

Bài 9. Phân hiện đối tượng (Object Detection)

AI Academy Vietnam

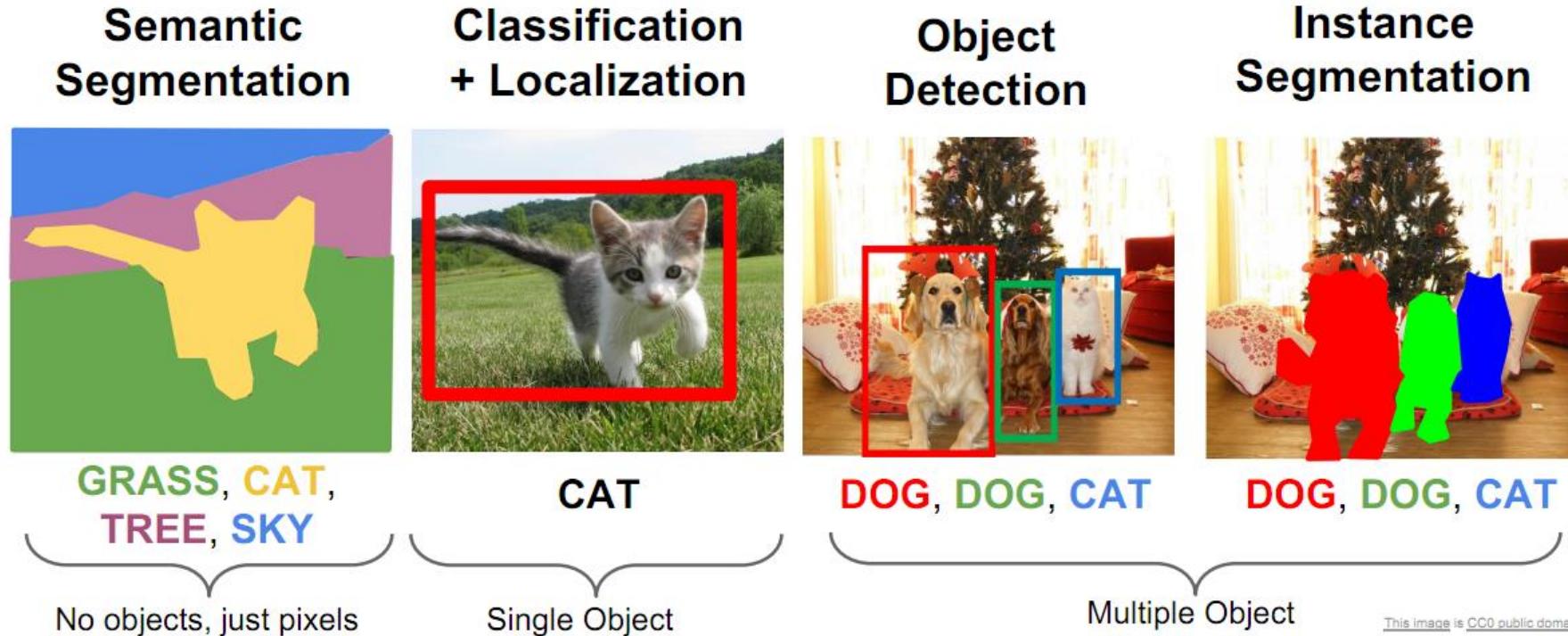
Nội dung buổi học

- Giới thiệu về bài toán phát hiện đối tượng
- Một số kỹ thuật truyền thống
- Các kỹ thuật học sâu



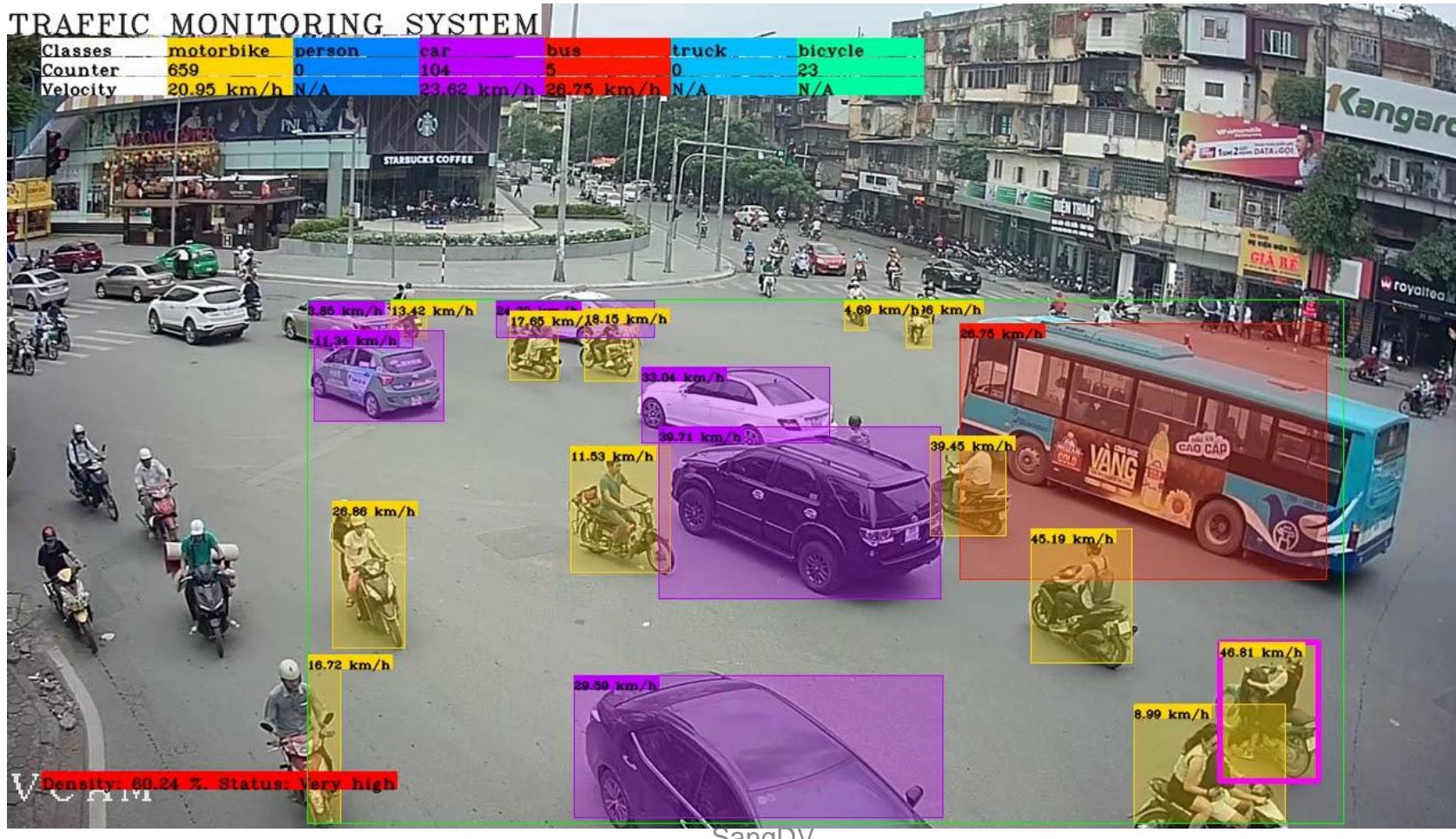
Giới thiệu bài toán phát hiện đối tượng

Các bài toán thị giác máy



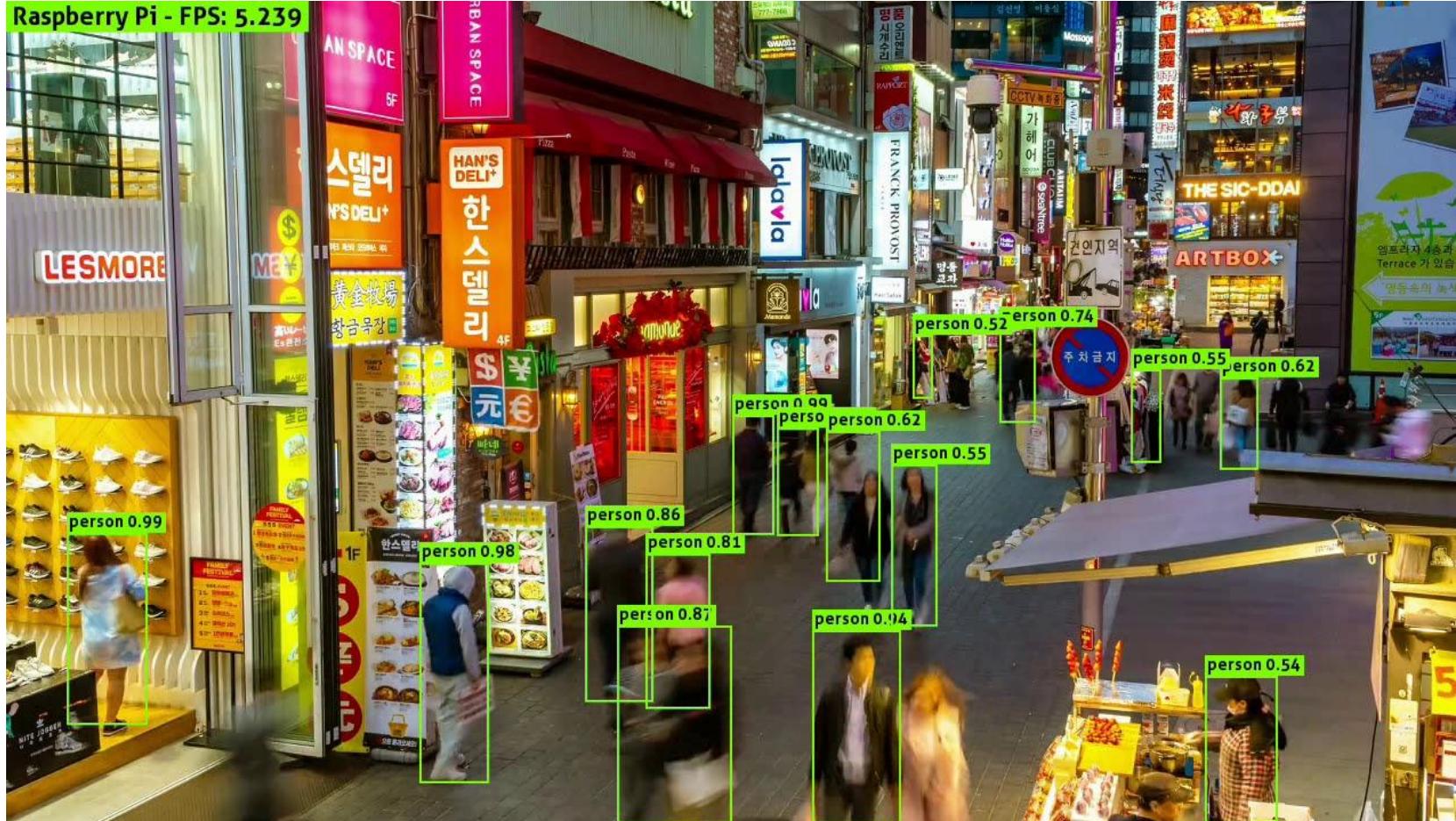
Một số ứng dụng bài toán phát hiện đối tượng

- Giao thông thông minh



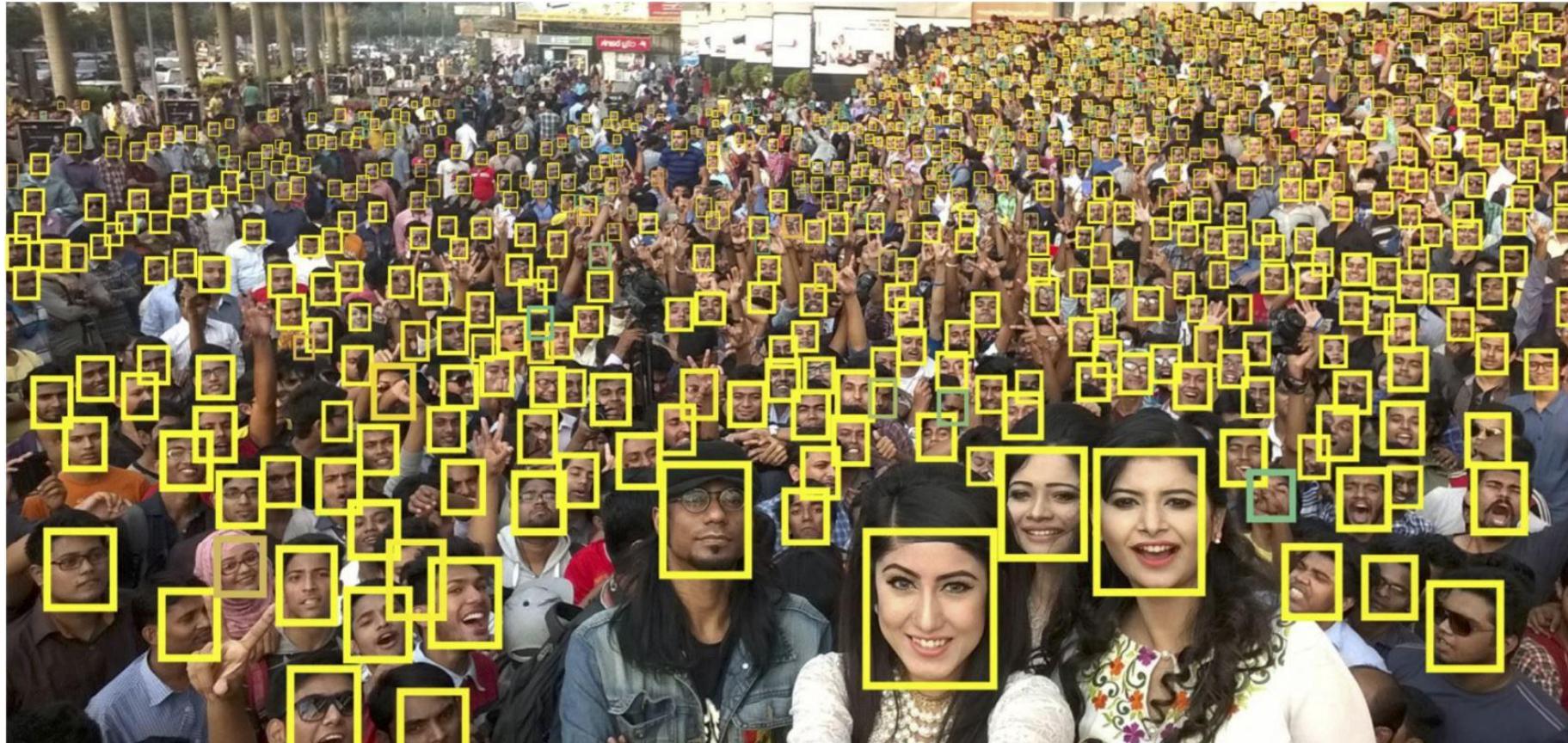
Một số ứng dụng bài toán phát hiện đối tượng

