TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

──────── \* ────────

Ảnh có chứa văn bản, ký hiệu

Mô tả được tạo tự động

**BÀI TẬP LỚN**

**HỌC MÁY VÀ KHAI PHÁ DỮ LIỆU**

**Đề tài: Object detection**

                   Lớp                                  :

|  |  |
| --- | --- |
| Đinh Chí Công | 20193996 |
|  |  |
|  |  |
|  |  |

                   Giáo viên hướng dẫn.      :

                   Nhóm sinh viên thực hiện :

Mục lục

[1. Giới thiệu đề tài 2](#_Toc91235398)

[1.1. Mô tả bài toán 2](#_Toc91235399)

[1.2. Phân tích chi tiết 2](#_Toc91235400)

[1.3. Phân công nhiệm vụ 3](#_Toc91235401)

[2. Cơ sở lý thuyết và giải pháp 4](#_Toc91235402)

[2.1. Cơ sở lý thuyết 4](#_Toc91235403)

[2.2. Giải pháp 4](#_Toc91235404)

[3. Model và thuật toán sử dụng 5](#_Toc91235405)

[3.1. Model 5](#_Toc91235406)

[3.2. Thuật toán 7](#_Toc91235407)

[4. Quy trình thực hiện 8](#_Toc91235408)

[4.1. Data 8](#_Toc91235409)

[4.1.1. Tìm hiểu dữ liệu 8](#_Toc91235410)

[4.1.2. Chuẩn bị data 9](#_Toc91235411)

[4.1.3. tiền xử lý data 9](#_Toc91235412)

[4.2. Model 10](#_Toc91235413)

[4.2.1. VGG 10](#_Toc91235414)

[4.2.2. Extract 11](#_Toc91235415)

[4.2.3. Loc 11](#_Toc91235416)

[4.2.4. Conf 11](#_Toc91235417)

[4.2.5. dBox 11](#_Toc91235418)

[5. Kết quả và đánh giá 11](#_Toc91235419)

[6. Hướng phát triển 12](#_Toc91235420)

[Tài liệu tham khảo 13](#_Toc91235421)

# Giới thiệu đề tài

## Mô tả bài toán

Đầu tiên đối với bài toán Object Detection này input là một bức ảnh bình thường,và output là một bounding box xung quanh đối tượng, mỗi đối tượng là một bouding box , bounding box này không chỉ là cái hình vuông mà còn là độ cao, rộng, vị trí của nó ở đâu mà còn cả nhãn (lable) tức là con người, con vật hay oto như trong hình bên dưới đây. Thậm chí có cả accuracy ( độ chính xác) của cái box là bao nhiêu trong khoảng [0,1] càng cao thì độ chính xác càng được khẳng định.

## Phân tích chi tiết

Bài toán là sự kết hợp giữa Classtification và Localization

Một object sẽ bao gồm:

* Tên
* Bounding box ( 2 góc)

INPUT:

Một bức ảnh

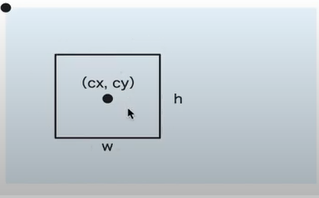
OUTPUT:

Bounding box:

Ảnh có chứa văn bản, cây, ngoài trời, bầu trời

Mô tả được tạo tự động

Trong bouding box nó có những gì?? Như hình bên dưới thì ta có một bức ảnh có dấu chấm đen ở vị trí (0,0) còn bên trong là một box với vị trí x,y cụ thể x\_min, y\_min, x\_max, y\_max hay vị trí top\_left, botton\_right. Hiện tại thì người ta đã biến đổi một chú đó là sử dụng c\_x,c\_y điểm ở vị trí trung tâm của box và width, height của box. Đối với phương pháp sử dụng SSD thì số lượng class sẽ bao gồm cả backgroud. Tức là trong cái output ngoài class người, ngựa mà còn cả backgroud,nên sẽ có 3 class.



Vị trí của bounding box xác định dựa vào tâm (cx, cy) và kích thước phụ thuộc vào w và h  
thông số : tên object, độ tự tin,

Class number:

Nền là 1 class riêng ( đây là quy định chung )

confidence

## Phân công nhiệm vụ

# Cơ sở lý thuyết và giải pháp

## Cơ sở lý thuyết

(sẽ làm như nào để giải quyết bài toán) cần khoanh các box, gộp ,….

