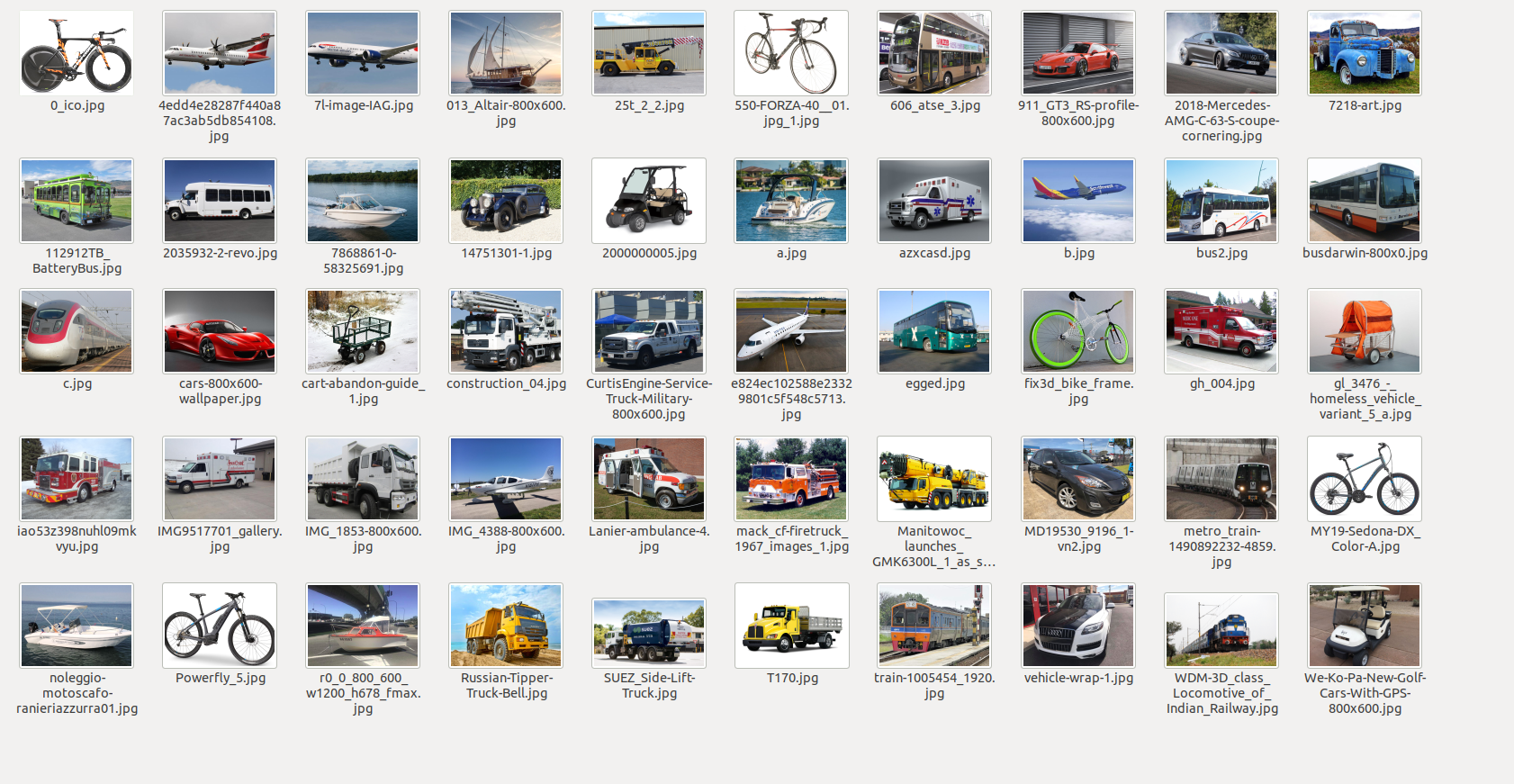