- Phát hiện người



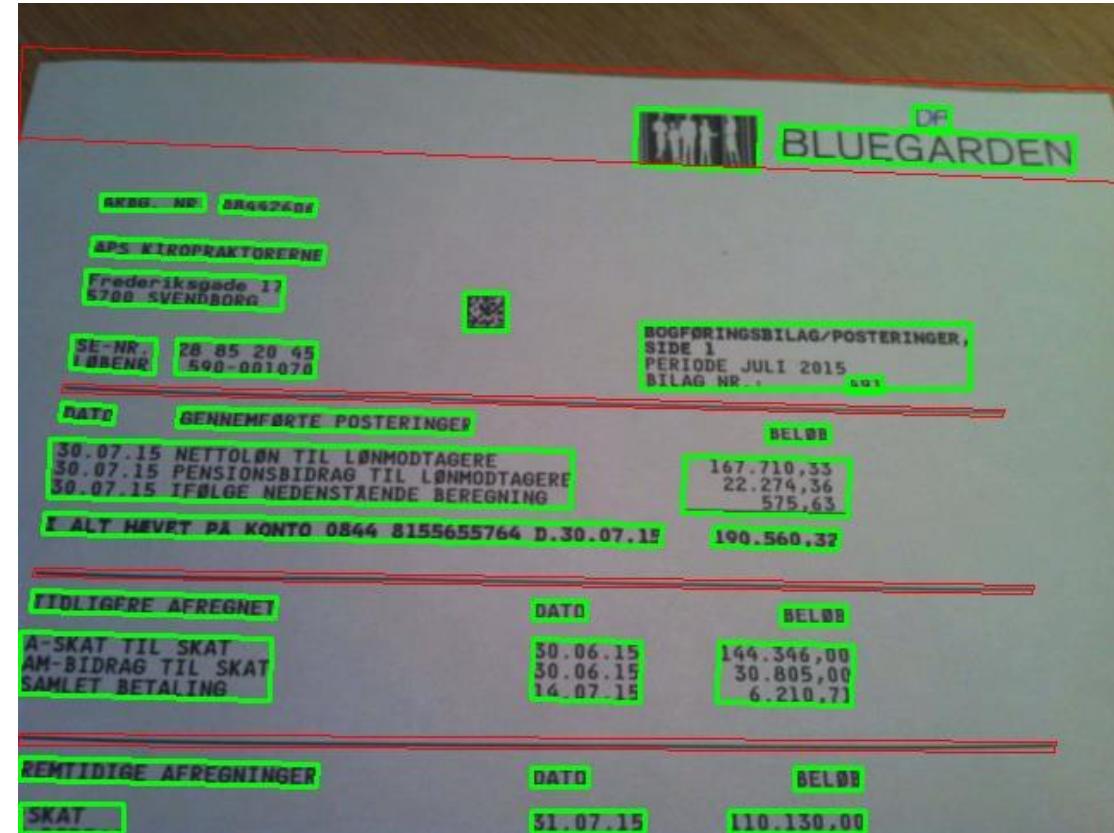
Một số ứng dụng bài toán phát hiện đối tượng

- Phát hiện khuôn mặt



Một số ứng dụng bài toán phát hiện đối tượng

- Phát hiện văn bản



Một số ứng dụng bài toán phát hiện đối tượng

- Robot tự động hái dâu





Các kỹ thuật truyền thống

Cửa sổ trượt

- Huấn luyện một bộ phân loại nhị phân



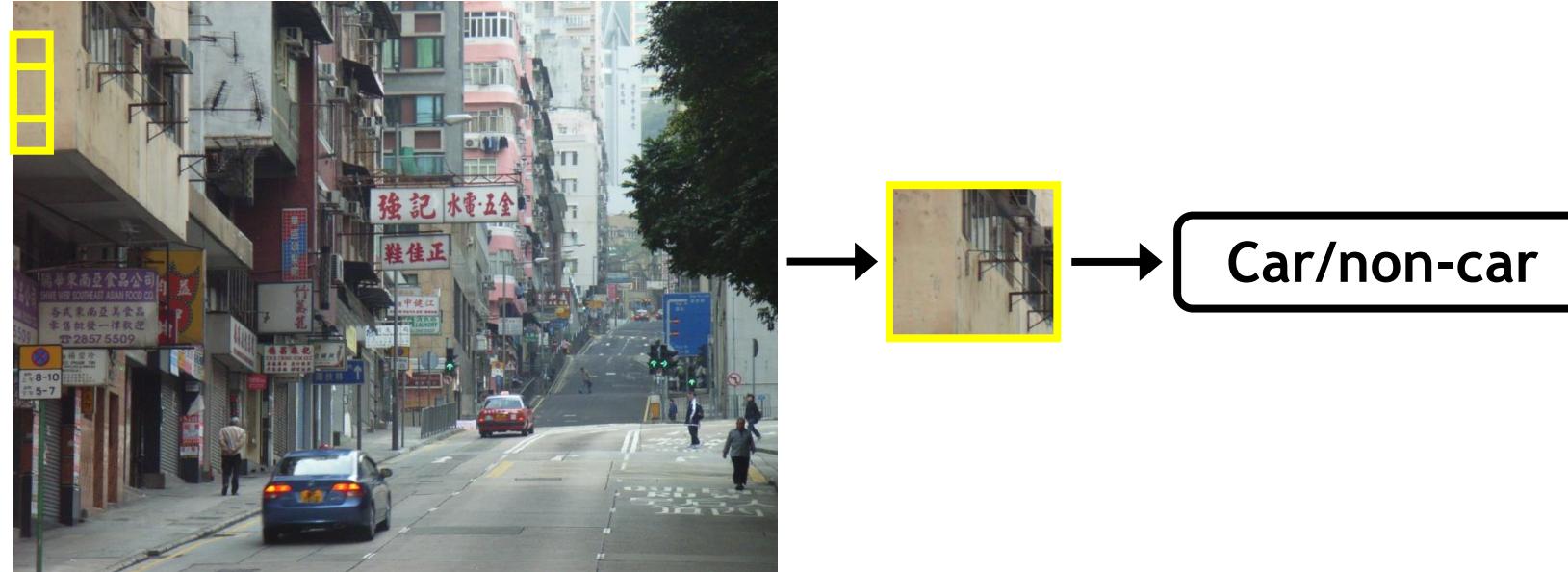
→ Car/non-car



No Yes or not car.

