Xác định tình trạng bánh xe

Huỳnh Hữu Hào  
19520521@gm.uit.edu.vnHuỳnh Nhật Nam   
19520750@gm.uit.edu.vn  
Phan Tấn Bình  
19521269@gm.uit.edu.vn

# Giới thiệu

## Giới thiệu

Trong quá trình tìm hiểu qua các công nghệ về ô tô, nhóm chúng tôi mong muốn có một phần mêm có khả năng phân tích tình trạng xe ô tô thông qua hình ảnh.

Hầu hết phần mềm hiện nay đều là phần mềm chuyên dụng cho các hang,

## Mục tiêu:

Xây dụng mô hình máy học có khả năng dự đoán tình trạng của bánh xe.

## Bài toán máy học:

* Dữ liệu đầu vào: Hình ảnh lốp xe bất kỳ.
* Dữ liệu đầu ra: tình trạng của lốp xe(được dán nhãn là full, flat, notire (tức căng, xẹp, hoặc không có).

# Dữ liệu

## Phương pháp tự xây dựng tập dữ liệu

### Tên bộ dữ liệu: tire-dataset.

### Cách thu thập: được thu thập thông qua các trang stock images như: adobe stock, shutterstock, istockphoto,…

Logo

Description automatically generated

Hình 1. Adobe Stock

### Cách thức thu thập: thủ công.

### Link bộ dữ liệu: shorturl.at/cwxE4

## Phân tích sơ bộ về tập dữ liệu

* Tập dữ liệu được chia thành 3 thư mục (train, test, val)và chứa các thư mục con cho mỗi nhãn hình ảnh (flat/full/notire).
* Có 1200 hình ảnh (JPEG) và 3 nhãn (flat/full/notire).
* Mô tả tập dữ liệu:

#### Mô tả về tập hình ảnh flat: là các hình ảnh lốp xe bị xẹp, không có hơi, phần lốp xe thể hiện rõ ràng với các bộ phận khác của xe.



Hình 2. Ví dụ về hình ảnh tập *flat*

#### Mô tả về tập hình ảnh full: là các hình ảnh lốp xe căng đầy, phần lốp xe thể hiện rõ ràng với các bộ phận khác của xe.

A picture containing car, outdoor, road, parked

Description automatically generated

Hình 3. Ví dụ về hình ảnh tập *full*

#### Mô tả về tập hình ảnh notire: là các hình ảnh về các khung bánh xe, đĩa xe, các hình ảnh về lốp xe bị biến dạng hoặc các hình ảnh không liên quan đến lốp xe.

A picture containing outdoor

Description automatically generated

Hình 4. Ví dụ về hình ảnh tập *notire*

# Phương pháp tiếp cận

## Tiền xử lý dữ liệu

Việc tiền xử lý dữ liệu là một bước vô cùng cần thiết, có ảnh hưởng tới các bước tiếp theo trong việc huấn luyện các mô hình học máy và học sâu đồng thời ảnh hưởng tới độ chính xác của mô hình.

Dữ liệu hình ảnh được thu thập từ rất nhiều nguồn. Chính vì vậy cần phải được điều chỉnh về một kích thước. Để phù hợp với mô hình sẽ sử dụng, cụ thể là VGG16, hình ảnh sẽ được đưa về dạng 224x224 bằng cách sử dụng công cụ của thư viện OpenCV cụ thể là hàm cv2.resize().

