1. Haar Casade:
   1. Haar Casade là gì:

Haar Casade là một thuật toán dành cho máy học với chức năng nhận diện vật thể có trong ảnh hay 1 đoạn video. Thuật toán này dựa trên đề xuất của Paul Viola và Michael Jones

* 1. Phân tích Haar Casade



Thuật toán này bao gồm 4 giai đoạn:

* Lựa chọn đặc tính:



Đặc tính là những hình chữ nhật liền kề nhau nằm tại một vùng nhất định từ khung hình đang được xem xét, sau đó tính tổng các pixel lại để ra được những tổng khác nhau.

* Tạo ảnh tích hợp:

Những đặc tính sau khi tính toán thường không liên quan nhau.

Cho ví dụ với hình dưới đây:



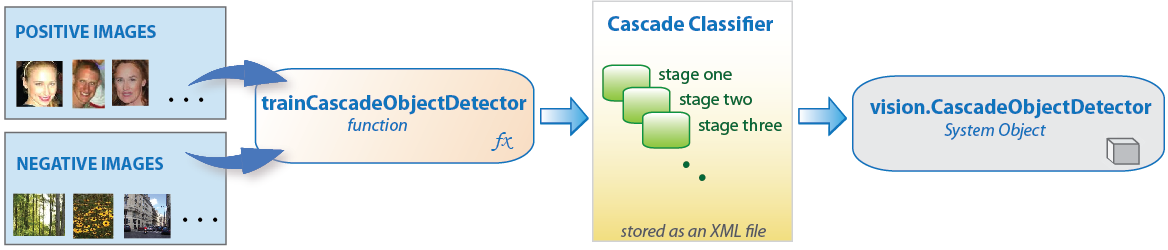
Ở đặc tính đầu tiên, tập trung vào vùng dưới trán. Vùng mắt sẽ nằm trong phần màu đen (tối hơn) và vùng mũi, má sẽ trong phần sang

Ở đặc tính thứ 2, tập trung từ trán tới sống mũi. Vùng mắt sẽ nằm 2 bên (màu đen) và sống mũi sẽ màu trắng (sáng hơn).

* Nhìn 2 đặc tính, không có gì liên quan nhau.
* Adaboost training:

Đây là một concept có chức năng chọn ra đặc tính tốt nhất và phân loại.

* Cascading Classifier:



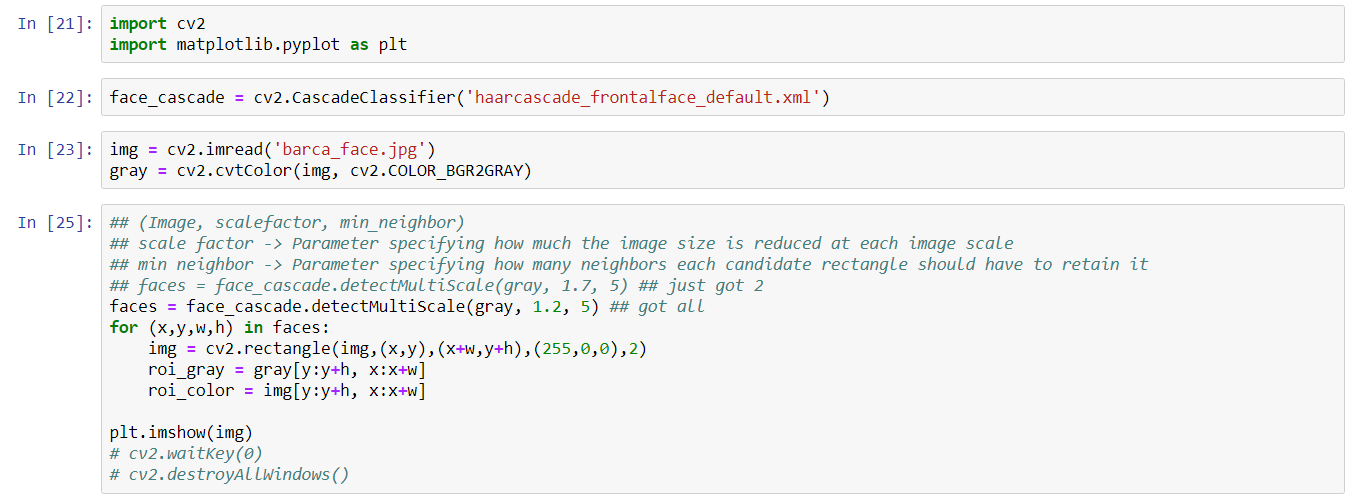
Đây là một chuỗi những giai đoạn khác nhau, những giai đoạn này là những đoạn máy học đơn giản. Những đoạn máy học này là bộ phân loại đơn giản gọi là *decision stumps.*