## Giải pháp

# Model và thuật toán sử dụng

## Model

<https://vjol.info.vn/index.php/tckhvl/article/view/59526/49885>

Đã bắt đầu xuất hiện các thuật toán objectdetection từ 2001 và tới nay có rất nhiều

Ảnh có chứa văn bản

Mô tả được tạo tự động

Các phương pháp trên thì phần classitification không cải tiến nhiều thuật toán, chủ yếu sự cải tiến đến từ phương pháp tìm ra window

* Lựa chọn SSD

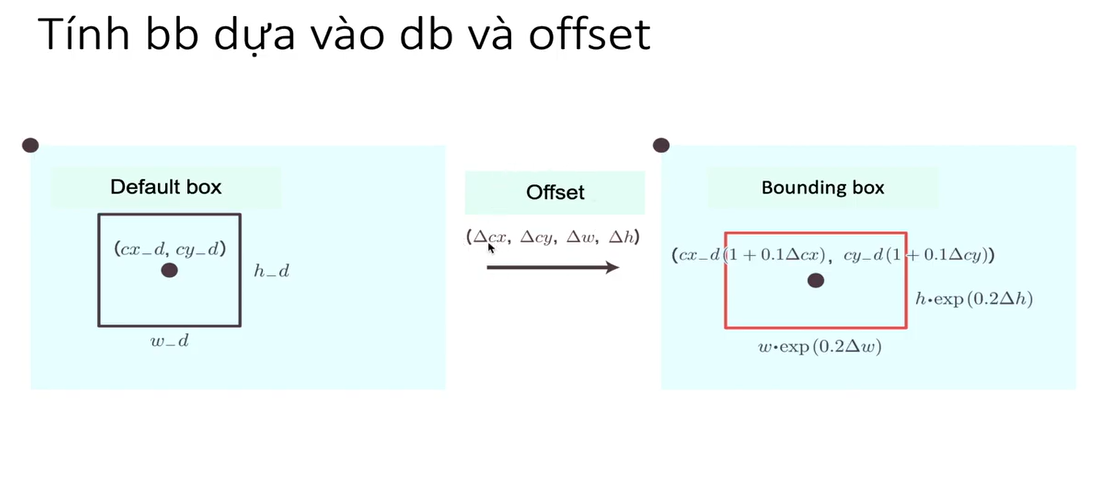
SSD có 2 loại

SSD 300

SSD 512

Ảnh có chứa văn bản

Mô tả được tạo tự động

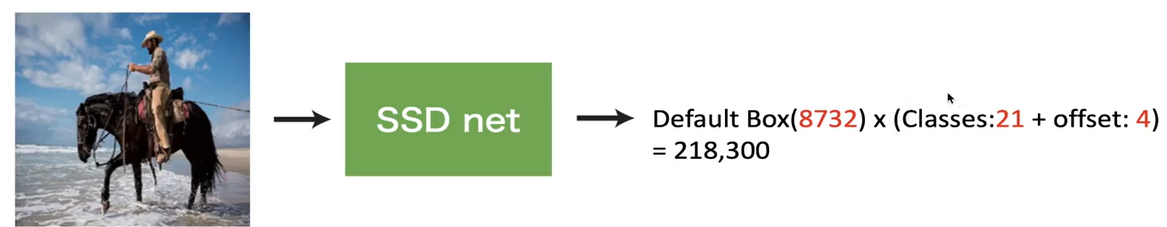


Quá trình chạy SSD bao gồm 6 bước

Bước 1: Resize ảnh về 300x300

Bước 2: Tạo ra 8732 default box

Bước 3: Truyền input vào mạng ssd



Bước 4: Lấy bb có confidence cao (200 trong số 8732)

Bước 5: Dùng NMS

Vì đầu ra sẽ có thể có nhiều bb cùng trỏ vào 1 object nên thuật toán này lựa chọn xem bb nào có độ tự tin cao nhất

Bước 6: Chọn ngưỡng threshold, độ tự tin mà lớn hơn ngưỡng này thì sẽ đưa ra kết quả đó

## Thuật toán

# Quy trình thực hiện

## Data

### Tìm hiểu dữ liệu

Đối với xử lý ảnh việc sử dụng data là rất nhiều, có những chỗ cung cấp free cho mình sử dụng, nhưng cũng có data không được cung cấp như thế (thương mại). Với việc làm một bài toán Computer Vision thì trước hết ta phải tìm được nguồn data , các đặc tính của data. Ví dụ như ta có tập data của người Châu Mĩ, Phi khi ta train được model khi áp dụng lên người Việt Nam thì độ chính xác lại không cao do đăc điểm khác nhau, phân bố khác nhau.