A picture containing text, transport, car

Description automatically generated

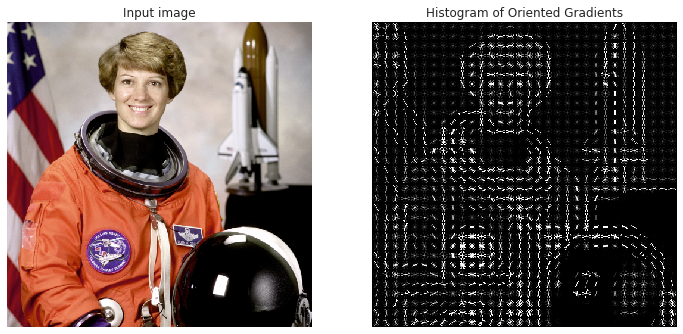
Hình 5. Hình ảnh trước khi tiền dữ liệu

A picture containing text, transport

Description automatically generated

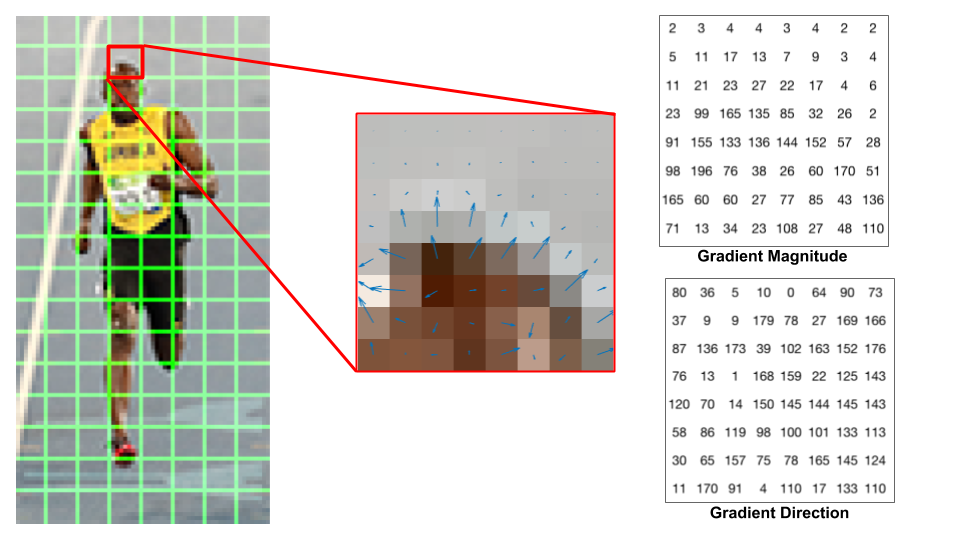
Hình 6. Hình ảnh sau khi tiền dữ liệu

Ngoài ra với các mô hình máy học truyền thống thì đầu vào phải là một array, chính vì vậy chúng tôi thực hiện trích xuất đặc trưng Histogram of oriented gradients của hình ảnh



Hình 7. HOG

Về HOG:



Hình 8: Tính toán HOG

Bản chất của phương pháp HOG là sử dụng thông tin về sự phân bố của các cường độ gradient (intensity gradient) hoặc của hướng biên (edge directins) để mô tả các đối tượng cục bộ trong ảnh. Các toán tử HOG được cài đặt bằng cách chia nhỏ một bức ảnh thành các vùng con, được gọi là “tế bào” (cells) và với mỗi cell, ta sẽ tính toán một histogram về các hướng của gradients cho các điểm nằm trong cell. Ghép các histogram lại với nhau ta sẽ có một biểu diễn cho bức ảnh ban đầu. [1]