Cửa sổ trượt

- Sinh các vùng cửa sổ và đánh giá điểm từng cửa sổ



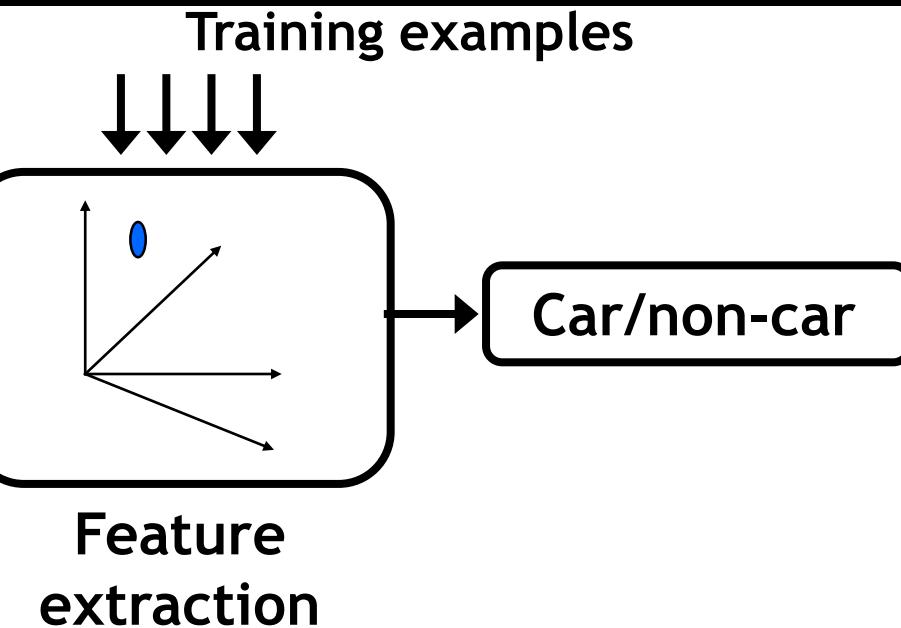
Cửa sổ trượt

- **Huấn luyện:**

1. Thu thập dữ liệu
2. Lựa chọn đặc trưng
3. Xây dựng bộ phân loại

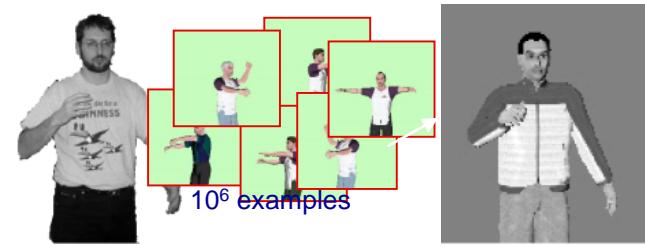
- **Cho một ảnh mới:**

1. Trượt cửa sổ
2. Đánh giá bằng bộ phân loại

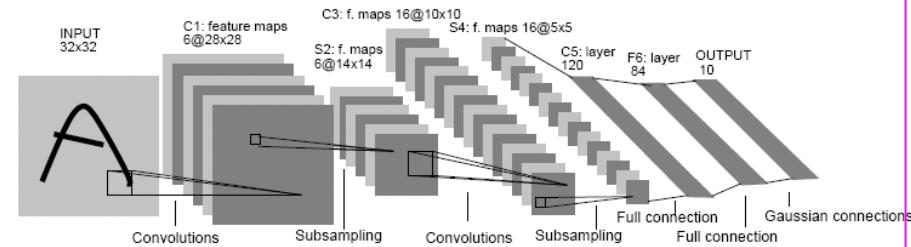


Các bộ phân loại khác nhau

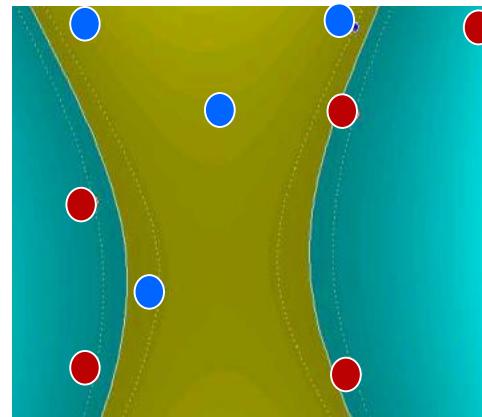
Nearest neighbor



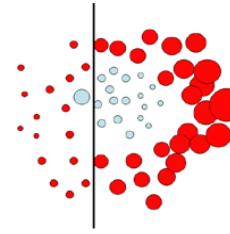
Neural networks



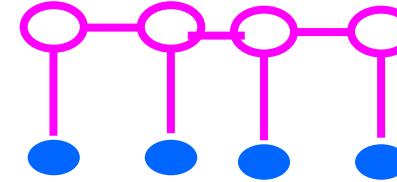
Support Vector Machines



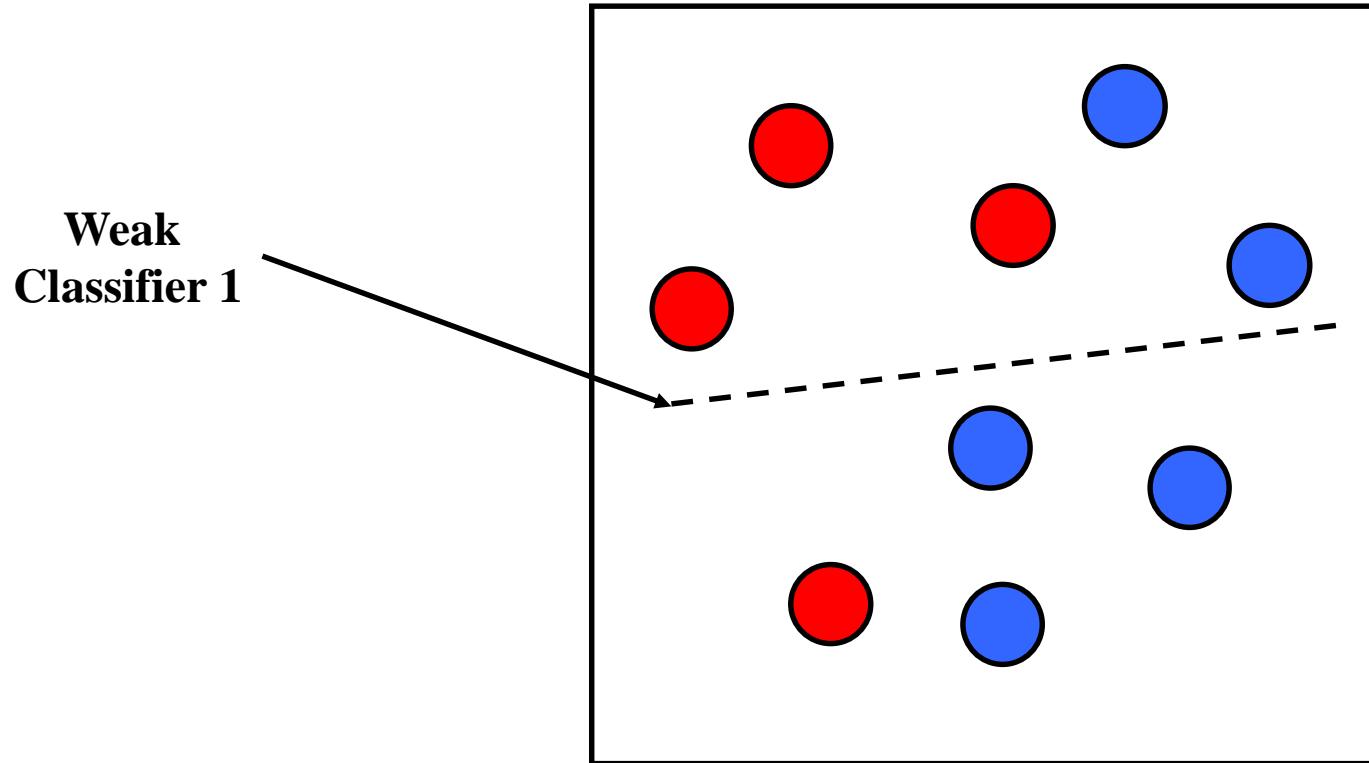
Boosting



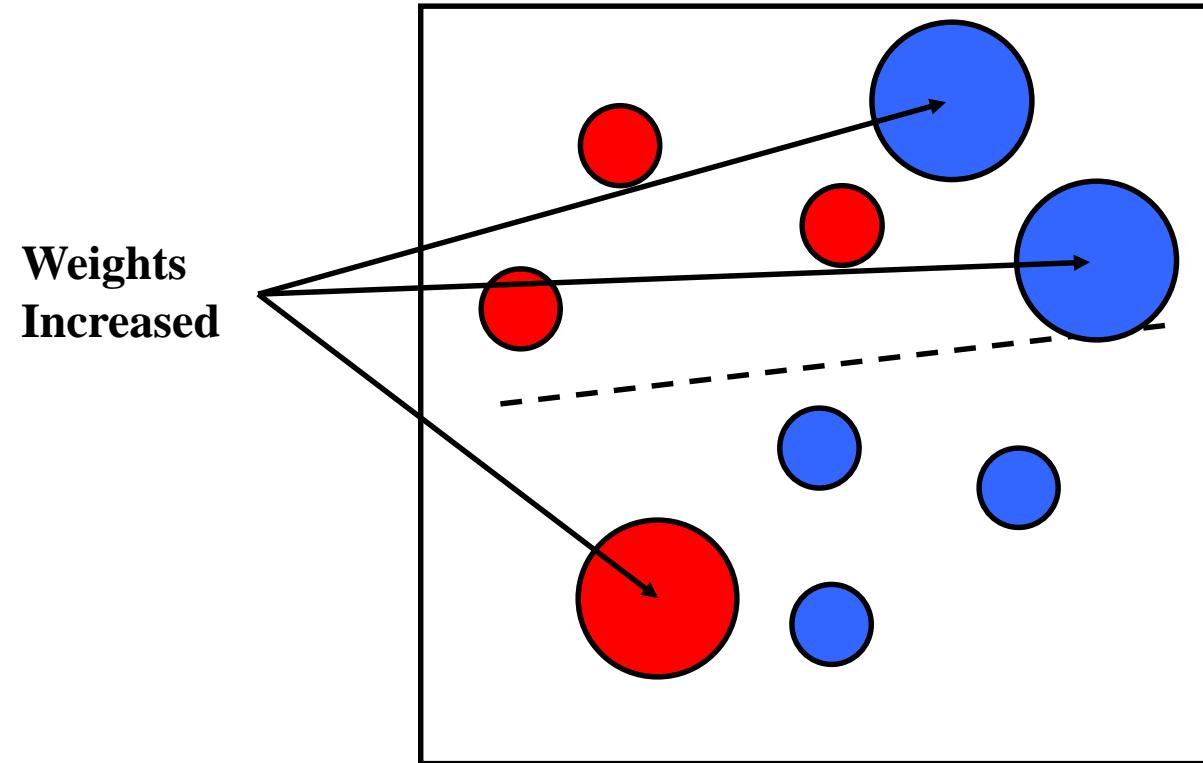
Conditional Random Fields



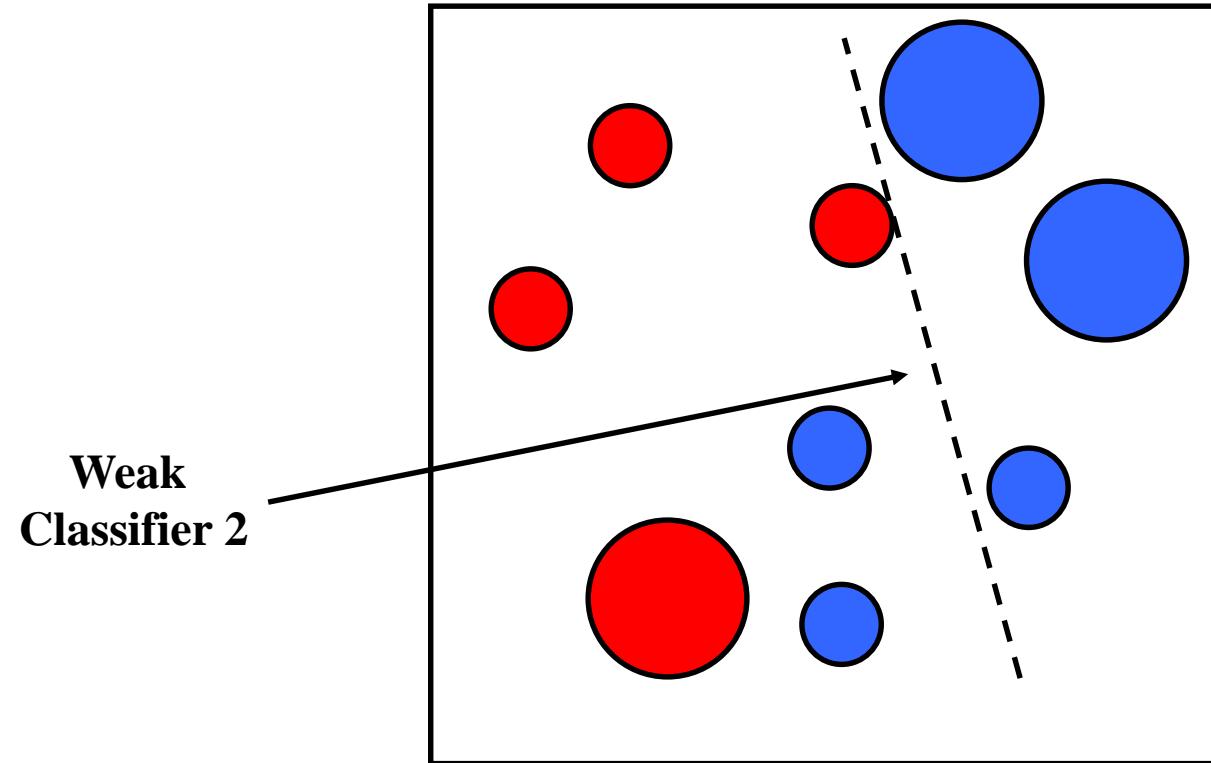
Minh họa Boosting



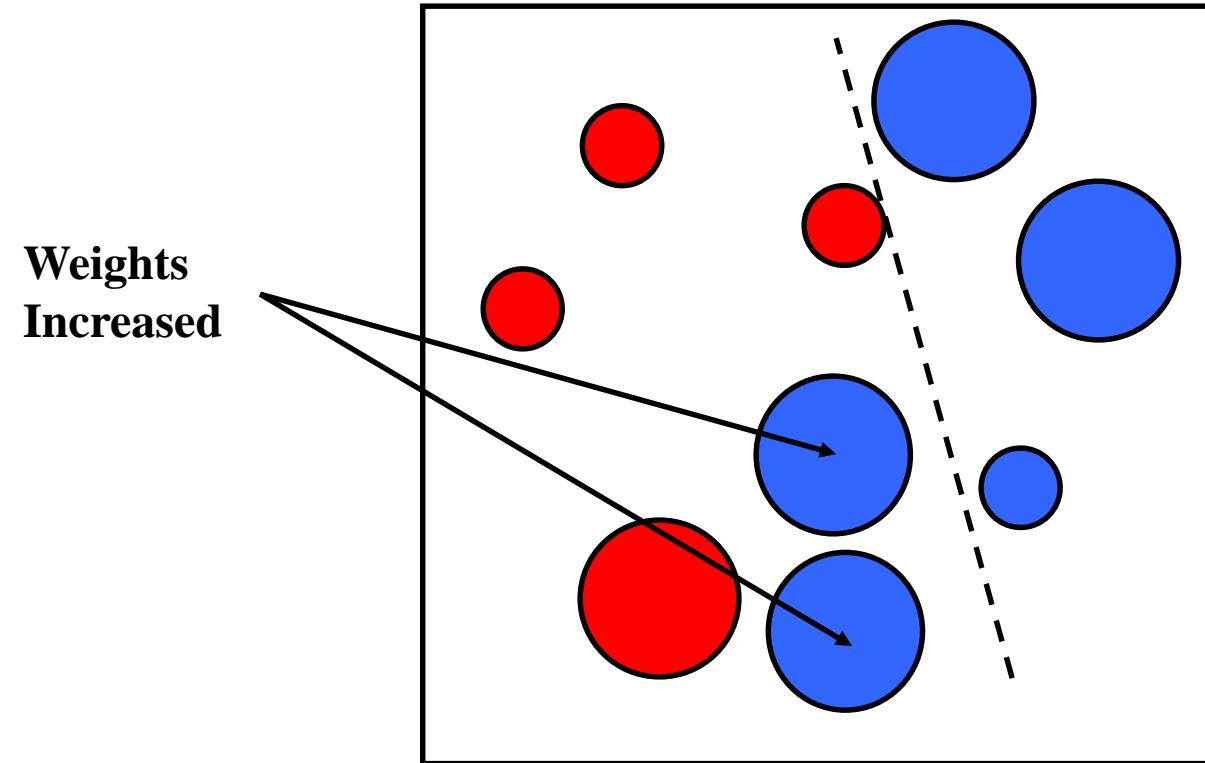
Minh họa Boosting



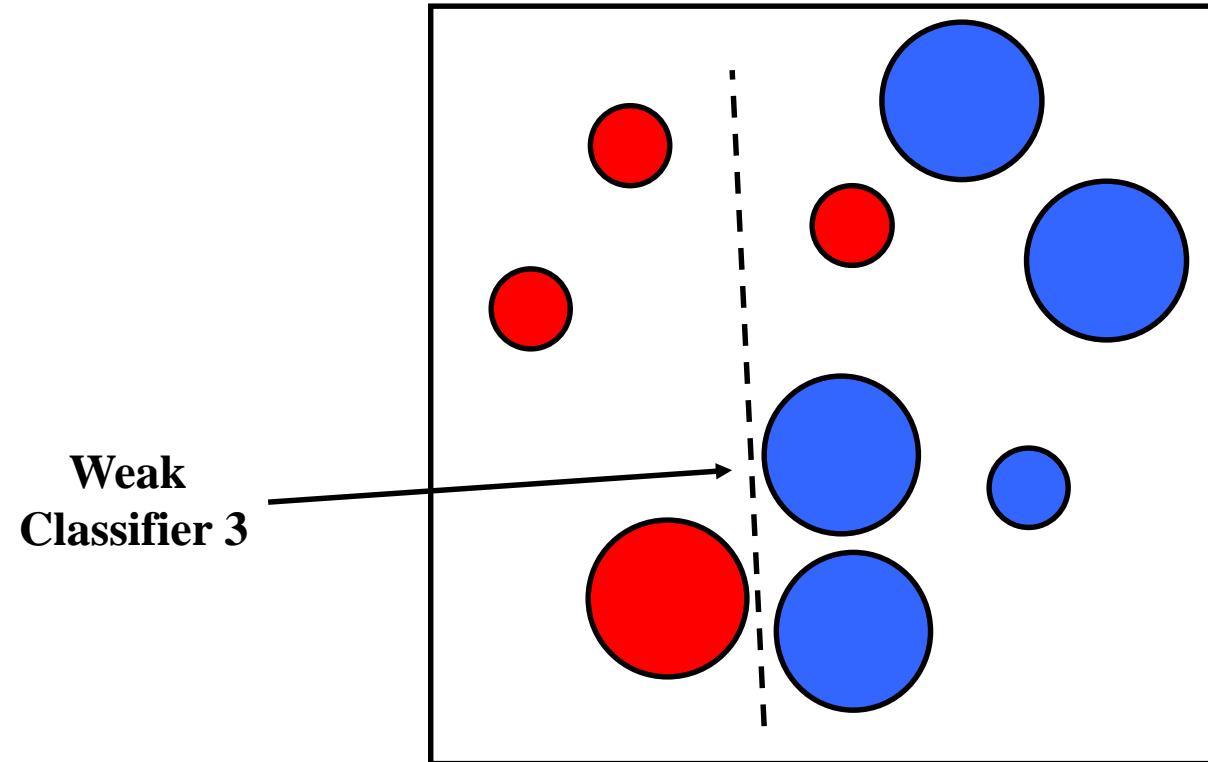
Minh họa Boosting



Minh họa Boosting

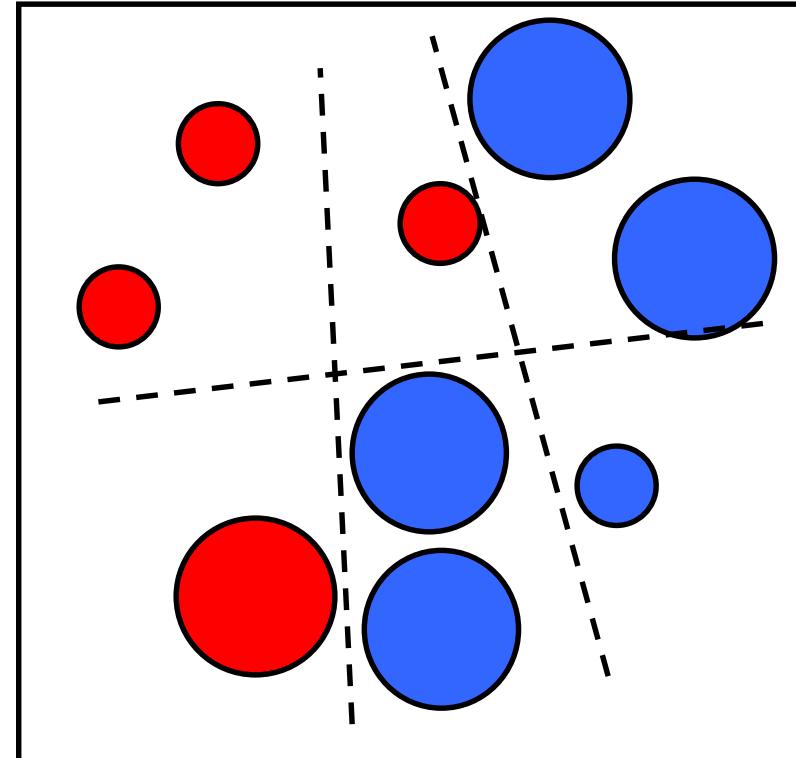


Minh họa Boosting



Minh họa Boosting

Bộ phân loại cuối cùng kết hợp các bộ phân loại yếu



Viola-Jones face detector

ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION 2001

Rapid Object Detection using a Boosted Cascade of Simple Features

Paul Viola
viola@merl.com
Mitsubishi Electric Research Labs
201 Broadway, 8th FL
Cambridge, MA 02139

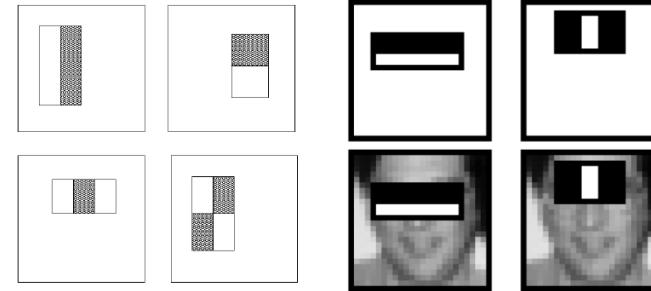
Michael Jones
mjones@crl.dec.com
Compaq CRL
One Cambridge Center
Cambridge, MA 02142

Abstract

This paper describes a machine learning approach for vi-

tected at 15 frames per second on a conventional 700 MHz Intel Pentium III. In other face detection systems, auxiliary information, such as image differences in video sequences,

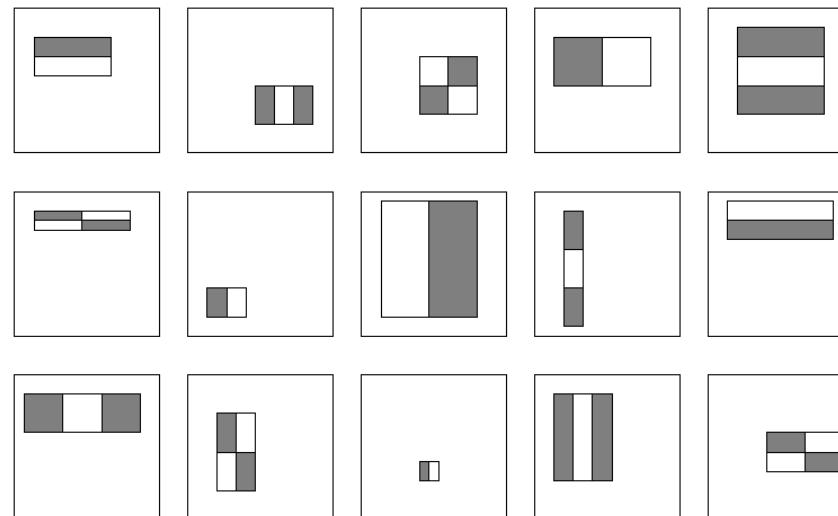
Viola-Jones detector: đặc trưng



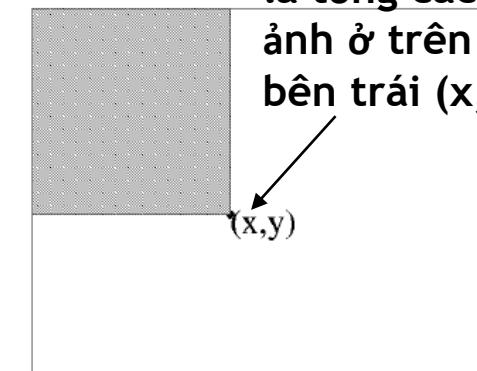
Bộ lọc “hình chữ nhật”

Độ chênh lệch giữa vùng đen và trắng