Mỗi giai đoạn gán mã cho một vùng hình đã cắt ra từ bức ảnh, và gán cho nó theo dạng nhị phân (Có hoặc không). Có ở đây là khung hình này chính là 1 vật thể. Còn không thì ngược lại.

Những gia đoạn này sẽ có 1 tỉ lệ đoán sai, điều này dùng để xem xét cho những giai đoạn sau đó (nên xét tiếp hay không).

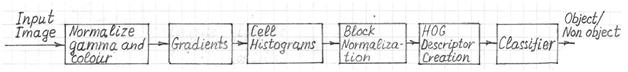
🡺Tổng hợp lại ta sẽ có những khung hình cắt thỏa yêu cầu.

* 1. Sử dụng Haar Cascade trong nhận diện khuôn mặt:





1. Histogram of Oriented Gradients (HOG):
   1. Lý thuyết:



Bước 1:

Chia ảnh thành những khối (ví dụ 16x16). Mỗi khối này lại chia nhỏ hơn (8x8). Thường những khối này đè chồng nhau, và từ mỗi khối đó ta thu được gradient. Để làm được điều này ta sử dụng thuật toán 1-D Sobel theo chiều dọc và chiều ngang như sau:

*Gx*(*y*,*x*) = *Y*(*y*,*x*+1) – *Y*(*y*,*x*-1); *Gy*(*y*,*x*) = *Y*(*y*+1,*x*) – *Y*(*y*-1,*x*)

Với:

Y(y, x) là cường độ điểm ảnh tại x, y.

*Gx*(*y*,*x*) là gradient theo chiều ngang.

*Gy*(*y*,*x*) là gradient theo chiều dọc.

Độ lớn và pha được định nghĩa như sau:

hog-object-detection-eq1

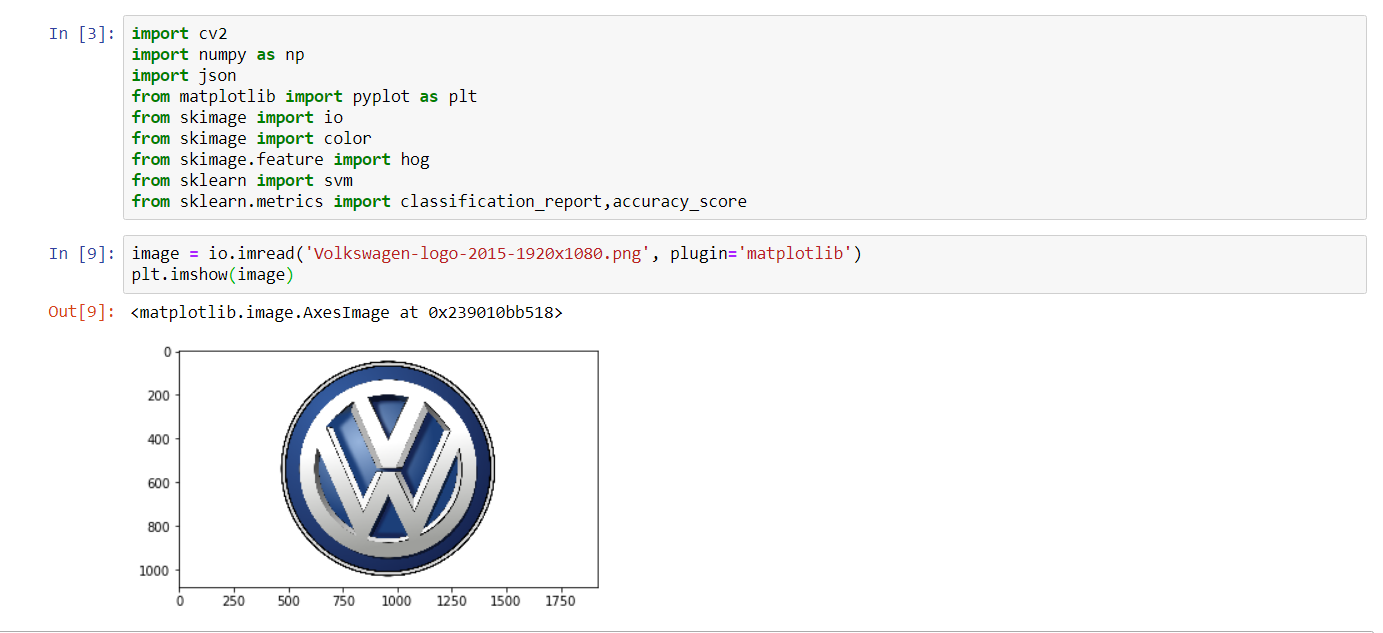
HOG được tạo ra cho từng cell, Chọn Q và hướng thì vô định cho nên góc sẽ quay từ nhỏ hơn 00 và tang đến 1800