## Tăng cường dữ liệu

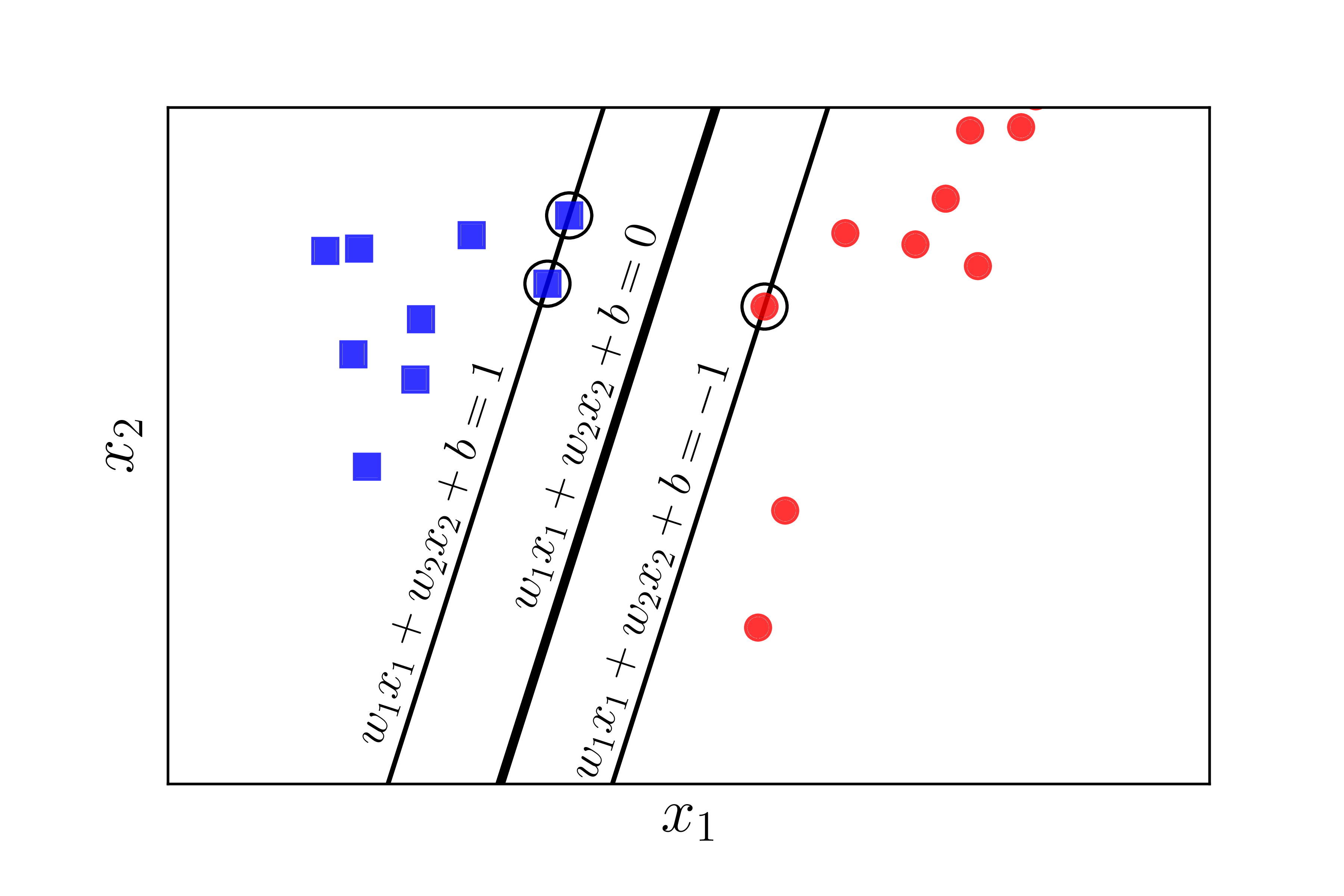
Với tập dữ liệu còn khá ít gây khó khan cho quá trình huần luyện mô hình. Nhóm chúng tôi sử dụng phương phát tăng cường dữ liệu (data augementation) lên tập dữ liệu giúp tạm thời tăng số lượng điểm dữ liệu nhằm tạo ra kết quả tốt nhất cho quá hình huấn luyện. Sử dụng hàm ImageDataGenerator của keras để tăng cường dữ liệu với các tham số:

* Rescale = 1./255 : đưa RGB coefficients từ 0-255 thành dạng (0,1)
* Rotation\_range = 5: random rotation 5 độ.
* Zoom\_range = 0.10: random zoom trong khoảng 0.9 đến 1.1.
* Width\_shift\_range = 0.05: dịch hình ảnh sang trái và phải trong khoảng 5% của độ rộng hình ảnh.
* Height\_shift\_range = 0.05: dịch hình ảnh lên xuống trong khoảng 5% của độ cao hình ảnh.

## Mô hình học máy truyền thống

### Linear Support Vector Classifier(lsvc)

*SVM* 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 đồi thị dữ liệu là các điểm trong n chiều ( ở đây n là số lượng các tính năng bạn có) với giá trị của mỗi tính năng sẽ là một phần liên kết. Sau đó chúng ta thực hiện tìm "đường bay" (hyper-plane) phân chia các lớp. Hyper-plane nó chỉ hiểu đơn giản là 1 đường thẳng có thể phân chia các lớp ra thành hai phần riêng biệt. [2]