Tính toán rất nhanh nhờ ảnh tích phân



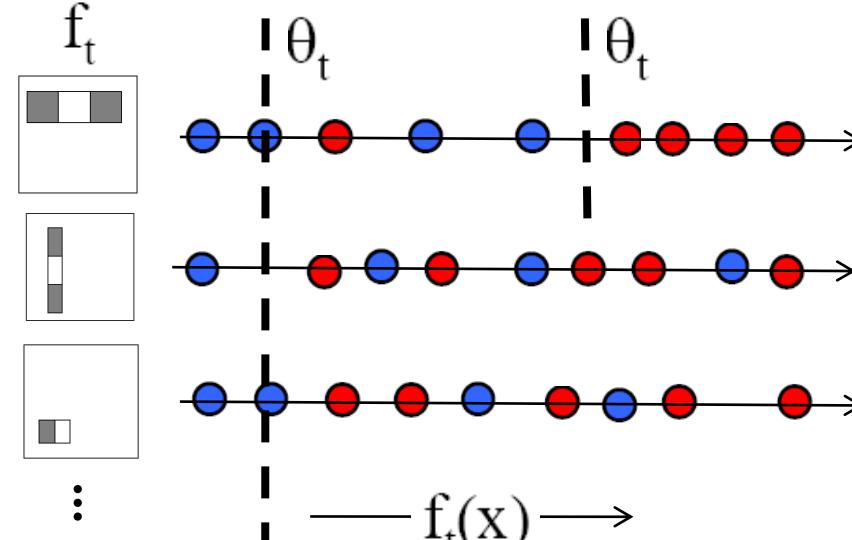
Giá trị tại (x,y)
là tổng các điểm
ảnh ở trên và
bên trái (x,y)



Ảnh tích phân

Viola-Jones detector: AdaBoost

- Muốn lựa chọn hình chữ nhật đặc trưng và ngưỡng để phân chia tốt nhất mẫu **dương tính** (faces) **âm tính** (non-faces), nghĩa là cực tiểu hóa *weighted error*.



Giá trị đặc trưng trên
tập huấn luyện faces
và non-faces.

Chọn ra bộ phân loại yếu:

$$h_t(x) = \begin{cases} +1 & \text{if } f_t(x) > \theta_t \\ -1 & \text{otherwise} \end{cases}$$

Bước tiếp theo, đánh trọng số lại
các mẫu dựa theo sai số, và chọn
cặp đặc trưng/ngưỡng khác

- Given example images $(x_1, y_1), \dots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \dots, T$:

- Normalize the weights,

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$

so that w_t is a probability distribution.

- For each feature, j , train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t , $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$.
- Choose the classifier, h_t , with the lowest error ϵ_t .
- Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

- The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

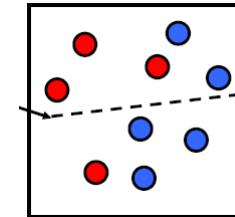
where $\alpha_t = \log \frac{1}{\beta_t}$

Giải thuật AdaBoost



Bắt đầu với
trọng số đồng
đều

For T vòng



$\{x_1, \dots, x_n\}$

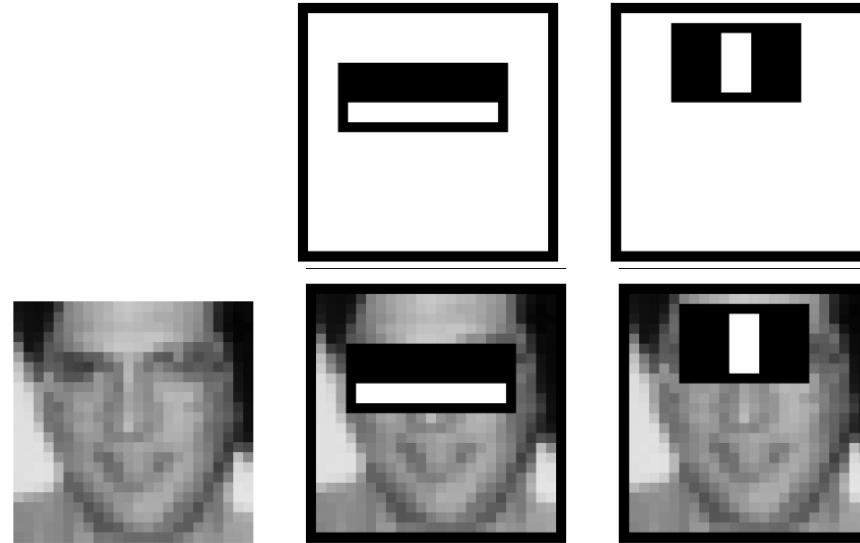
Đánh giá weighted error cho
từng đặc trưng và chọn cái tốt
nhất

Đánh lai trọng số các mẫu:

Mẫu sai \rightarrow tăng trọng số
Mẫu đúng \rightarrow giảm trọng số

Bộ phân loại cuối là kết hợp các bộ
phân loại yếu trên

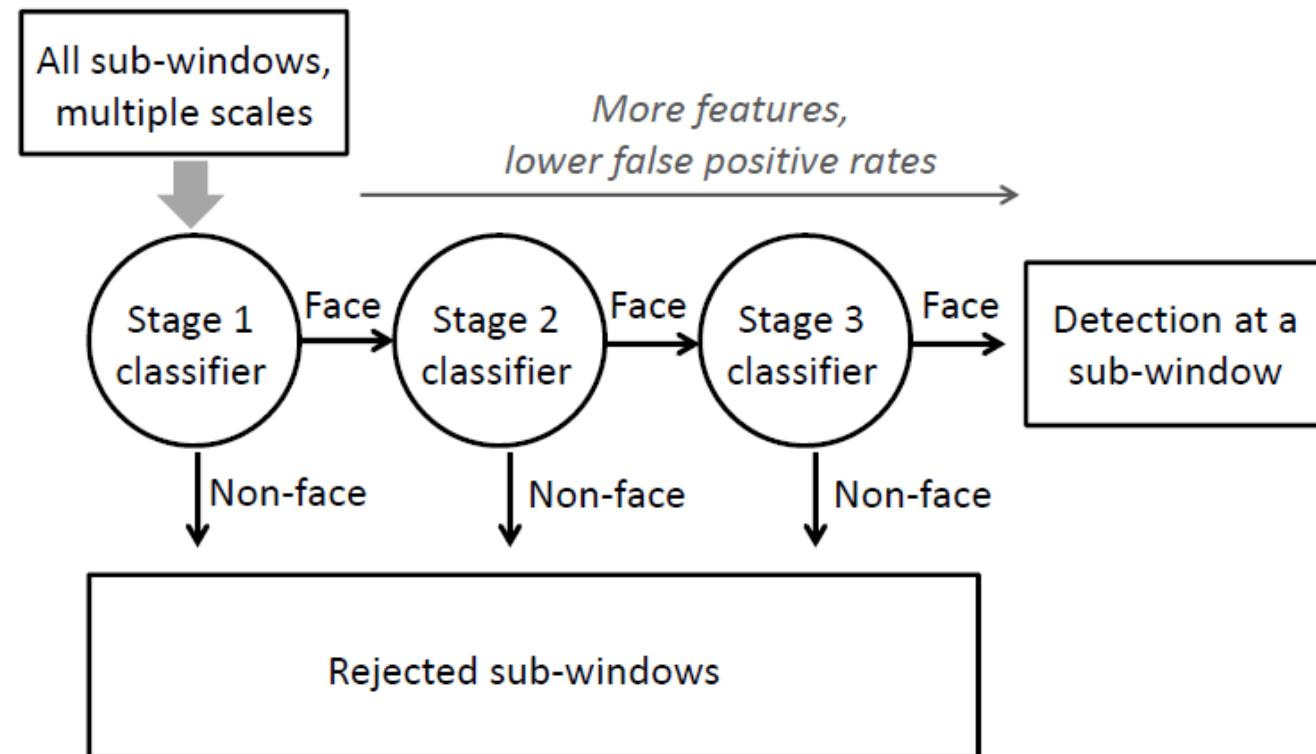
Viola-Jones Face Detector: Kết quả



Hai đặc trưng được lựa chọn đầu tiên

Mô hình phân tầng

- Sử dụng các bộ phân loại ít chính xác nhưng nhanh ở đầu để nhanh chóng loại bỏ các cửa sổ rõ ràng không có mặt