Vì từng bức ảnh có từng độ tương phản khác nhau cho nên cân bằng độ tương phản có thể rất cần thiết. Cân bằng sử dụng vecto v với từng khối, một trong những công thức được dung như sau:

hog-object-detection-eq2

Và cuối cùng những đặc tính này được đưa vào SVM để huấn luyện phân lớp.

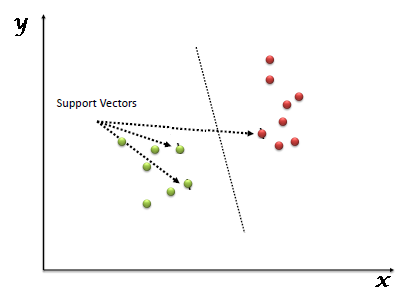
* 1. Ví dụ:





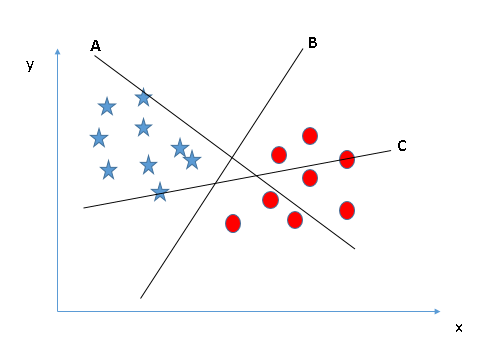
1. SVM:
   1. SVM là gì

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" phân chia các lớp. Đường bay - nó chỉ hiểu đơn giản là 1 đường thằng có thể phân chia các lớp ra thành hai phần riêng biệt.



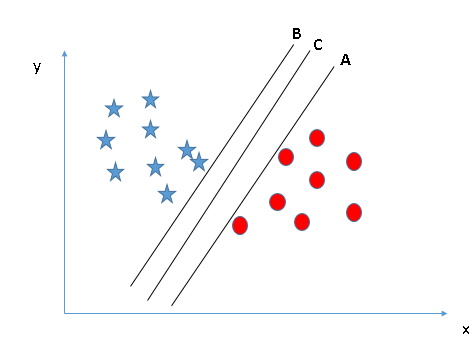
* 1. SVM làm việc như thế nào

- Identify the right hyper-plane (Scenario-1):



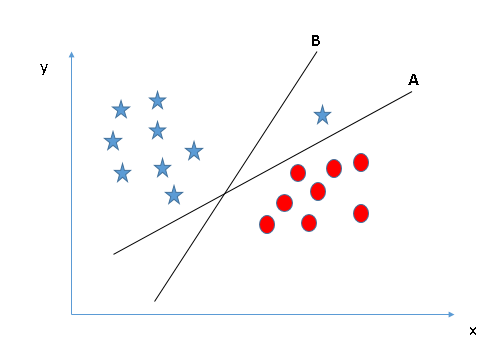
Quy tắc số một để chọn 1 hyper-lane, chọn một hyper-plane để phân chia hai lớp tốt nhất. Trong ví dụ này chính là đường B

- Identify the right hyper-plane (Scenario-2):