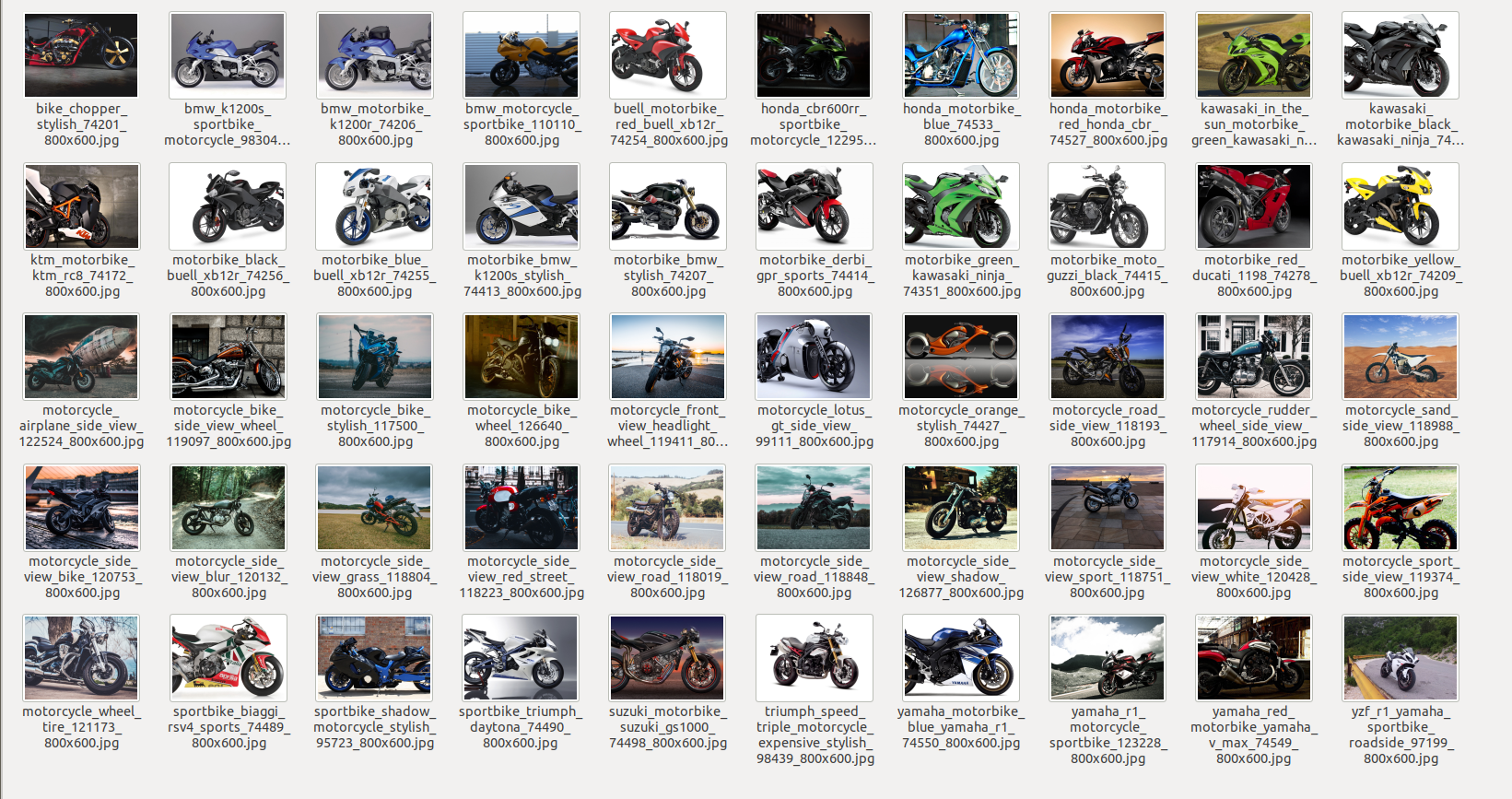