Huấn luyện mô hình phân tầng

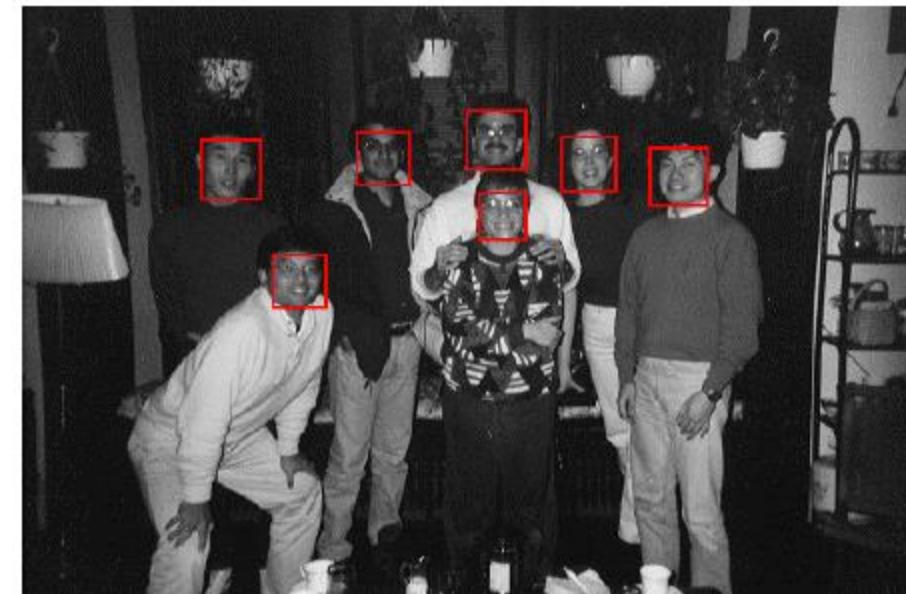
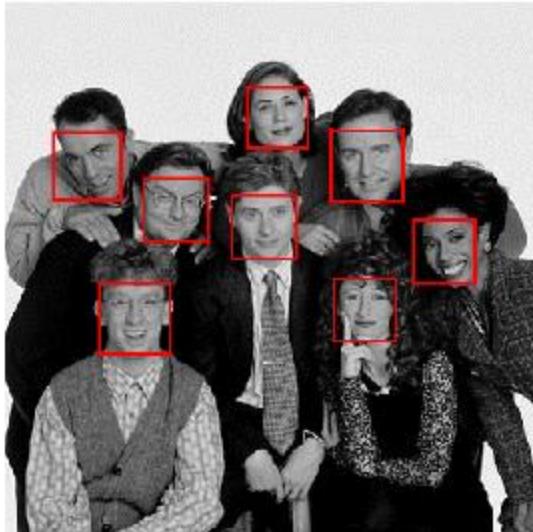
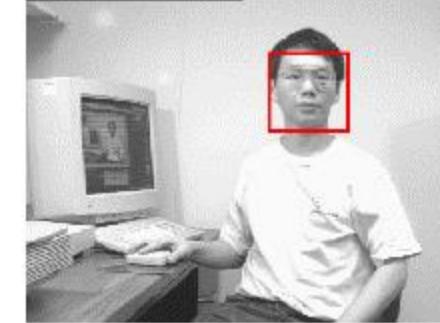
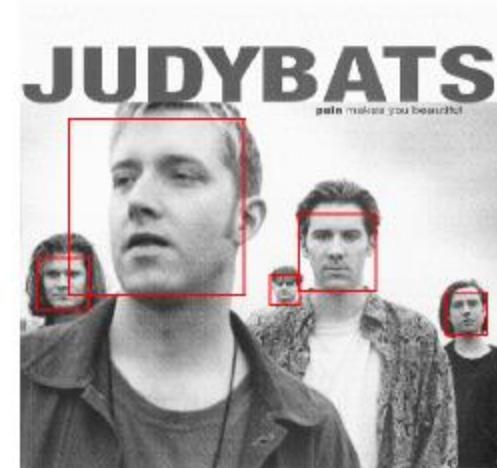
- Thiết lập độ chính xác phát hiện mặt và tỉ lệ nhận nhầm (false positive rate) cho mỗi tầng
- Thêm đặc trưng cho tới khi tầng hiện tại đạt được tỉ lệ mong muốn ở trên
- Nếu tỉ lệ nhận nhầm cả mô hình không đủ thấp thì tiếp tục thêm tầng mới
- Sử dụng các mẫu nhận nhầm (false positives) của tầng này làm dữ liệu âm tính (negative samples) cho tầng sau

Viola-Jones Face Detector: Kết quả

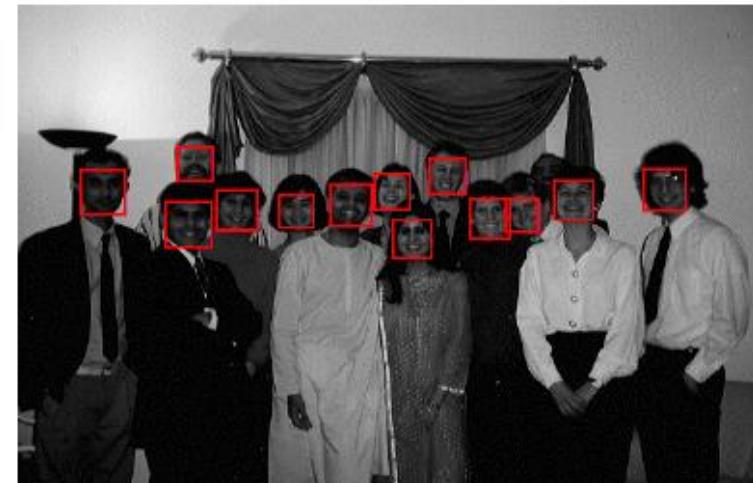
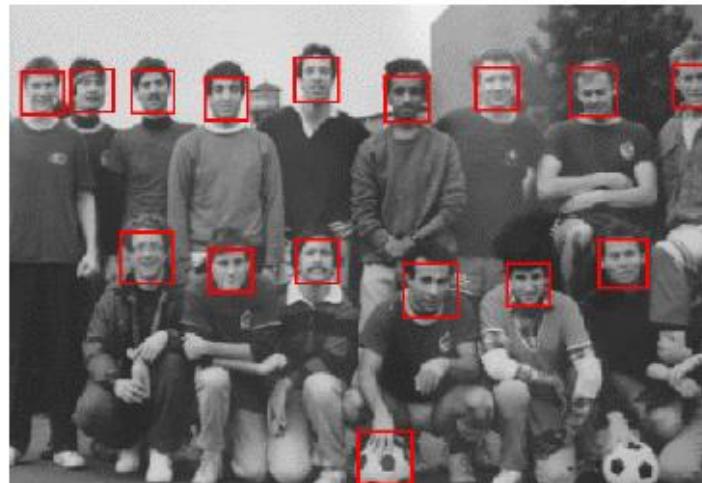
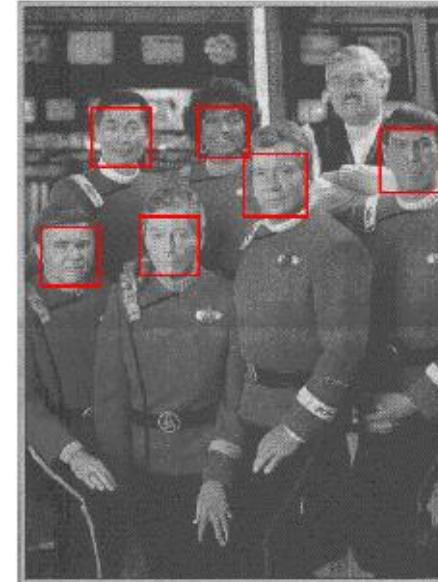
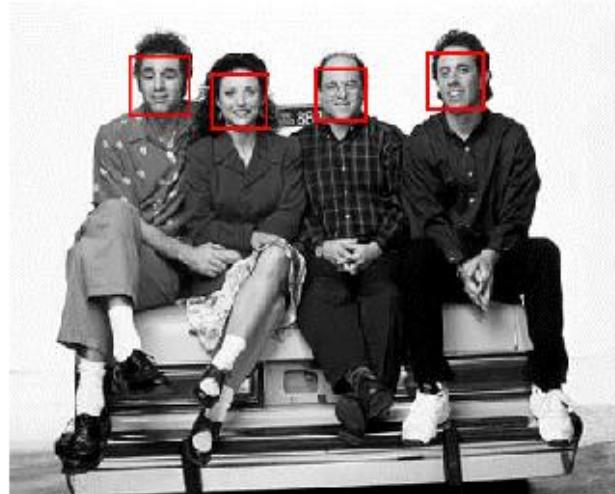


VINBIGDATA
VINGROUP

vdS AI Academy
Vietnam



Viola-Jones Face Detector: Kết quả



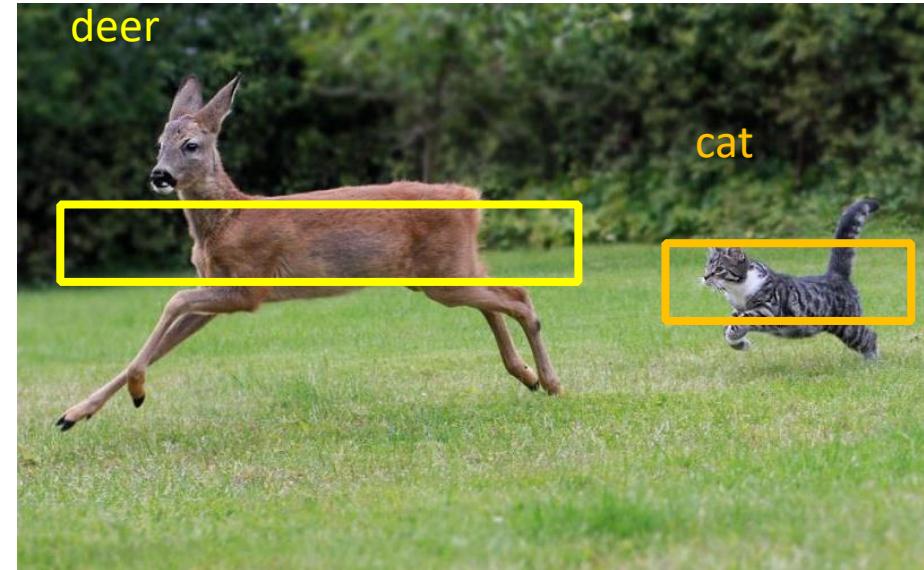
Viola-Jones Face Detector: Kết quả





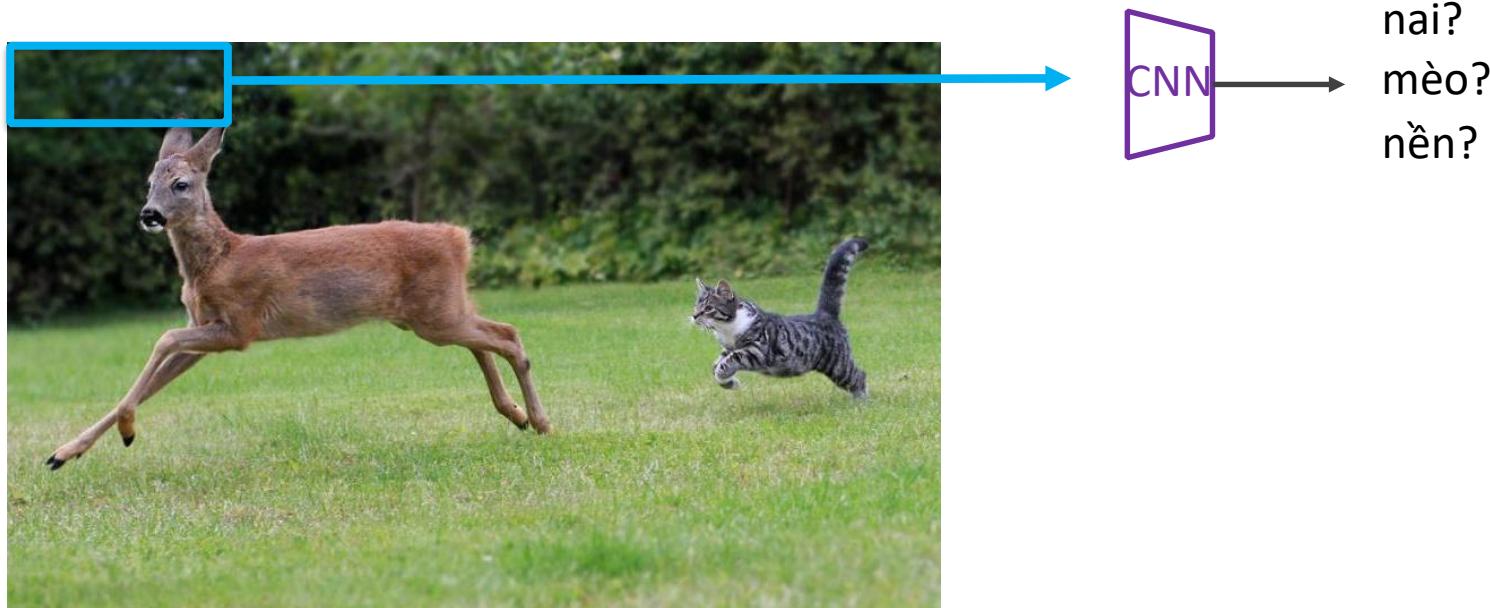
Kỹ thuật học sâu

Phương pháp cửa sổ trượt (sliding windows)