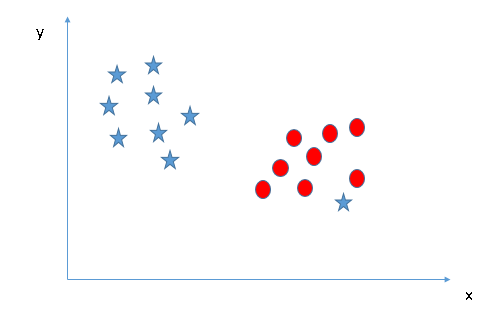
Quy tắc thứ hai chính là xác định khoảng cách lớn nhất từ điểu gần nhất của một lớp nào đó đến đường hyper-plane. Khoảng cách này được gọi là "Margin", Hãy nhìn hình bên dưới, trong đấy bạn có thể nhìn thấy khoảng cách margin lớn nhất đấy là đường C. Ở đây bạn nhớ nếu chọn lầm hyper-lane có margin thấp hơn thì sau này khi dữ liệu tăng lên thì sẽ sinh ra nguy cơ cao về việc xác định nhầm lớp cho dữ liệu

- Identify the right hyper-plane (Scenario-3):



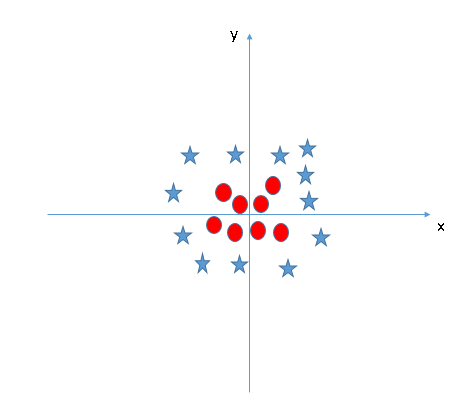
Có thể có một vài bạn sẽ chọn đường B bởi vì nó có margin cao hơn đường A, nhưng đấy sẽ không đúng bởi vì nguyên tắt đầu tiên sẽ là nguyên tắc số 1., chúng ta cần chọn hyper-plane để phân chia các lớp thành riêng biệt. Vì vậy đường A mới là lựa chọn chính xác.

- Can we classify two classes (Scenario-4)?

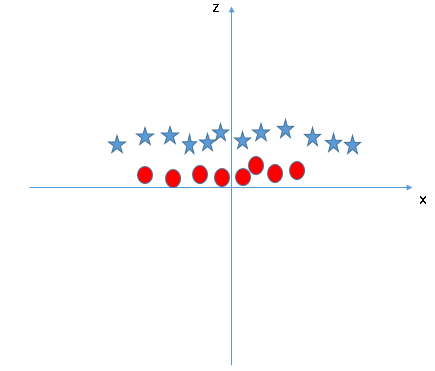


Một ngôi sao ở bên ngoài cuối được em như một ngôi sao phía ngoài hơn, SVM có tính năng cho phép bỏ qua các ngoại lệ và tìm ra hyper-plane có biên giới tối đa . Do đó chúng chúng ta có thể nói, SVM có khả năng mạnh trong việc chấp nhận ngoại lệ.

Find the hyper-plane to segregate to classes (Scenario-5)



 SVM có thể giải quyết vấn đề này, Khá đơn giản, nó sẽ được giải quyết bằng việc thêm một tính năng, Ở đây chúng ta sẽ thêm tính năng z = x^2+ y^2. Bây giờ dữ liện sẽ được biến đổi theo trục x và z như sau



* Tất cả dữ liệu trên trục z sẽ là số dương vì nó là tổng bình phương x và y
* Trên biểu đồ các điểm tròn đỏ xuất hiện gần trục x và y hơn vì thế z sẽ nhỏ hơn => nằm gần trục x hơn trong đồ thị (z,x)

Trong SVM, rất dễ dàng để có một siêu phẳng tuyến tính (linear hyper-plane) để chia thành hai lớp, Nhưng một câu hỏi sẽ nảy sinh đấy là, chúng ta có cần phải thêm một tính năng phân chia này bằng tay hay không. Không, bởi vì SVM có một kỹ thuật được gọi là kernel trick ( kỹ thuật hạt nhân), đây là tính năng có không gian đầu vào có chiều sâu thấm và biến đổi nó thành không gian có chiều cao hơn, tức là nó không phân chia các vấn đề thành các vấn đề riêng biệt, các tính năng này được gọi là kernel. Nói một cách đơn giản nó thực hiện một số biết đổi dữ liệu phức tạp, sau đó tìm ra quá trình tách dữ liệu dựa trên các nhãn hoặc đầu ra mà chúng ra đã xác định trước.