- Input của mạng VGG16 là 1 ảnh có kích thước 244x244x3 (pixels), output là 1 layer với 1000 node, tương ứng với 1000 class trong Imagenet, sử dụng Softmax với phân phối xác suất đầu ra ứng với từng class. Các lớp hidden layers ở giữa bao gồm các layer cơ bản của 1 mạng CNN như: lớp Convolution, Pooling (Max Pooling, Avegare Pooling, ..), Fully Connected Layer và Activation Function (Sigmoid, Tanh, Relu, ..). Như mình đã nói ở trên, mạng neural hoạt động cũng tương tự như 1 feature extraction, với các layer ở lớp đầu dùng để bóc tách các low level feature như: góc, cạnh, hình khối vật thể. Các layer ở lớp cao hơn thực hiện bóc tách cách high level feature, ứng với các đặc trưng riêng của từng đối tượng.



- SVM (Support Vector Machine): là một thuật toán giám sát, nó có thể sử dụng cho cả việc phân loại hoặc đệ quy. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, chúng ta vẽ đồ thị dữ liệu là các điểm trong n chiều (ở đây n là số lượng các feature bạn có) với giá trị của mỗi feature sẽ là một phần liên kết. Sau đó chúng ta thực hiện tìm đường phân chia các lớp - là 1 đường thằng có thể phân chia các lớp ra thành hai phần riêng biệt. Support Vectors hiểu một cách đơn giản là các đối tượng trên đồ thị tọa độ quan sát, Support Vector Machine là một biên giới để chia hai lớp tốt nhất.

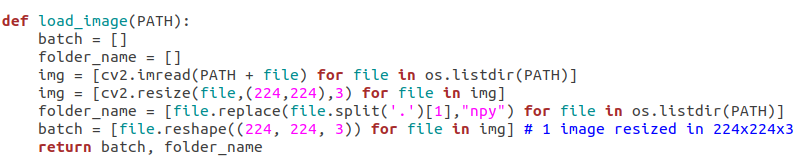
1. **Cài đặt chương trình:**
2. **Chuẩn bị dữ liệu:**



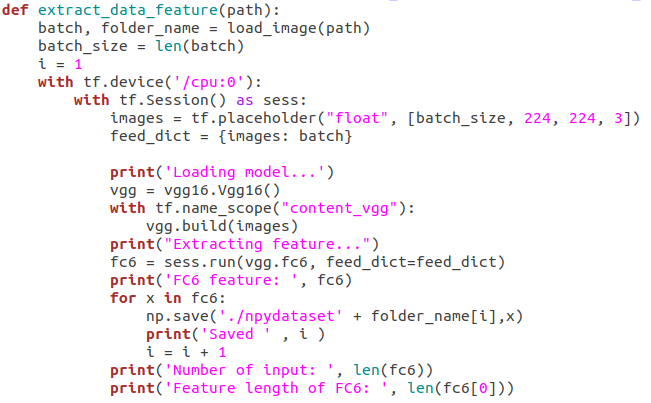
- Nhóm em làm về phân lớp ảnh giữa Motorbike và Non-motorbike, tải về ảnh tìm từ google với kích thước 800x600 càng tốt. Để thuận tiền cho việc gán nhãn nhóm em đặt tên cho ảnh lần lượt là motor-? và nonmotor-? (? là số ).

**2. Load ảnh từ thư mục:**

- Dùng thư viện cv2.imread() để load ảnh, vì định dạng của VGG16 là (224x224x3) nên phải reshape ảnh theo đúng kích thước. Đồng thời đổi đuôi file .jpg thành .npy để phục vụ cho việc rút trích đặc trưng VGG16 theo FC6.