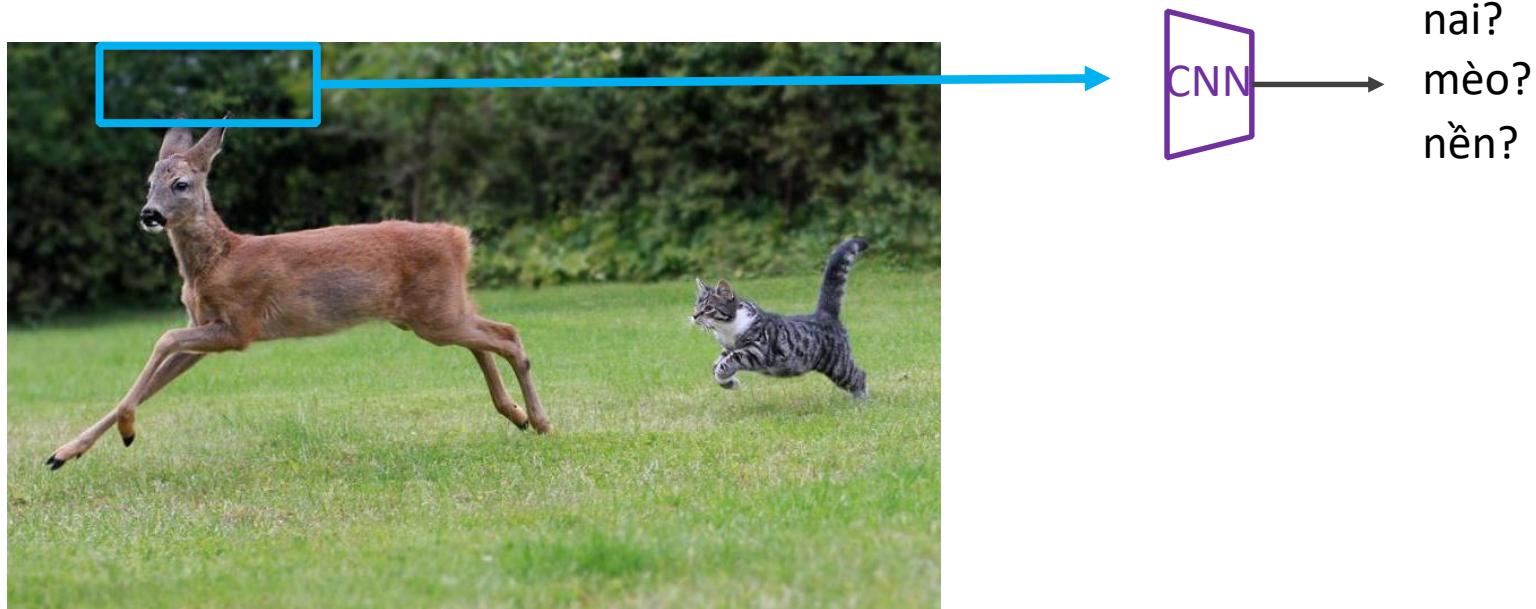
Phương pháp cửa sổ trượt (sliding windows)

- Quét cửa sổ từ trái sang phải, từ trên xuống dưới. Tại mỗi vị trí thực hiện bài toán phân loại vùng cửa sổ hiện tại thành nhiều lớp đối tượng cộng thêm lớp nền.



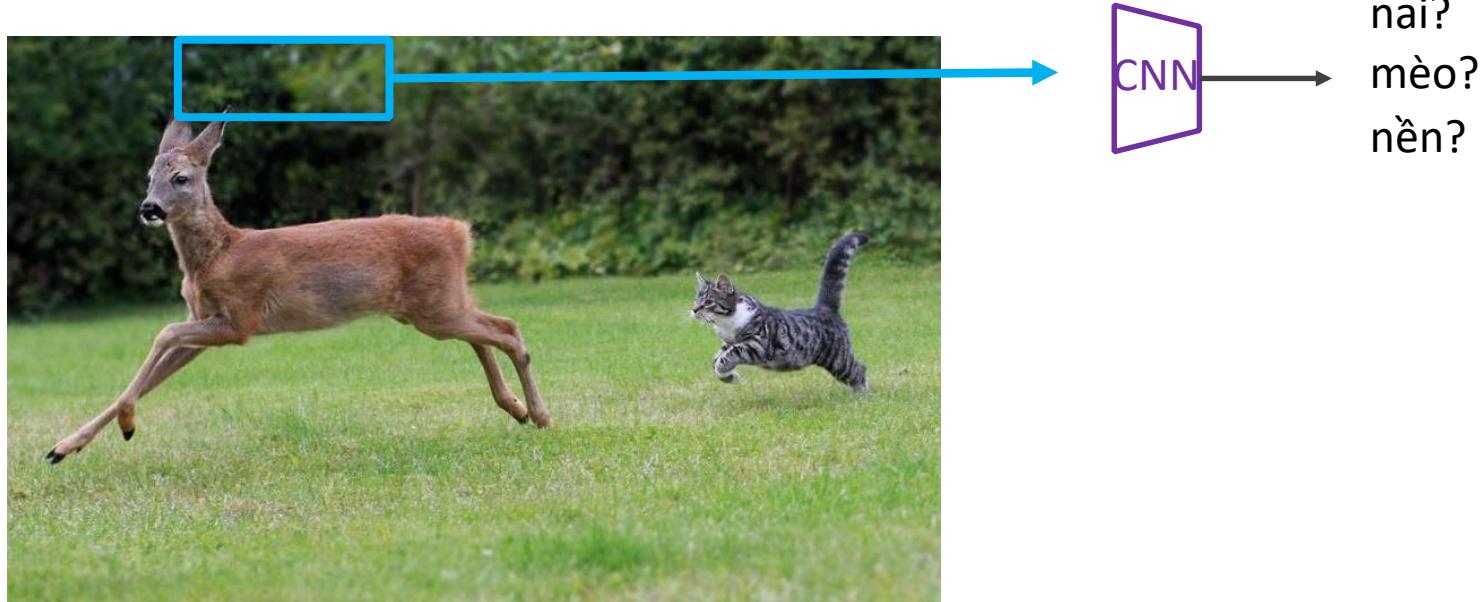
Phương pháp cửa sổ trượt (sliding windows)

- Quét cửa sổ từ trái sang phải, từ trên xuống dưới. Tại mỗi vị trí thực hiện bài toán phân loại vùng cửa sổ hiện tại thành nhiều lớp đối tượng cộng thêm lớp nền.



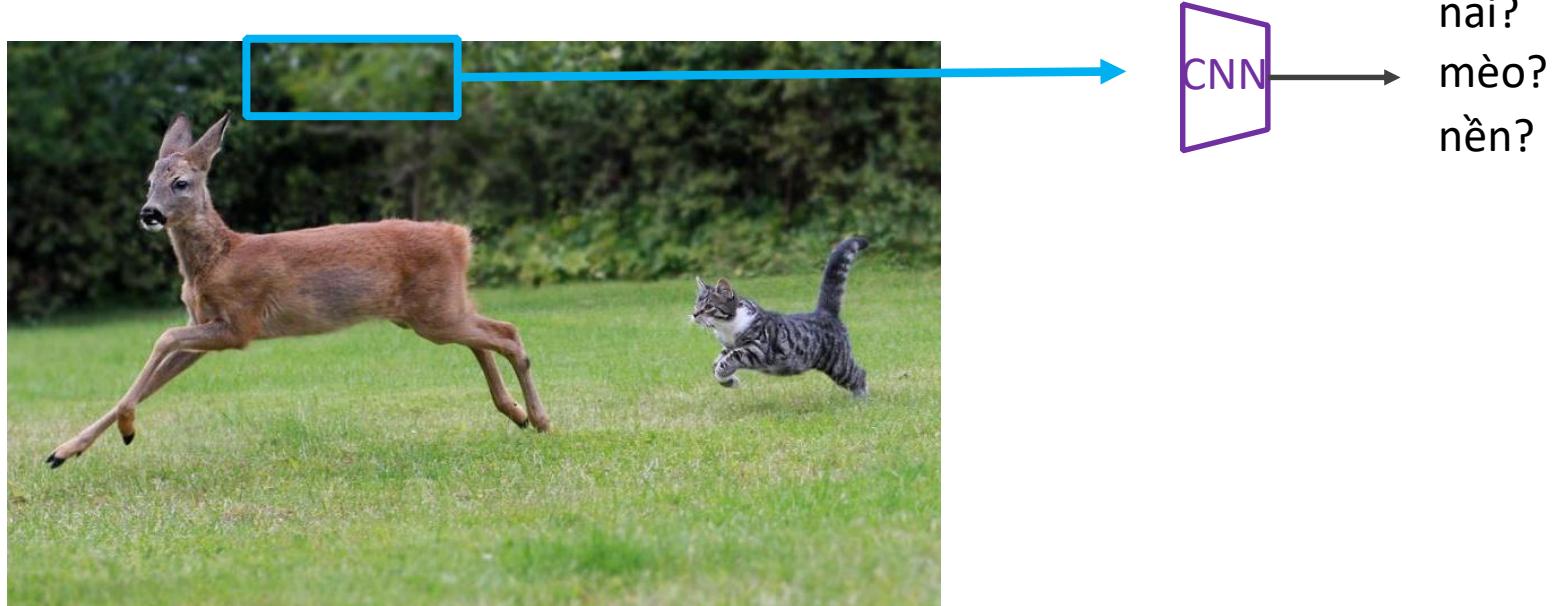
Phương pháp cửa sổ trượt (sliding windows)

- Quét cửa sổ từ trái sang phải, từ trên xuống dưới. Tại mỗi vị trí thực hiện bài toán phân loại vùng cửa sổ hiện tại thành nhiều lớp đối tượng cộng thêm lớp nền.



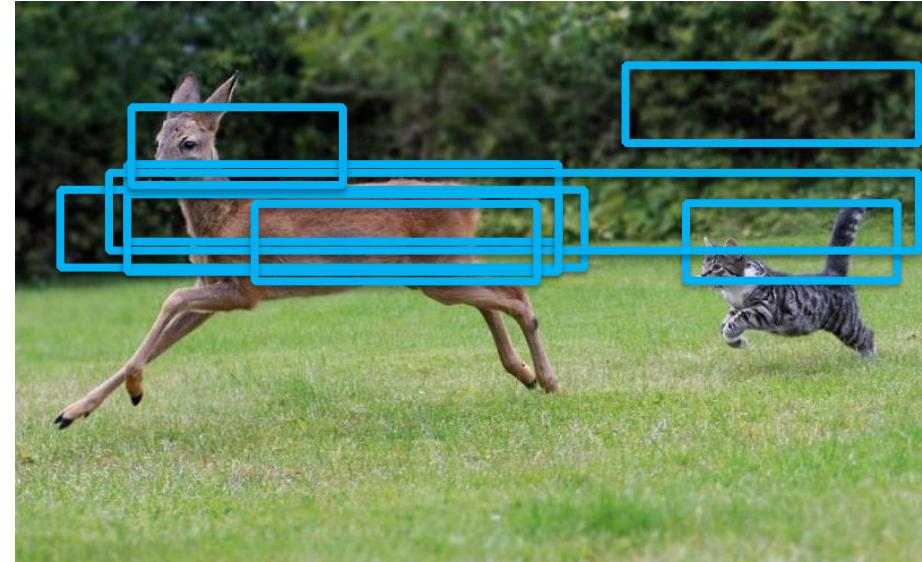
Phương pháp cửa sổ trượt (sliding windows)

- Quét cửa sổ từ trái sang phải, từ trên xuống dưới. Tại mỗi vị trí thực hiện bài toán phân loại vùng cửa sổ hiện tại thành nhiều lớp đối tượng cộng thêm lớp nền.



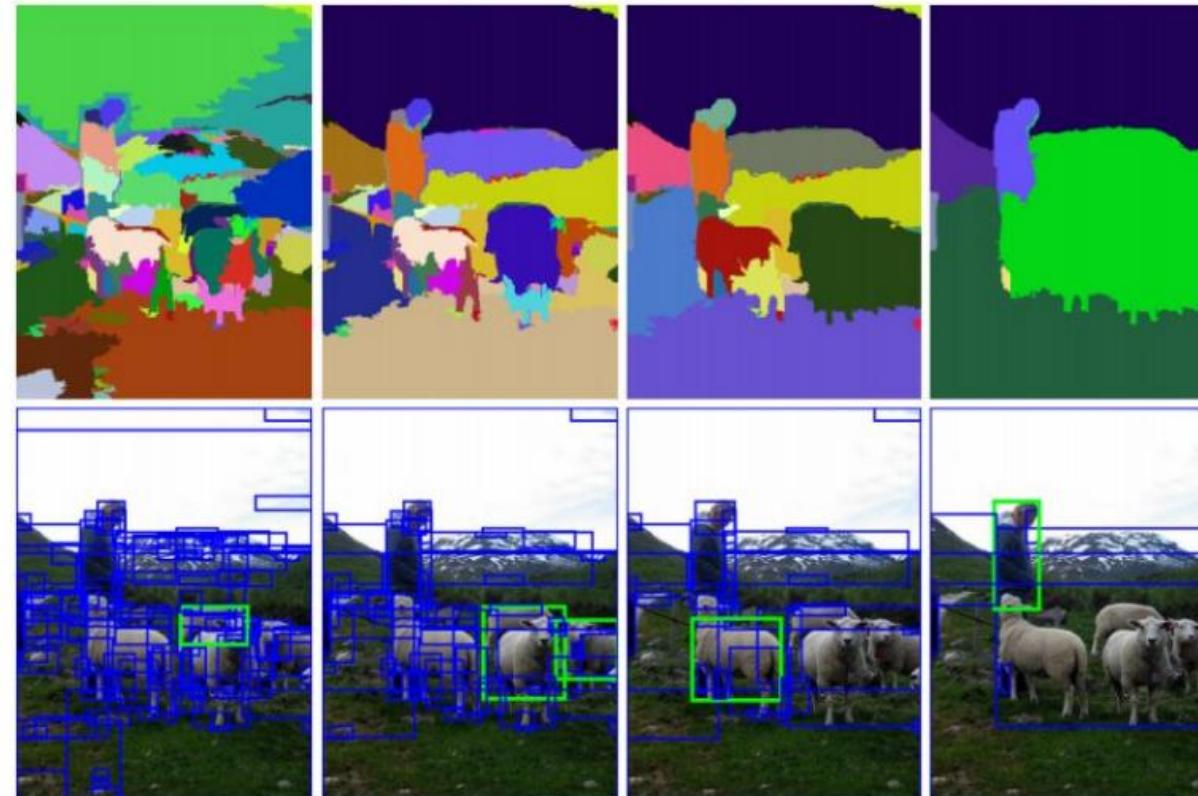
Phương pháp dựa trên đề xuất vùng

- Thay vì quét tất cả vị trí (số lượng rất lớn!), chỉ phân tích để đề xuất ra một số vùng (box) có khả năng cao chứa đối tượng
- Các phương pháp này có hai giai đoạn (two-stage):
 - 1) đề xuất vùng
 - 2) xử lý từng vùng để phân loại và hiệu chỉnh tọa độ box