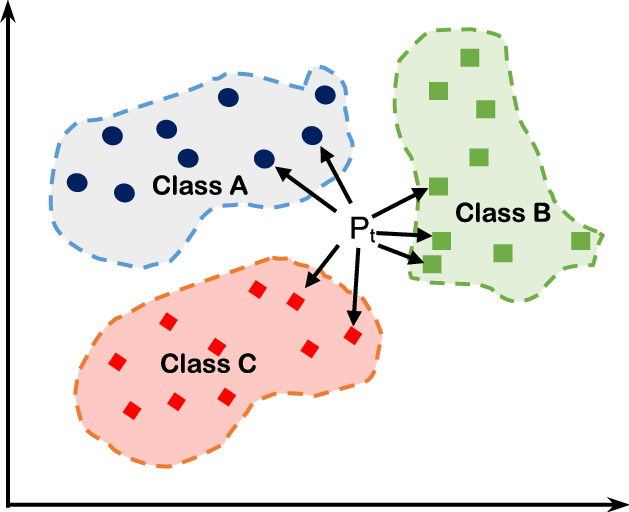
Hình 9: SVM

Linear Support Vector Classification là mô hình được gọi từ hàm LinearSVC của thư viện sklearn. Đây là mô hình phù hợp với các tập dữ liệu có dạng là tuyến tính, ở đây là array 1D chứa giá trị HOG. Cách hoạt động tương tự như cách họp động của SVM được thêm vào tham số kernel=’linear’.

### K-nearest Neighbors

*K-nearest neighbors (KNN)* là một kĩ thuật học có giám sát (supervised learning) dùng để phân loại dữ liệu mới bằng cách tìm điểm tương đồng giữa dữ liệu mới này với dữ liệu sẵn có. Khi training, thuật toán này không học một điều gì từ dữ liệu training (đây cũng là lý do thuật toán này được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới.

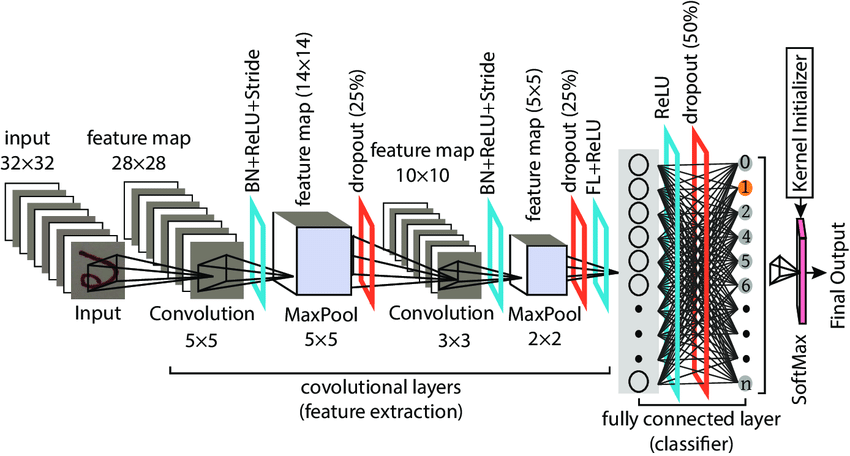
Với KNN, trong bài toán Classification, label của một điểm dữ liệu mới được suy ra trực tiếp từ K điểm dữ liệu gần nhất trong training set. Label của một test data có thể được quyết định bằng major voting (bầu chọn theo số phiếu) giữa các điểm gần nhất, hoặc nó có thể được suy ra bằng cách đánh trọng số khác nhau cho mỗi trong các điểm gần nhất đó. KNN có thể áp dụng được vào cả hai bài toán phân lớp và hồi quy. [3]



Hình 10: KNN

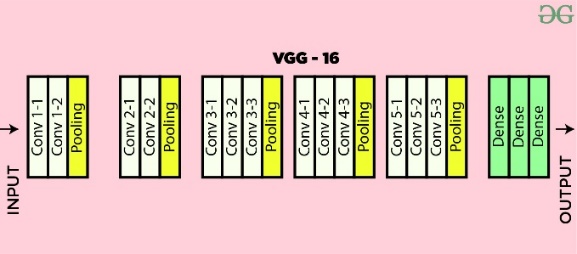
## Mô hình deep learning CNN

Convolutional Neural Network (CNN) là một trong những mô hình Deep Learning tiên tiến. Được sử dụng nhiều trong các bài toán nhận dạng các object trong ảnh. CNN là một kiến trúc mạng nơ-ron đa tầng được phát triển để phục vụ các nhiệm vụ phân loại. Nó có thể phát hiện các tính năng kết hợp và đưa ra kết quả được phân loại bằng các phép tính và hàm kích hoạt (Activation Function) ở các lớp chập.