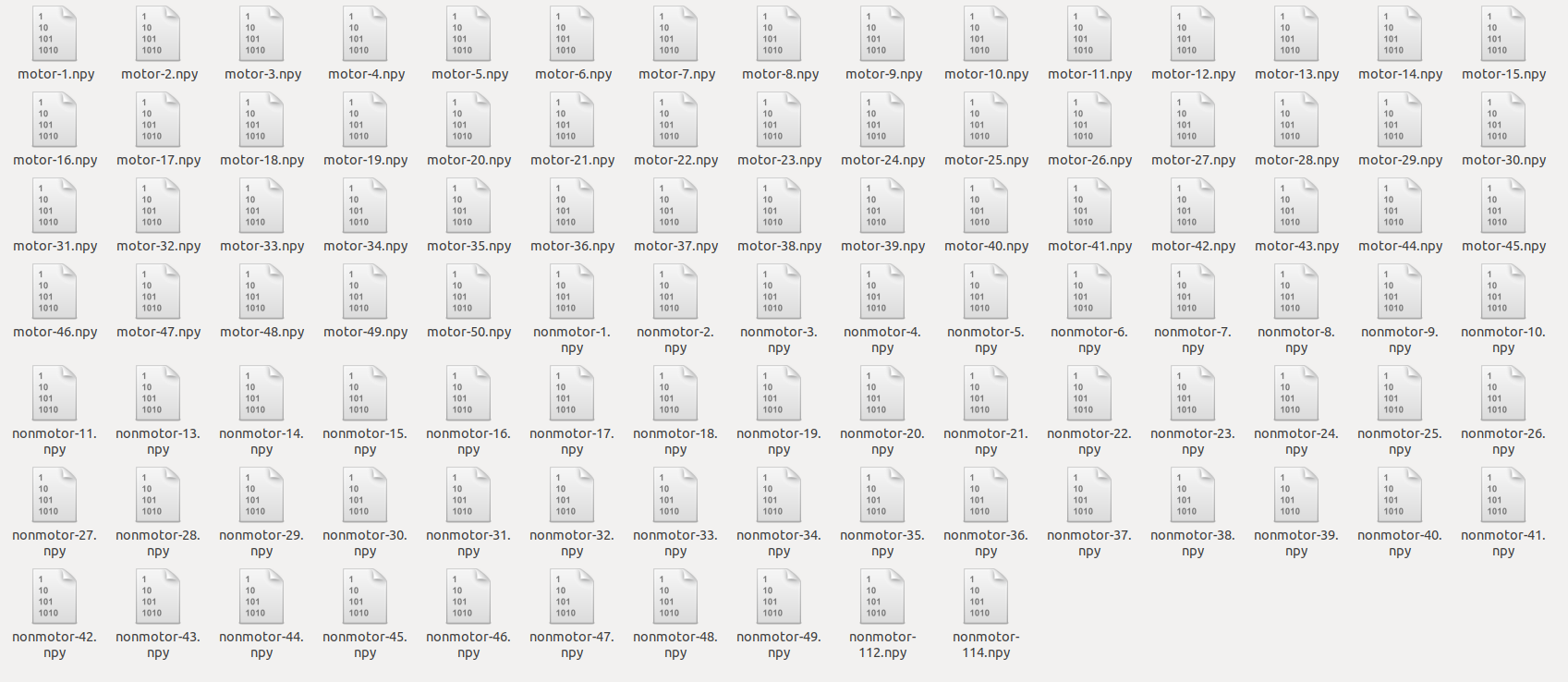
**3. Rút trích đặc trưng:**

- Rút trích đặc trưng 100 tấm ảnh, với output là ma trận có 100 vector tương ứng với 100 ảnh, với mỗi vector có kích thước là 4096.



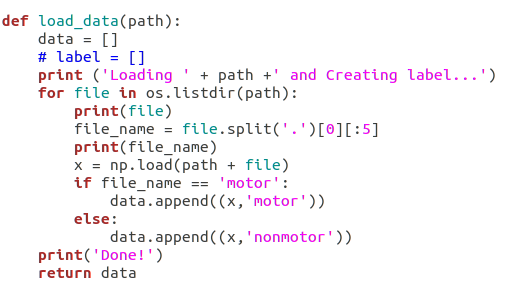
Mạng vgg16 được sử dụng trong chương trình, bao gồm 5 lớp convolutional và xem kẽ là 5 lớp pooling dùng max\_pool filter, đến 3 lớp fully connected fc6, fc7 và fc8 (ở đây chương trình sẽ trích xuất đặc trưng fc6 từ mạng vgg16) với hàm activation được sử dụng là relu, cuối cùng là hàm softmax

Sau khi rút trích đặc trưng xong, chương trình sẽ lưu các vector đặc trưng này xuống file, với tên file được đặt theo tên của ảnh đã được lấy trước đó.