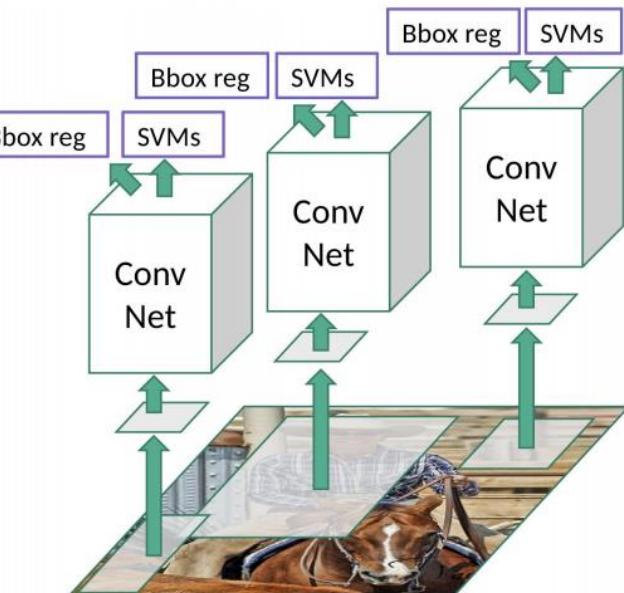
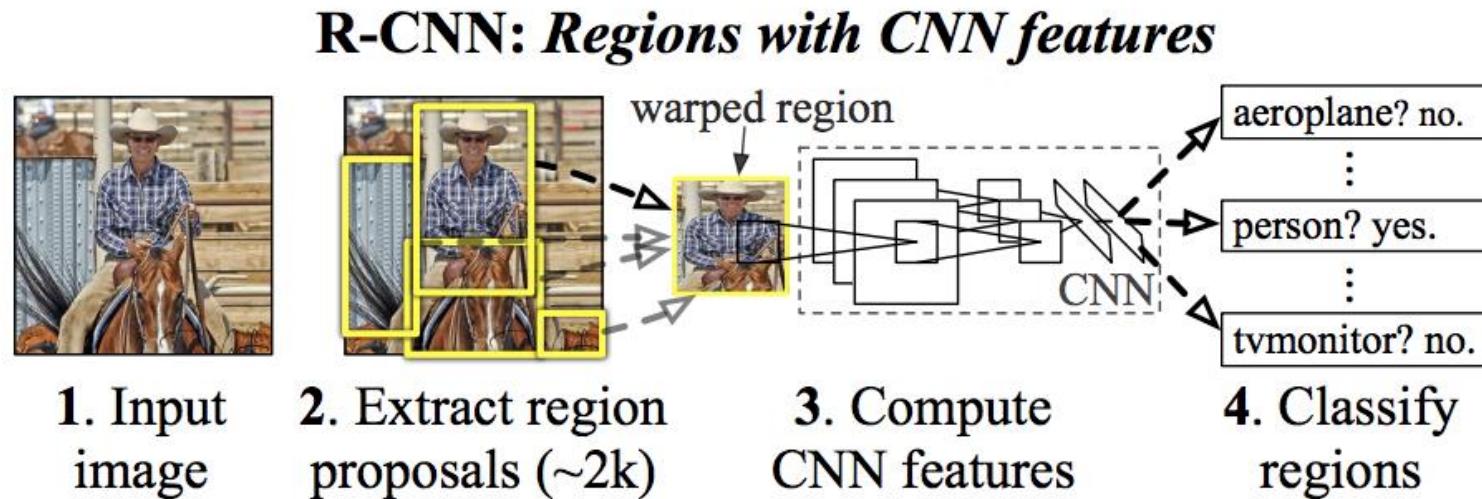
SS: Selective Search

- Segmentation As Selective Search for Object Recognition. van de Sande et al. ICCV 2011



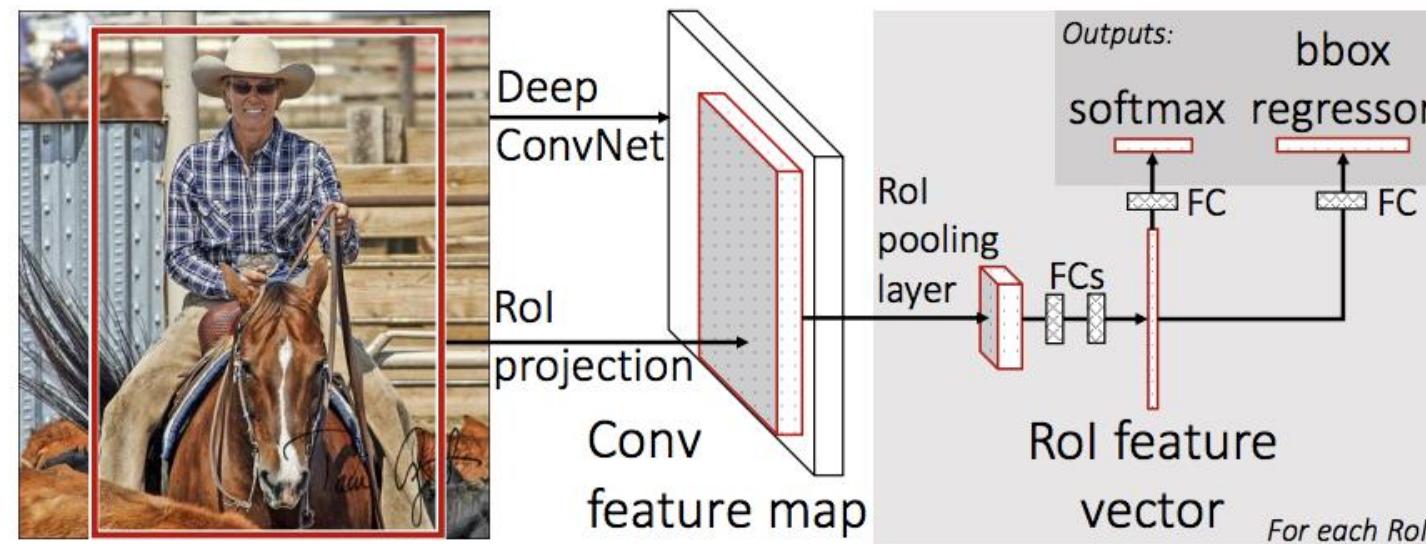
R-CNN (Region-based ConvNet)

- Đề xuất một số vùng tiềm năng bằng thuật toán thô khác, chẳng hạn selective search
- Dùng mạng CNN trích xuất đặc trưng từng vùng rồi phân loại bằng SVM



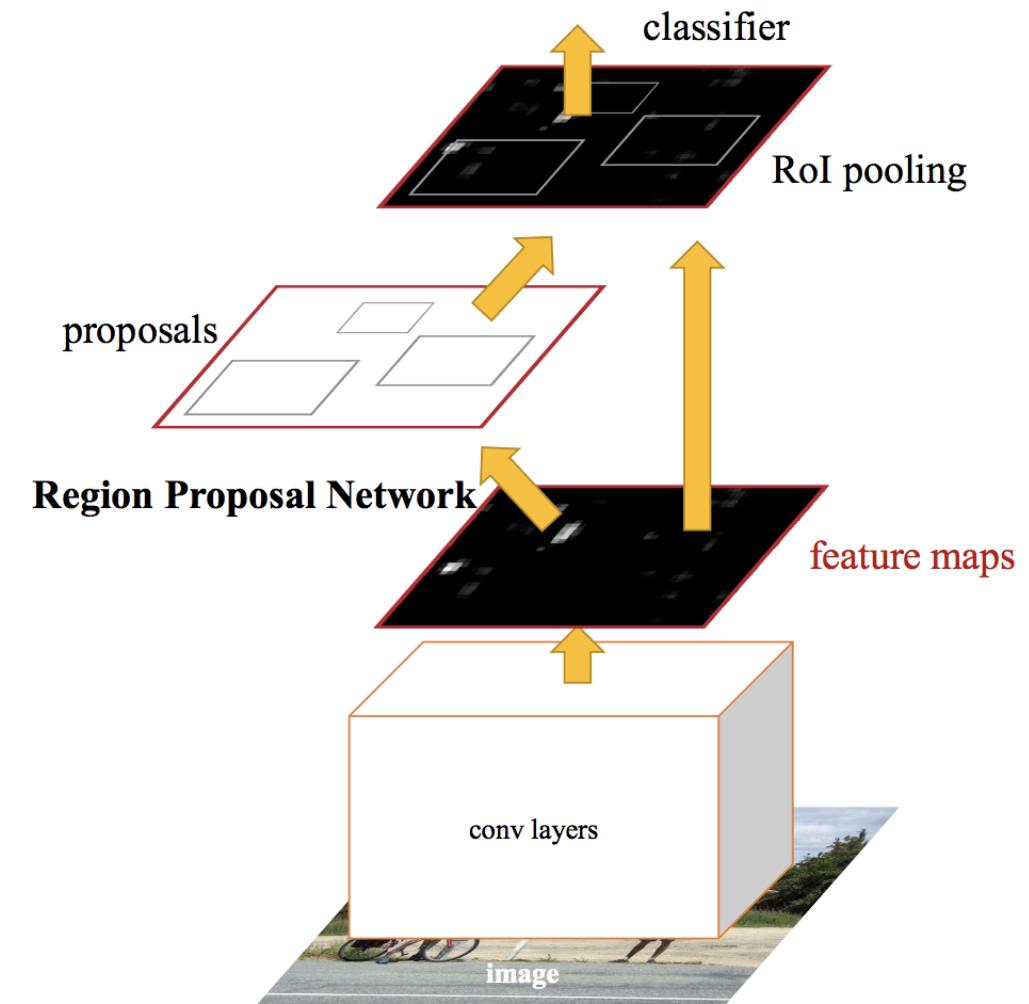
Fast-RCNN

- Đẩy tất cả các vùng (khoảng 2000) qua mạng trích xuất CNN cùng một lúc
- Crop thông tin ở lớp đầu ra của CNN thay vì crop vùng trên ảnh gốc như R-CNN
- Đẩy qua nhánh phân loại và nhánh hiệu chỉnh tọa độ box



Faster-RCNN

- Dùng một mạng riêng để đề xuất vùng thay cho selective search
- Còn gọi là phương pháp phát hiện đối tượng hai giai đoạn (two-stage object detector)



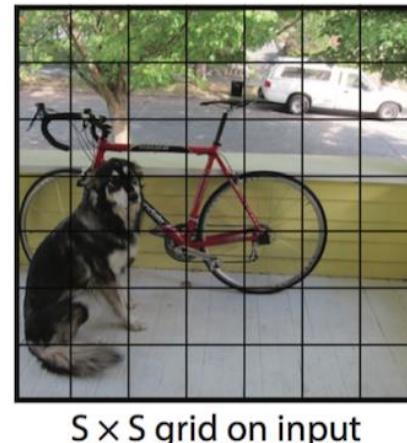
Đặc điểm các mạng không đề xuất vùng

- Còn gọi là mạng một giai đoạn (one-stage)
- Các mạng này thường đề xuất một lưới box dày đặc trên ảnh ban đầu, thường có bước nhảy đều (stride)
- Từng box này sẽ được phân loại và hiệu chỉnh tọa độ (nếu box chứa đối tượng) bằng mạng CNN
- Các mạng một giai đoạn thường nhanh hơn và đơn giản hơn các mạng hai giai đoạn, nhưng độ chính xác có thể không cao bằng.