Qua tìm hiểu chúng em quyết định sử dụng VOC 2012 với 20 class(người, ngựa, máy bay,....) với 5717 image cho việc traning, 5823 image cho validation.Và như đã nói ở các phần trên thì do tính cả class là background nên ta sẽ có 21 class tất cả, ta nên chú ý đến cái này. Tiêp theo là xml annotation với x\_min, y\_min, x\_mã, y\_max, lable. Khái niệm annotation là các thông tin của bức ảnh dành cho việc trainning như là bounding box ở đâu, nhãn là cái gì.

Một bức ảnh bắt đầu từ điểm (1,1)

Dữ liệu bao gồm phần annotation để miêu tả về các thông tin trong bức

Ảnh có chứa văn bản

Mô tả được tạo tự động

Dữ liệu gồm các trường bên trên

### Chuẩn bị data

prepare\_data.ipyn

2 cái file get

### tiền xử lý data

data\_transform

tiền xử lý ảnh trong lúc tranning và val

trainning:

* thay đổi màu sắc
* thay đổi size của ảnh
* random cut
* trừ đi các thông số trung bình của ảnh  đưa về dạng chuẩn

val: chỉ resize về 300\*300, trừ đi các màu trung bình trong các chanel để đưa về dạng tiêu chuẩn

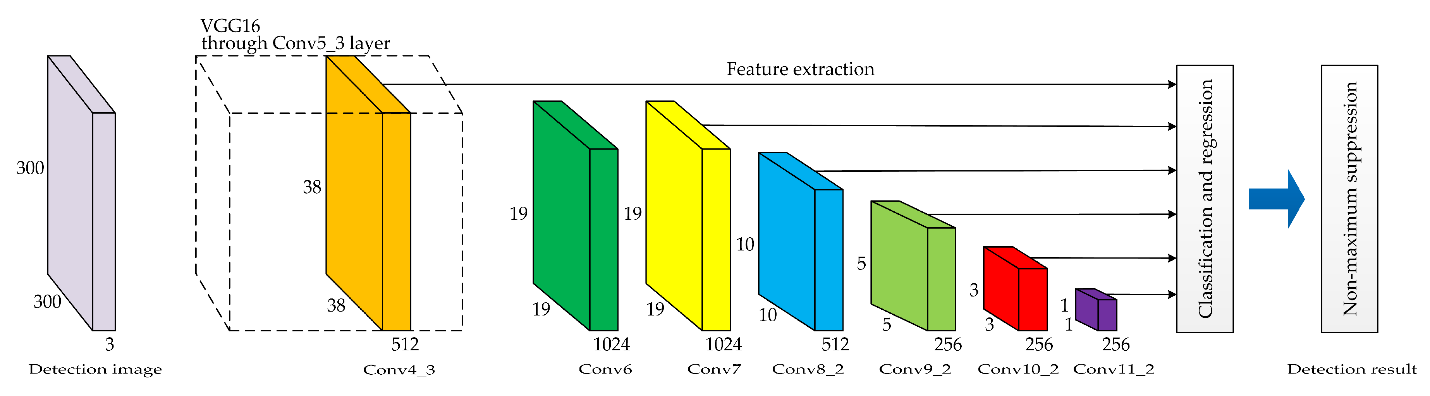
sử dụng augumentation của chính tác giả

## Model

Ảnh có chứa bàn

Mô tả được tạo tự động

Bao gồm 4 modul



cụ thể xem video https://www.youtube.com/watch?v=FsBtkzVPEY4&list=PLVnyU2MngPdbMX8NmJ5VFwD\_g6Al4fH1m&index=7