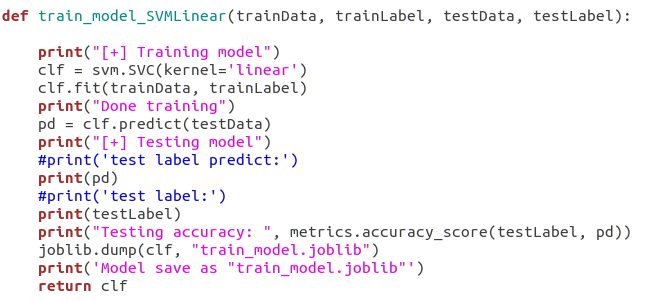
- Kết quả là folder npydataset sẽ lưu 100 vector đặc trưng của 100 ảnh dưới dạng .npy

**4. Kết hợp mô hình SVM để train dữ liệu và test:**

**- Thực hiện load file .npy, bằng cách split 5 kí tự đầu để lấy label cho ảnh, và lưu vector đặc trưng ứng với label của ảnh đó vào mảng.**



- Từ mảng data ta có phần tử data[0] sẽ là vector đặc trưng, data[1] là label, dùng hàm có sẵn shuffle để chia data ngẫu nhiên theo tỷ lệ 80 % để train, 20% để test. Ta gọi hàm fit(trainData, trainLabel) để học từ tập dữ liệu train. Sau đó theo dữ liệu đã học được lưu trong clf ta gọi hàm predict để dự đoán vector đặc trưng của testData. Tiếp theo là sẽ in ra phần trăm dự đoán của mô hình và dữ testLabel thực tế.



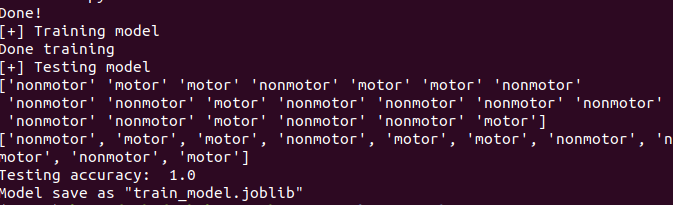
Thực hiện training data và tiến hành test. Model huấn luyện sẽ được lưu lại với tên "train\_model.joblib" bằng joblib để sau này có thể load lên lại và tiến hành test nhanh

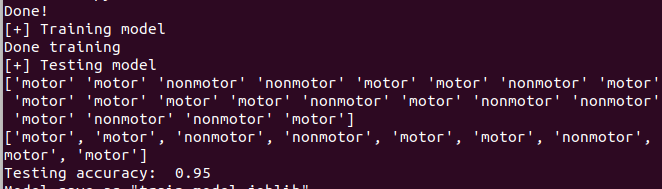
1. **Kết quả:**

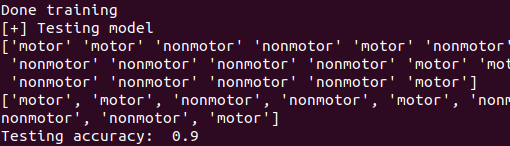
Kết quả sau khi chạy rút trích đặc trưng fc6 bằng cách chạy file extract\_feat.py

- Kết quả của mô hình sẽ được lưu lại với đuôi .joblib (train\_model.joblib)

- Một vài lần chạy model:







**5. Hướng dẫn chạy model:**

- “ python3 train.py “

1. **Nhận xét về đồ án:**

- Kết quả test của model có độ chính xác trung bình từ 0,95 ( tốt ), tuy nhiên để độ chính xác cao hơn cần thêm dữ liệu train nhiều và đa dạng.

- Với việc gọi hàm chọn ngẫu nhiên tập data (hàm shuffle) sẽ xảy ra trường hợp lớp này nhiều hơn lớp kia và ngược lại. Nên đôi khi ảnh hưởng đến độ chính xác.

- Code có nhiều đường dẫn gán cứng nên, khi muốn test hoặc train model mới phải sửa lại một số chỗ.