Hình 11: Mô hình CNN

VGG16 là mô hình CNN được huấn luyện sẵn được tạo ra K. Simonyan và A. Zisserman tại trường đại học Oxford (Mỹ) đã được kiểm thử qua rất nhiều tập dữ liệu khác nhau. Model đã đạt được 92.7% top5-test accuracy tại ImageNet.



Hình 12: Mô hình VGG16

## Các độ đo sử dụng để đánh giá

### Độ đo chính

Độ đo chính được sử dụng là accuracy.

Sử dụng độ đo này bởi nhu cầu về phân lớp ảnh quan tâm nhiều hơn đến True Positive và True Negative, ngoài ra tập dữ liệu có đặc điểm là phân chia đều giữa các nhãn nên việc sử dụng độ đo này là hợp lí.

Accuracy =

### Các độ đo khác

#### Dành riêng cho mô hình máy học truyền thống

Cross\_val\_score: bằng cách áp dụng k-Fold Cross Validation để đánh giá khả năng của 2 mô hình máy học truyền thống

#### Các độ đo tương đồng khác

• Precision: là tỉ lệ thực sự positive trên tổng số các trường hợp được mô hình dán nhãn “Positive”. Precision càng cao, tức là số điểm mô hình dự đoán là positive đều là positive càng nhiều.

Precision =

• Recall: chỉ số này còn được gọi là độ bao phủ tức là xem xét xem mô hình tìm được có khả năng tổng quát hóa như thế nào. Recall càng cao, tức là số điểm là positive bị bỏ sót càng ít.

Recall =

• F1-score : là trung bình điều hòa (harmon-icmean) của precision và recall

F1 =

# CÁC TINH CHỈNH MÔ HÌNH

## LSVC

Trong mô hình LSVC tham số đầu vào quan trọng nhất là tham số C.

Để tinh chỉnh mô hình sử dụng hàm GridSearchCV để tìm ra tham số C tốt nhất, trong đồ án chúng tôi so sánh tìm ra giá trị C tốt nhất trong khoảng dãy [0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4] và tìm ra tham số tốt nhất là C = 1 tức C mặc định.

## KNN

Trong mô hình KNN, tham số đầu vào quan trọng nhất là K tức số neighbors gần nhất để xác định nhãn cho giá trị đầu vào.

Để tinh chỉnh mô hình, cho thử K với giá trị từ 1 đến 50 sau đó dùng tập val để kiểm thử. K có độ sai càng thấp thì càng tốt

Chart, line chart

Description automatically generated

Hình 13: Kết quả độ sai so với tập val của K từ 1 đến 50

Qua biểu đồ trên ta tìm được giá trị K=22 có giá trị tốt nhất với ít lỗi nhất. Ta bỏ qua K=1 vì K khi đó quá nhỏ và dễ gây nhiễu.

## VGG16

Vì là một mô hình được huấn luyện sẵn, khả năng học hỏi của mô hình VGG16 là rất cao, chính vì vậy rất dễ xảy ra tình trang overfitting.

Chính vì vậy phải sử dụng phương pháp optimizer và Early stopping trên mô hình

Về optimizer thì chúng tôi sử dụng adam của thư viện keras với learning rate = 0.01.

Về Early stopping, quá trình huấn luyện sẽ dừng khi accuracy trên tập val không tăng trong vòng 5 chu kì huấn luyện, sử dụng model checkpoint (mc) để lưu lại mô hình có Accuracy tốt nhất.