### VGG

trích xuất thông tin

conv4\_3 có 512 channels và kích thước 38\*38

### Extract

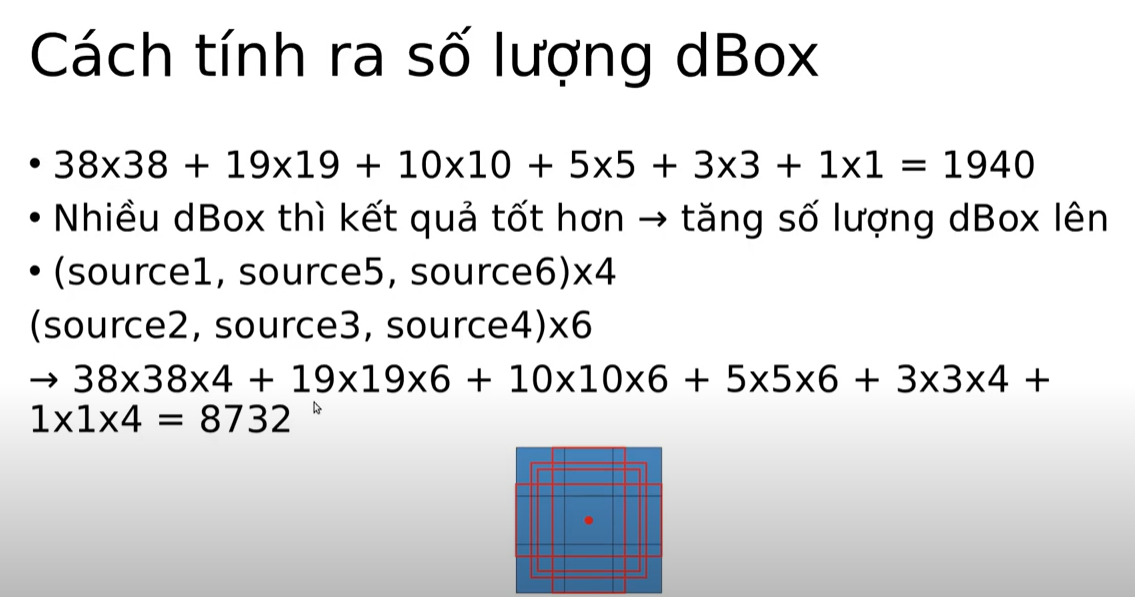
### Loc

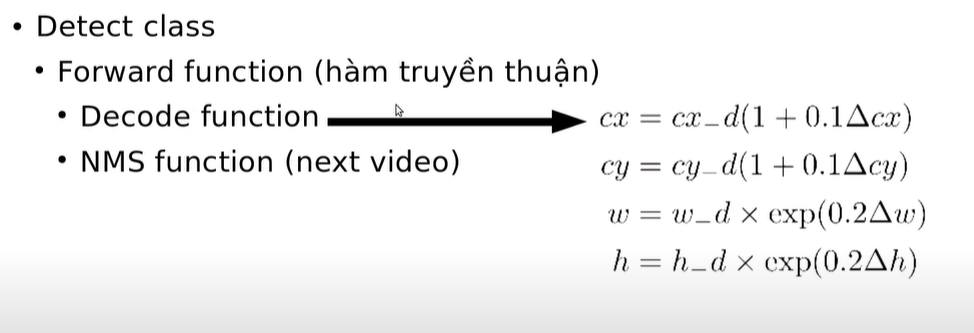
Tìm ra các thông số về location, offset

### Conf

Đưa ra độ tự tin

### dBox

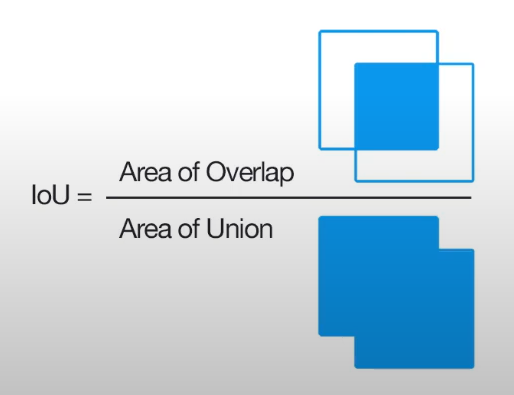




### Detect function

decode

nms function



nếu trùng lớn hơn 0.45 thì bỏ

# Kết quả và đánh giá

# Hướng phát triển

# Tài liệu tham khảo

https://viblo.asia/p/tim-hieu-ve-ssd-multibox-real-time-object-detection-3P0lPEPG5ox