# Kết quả

Sau khi tinh chỉnh mô hình nhóm chúng tôi thu được kết quả sau đây

## So sánh giá trị trung bình

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mô hình | Độ đo | | | |
| Precision | Recall | F1-score | Accuracy |
| VGG16 | 0.98 | 0.96 | 0.96 | 0.96 |
| LSVC | 0.84 | 0.81 | 0.81 | 0.84 |
| KNN | 0.68 | 0.45 | 0.38 | 0.68 |

Hình 14: Giá trị trung bình các độ đo của các mô hình

## So sánh kết quả với từng nhãn

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nhãn | Mô hình | | | | | |
| Vgg16 | | LSVC | | KNN | |
| Recall | F1-score | Recall | F1-score | Recall | F1-score |
| Flat | 0.94 | 0.94 | 0.95 | 0.85 | 0.90 | 0.54 |
| Full | 0.95 | 0.95 | 0.88 | 0.82 | 0.40 | 0.50 |
| Notire | 1.00 | 0.99 | 0.62 | 0.76 | 0.05 | 0.10 |

Hình 15: Giá trị Recall và F1-score với từng nhãn

# Phân tích lỗi và hướng phát triển

## Phân tích lỗi

Các lỗi chúng tôi mắc phải hầu hết đến từ tập dữ liệu. Tập dữ liệu có những vấn đề như sau:

* Hình ảnh lốp xe bị khuyết.
* Hình ảnh lốp xe ở các môi trường khác nhau: như tuyết hoặc bùn.
* Hình ảnh lốp xe của các loại xe ngoài xe ô tô còn ít

Ngoài ra, việc áp dụng data augmentation đã có thể làm ảnh hưởng đến một phần tính chất ban đầu của ảnh, làm việc học trở nên sai sót.

A picture containing text, road, outdoor, transport

Description automatically generatedA picture containing text, transport

Description automatically generated

Hình 16 và 17: Dự đoán sai vì không giống chuẩn đề ra.

A picture containing grass, outdoor, transport, handcart

Description automatically generatedA close-up of a car tire

Description automatically generated with medium confidence

Hình 18 và 19: Dự đoán sai vì thiếu hình ảnh của các lốp xe khác ô tô

A close up of a car tire

Description automatically generated with medium confidence

Hình 20: Dự đoán sai vì trong điều kiện trời tuyết

## Giải pháp

Để giải quyết các lỗi đã nêu ở mục A, nhóm chúng tôi có 2 giải pháp:

• Giảm thiểu số lượng ảnh xe bị nhiễu, bị khuyết

• Tăng số lượng hình ảnh lốp xe ở các loại phương tiện khác ô tô, số lượng lốp xe ở các môi trường khác.

# Kết luận và hướng Phát triển

## Kết luận

Qua việc áp dụng các kiến thức về Machine learning và Deep Learning nhóm chúng tôi đã tạo ra các mô hình máy học có kết quả tốt có Accuracy lên đến 96%.

Tuy nhiên việc sử dụng mô hình KNN lại không đạt được kết quả như mong đợi. Việc thu thập dữ liệu bằng cách thủ công còn nhiều vấn đề đặc biệt là vấn đề trùng lặp dữ liệu. Còn nhiều hạn chế ở bước tiền xử lý dữ liệu.

Nhưng nhìn chung nhóm đã đạt được mục tiêu đề ra ban đầu.

## Hướng phát triển

Nhóm chúng tôi sẽ tiếp tục thu thập và làm sạch dữ liệu để giải quyết các lỗi còn xót lại. Từ đó tạo ra mô hình có độ tin cậy cao hơn.

Phát triển các ứng dụng dựa trên mô hình, có thể tích hợp vào ứng dụng lớn để phân tích tình trạng ô tô.

Tài liệu Tham khảo

|  |  |
| --- | --- |
| [1] | H. Hà, "Tìm hiểu về phương pháp mô tả đặc trưng HOG (Histogram of Oriented Gradients)," [Online]. Available: https://viblo.asia/p/tim-hieu-ve-phuong-phap-mo-ta-dac-trung-hog-histogram-of-oriented-gradients-V3m5WAwxZO7. [Accessed 25 12 2021]. |
| [2] | H. C. Trung, "Giới thiệu về Support Vector Machine (SVM)," [Online]. Available: https://viblo.asia/p/gioi-thieu-ve-support-vector-machine-svm-6J3ZgPVElmB. [Accessed 25 12 2021]. |
| [3] | M. L. c. bản, "Bài 6: K-nearest neighbors," [Online]. Available: https://machinelearningcoban.com/2017/01/08/knn/. [Accessed 25 12 2021]. |
| [4] | G. Learning, "What is VGG16? — Introduction to VGG16," [Online]. Available: https://medium.com/@mygreatlearning/what-is-vgg16-introduction-to-vgg16-f2d63849f615. [Accessed 25 12 2021]. |