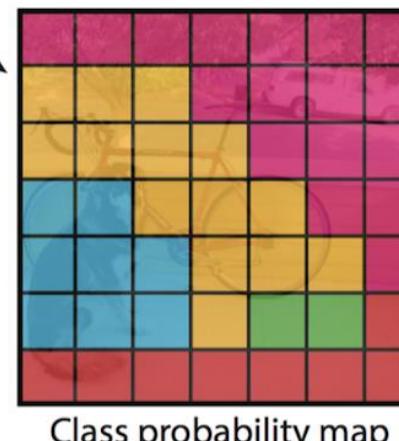
YOLO- You Only Look Once

$S \times S \times B$ bounding boxes

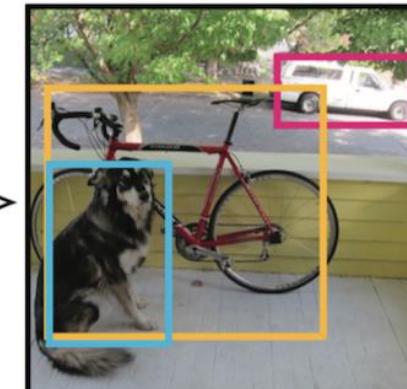
confidence = $Pr(\text{object}) \times \text{IoU}(\text{pred, truth})$



Bounding boxes + confidence



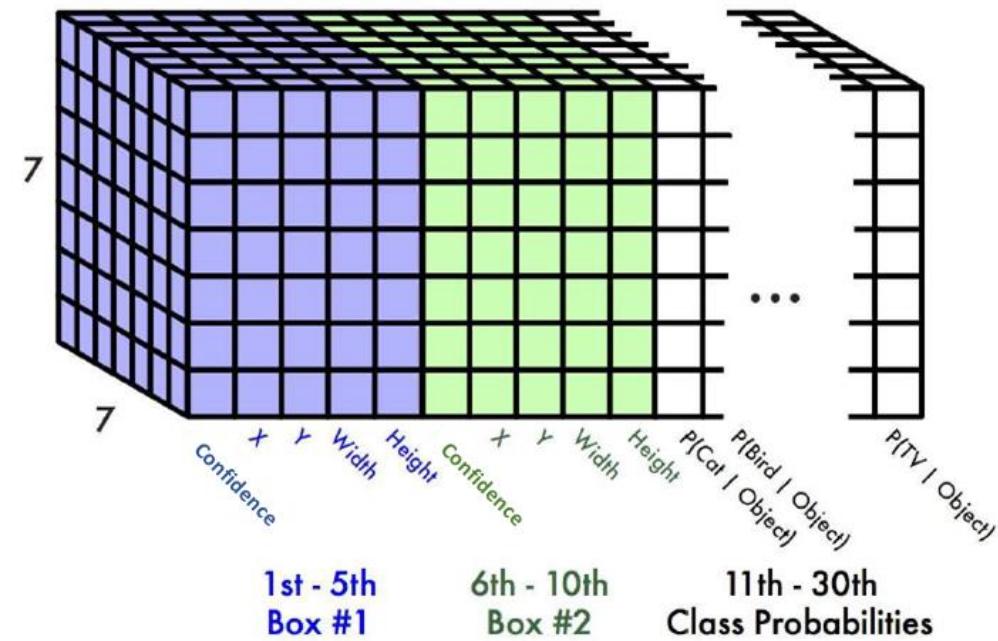
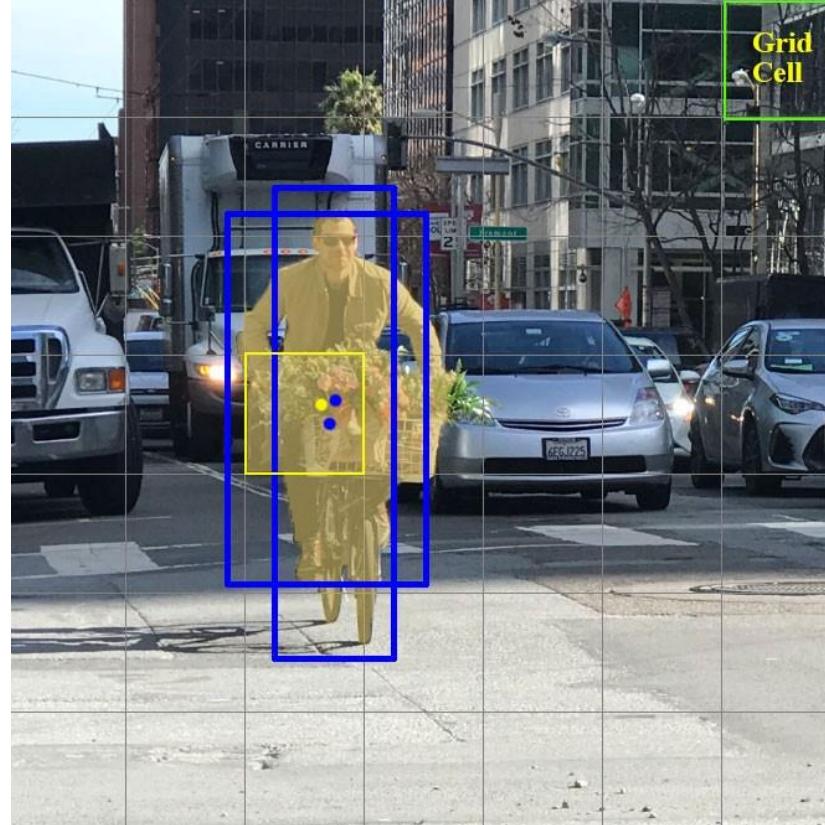
Class probability map



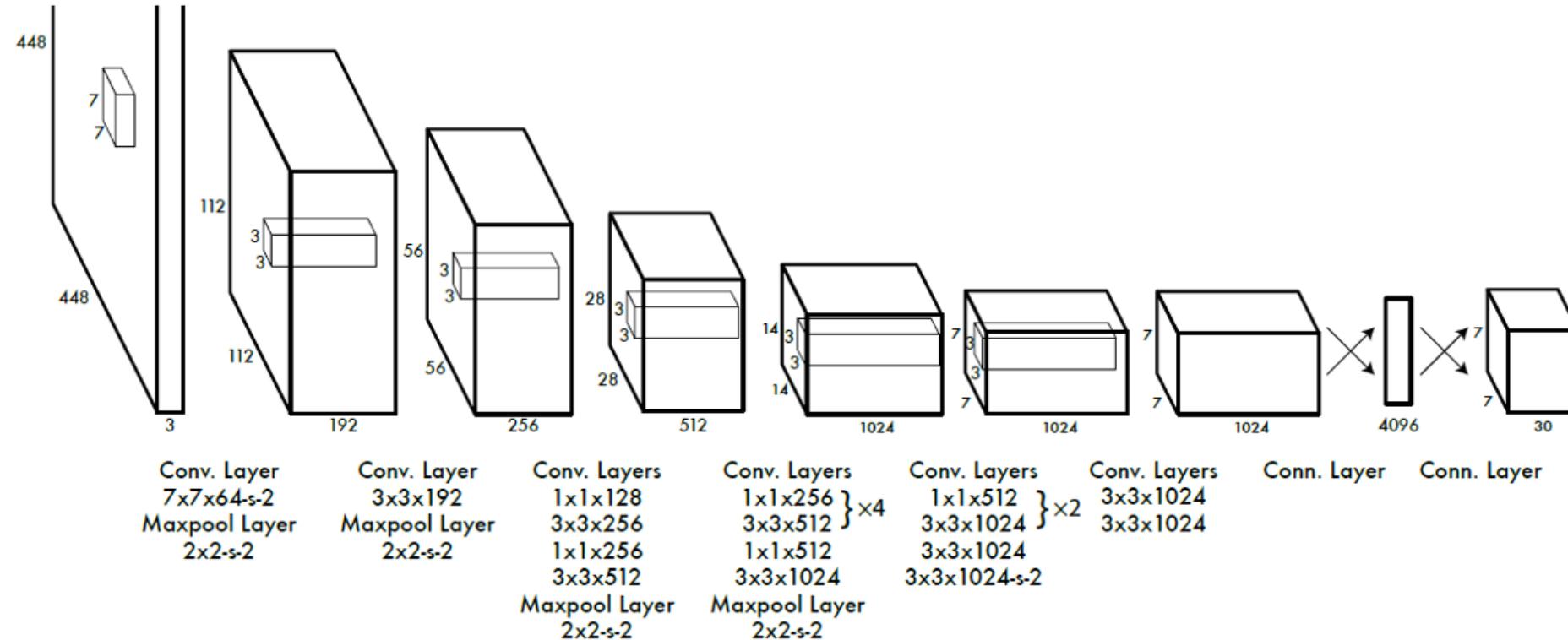
Final detections

$Pr(\text{Class}_i | \text{object})$

YOLO- You Only Look Once



YOLO- You Only Look Once



YOLO- You Only Look Once

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

1 when there is object, 0 when there is no object

Bounding Box Location (x, y) when there is object

Bounding Box size (w, h) when there is object

Confidence when there is object

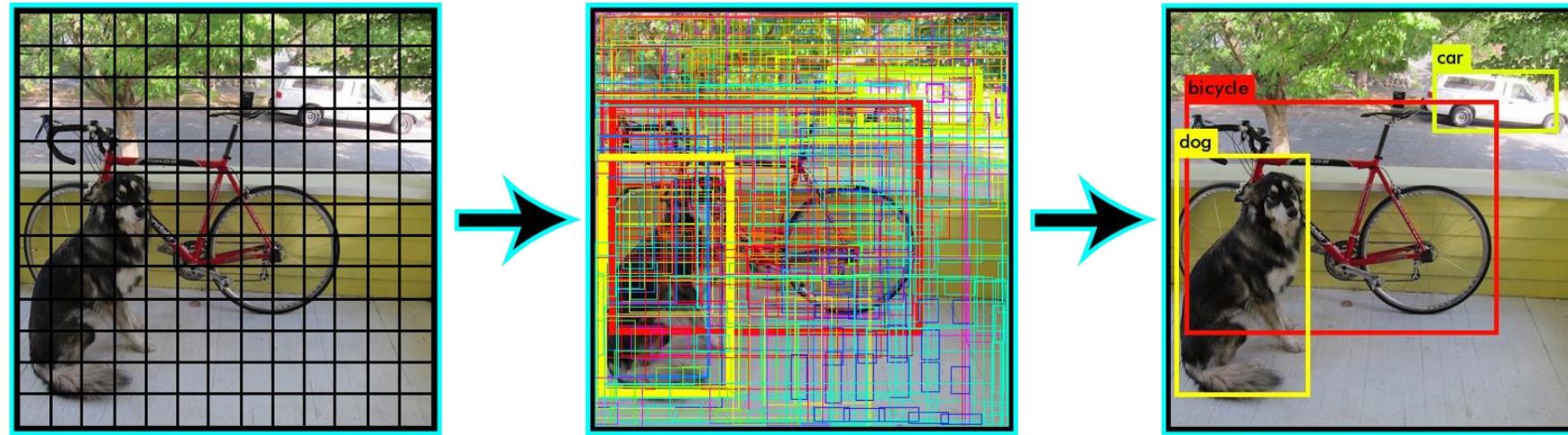
1 when there is no object, 0 when there is object

Confidence when there is no object

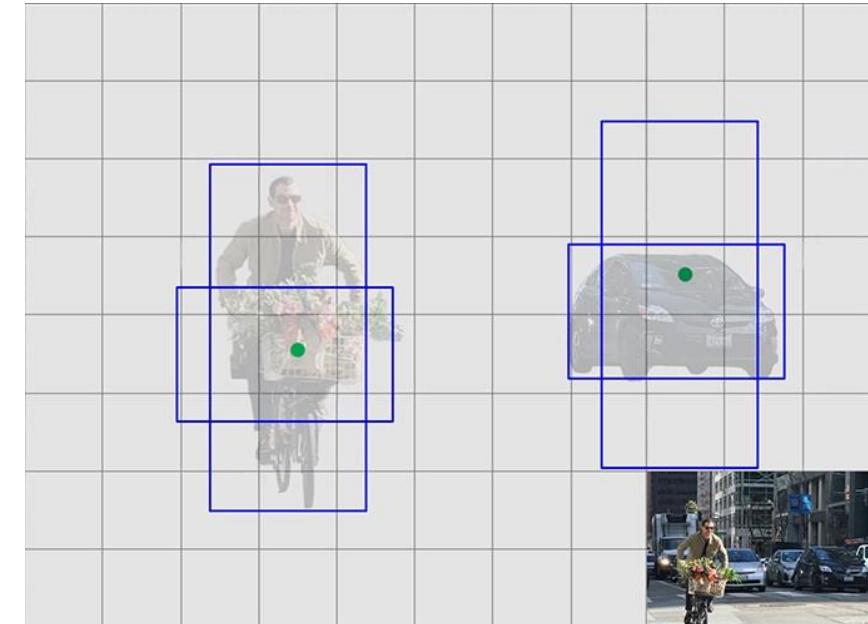
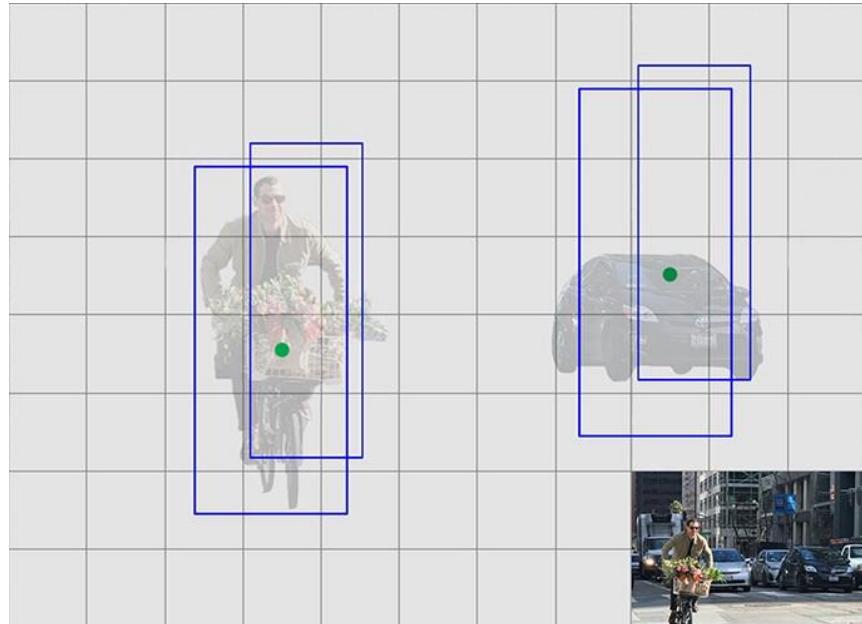
Class probabilities when there is object

YOLO- You Only Look Once

- Non-maximal suppression: gom các box lại để đưa ra kết quả cuối cùng



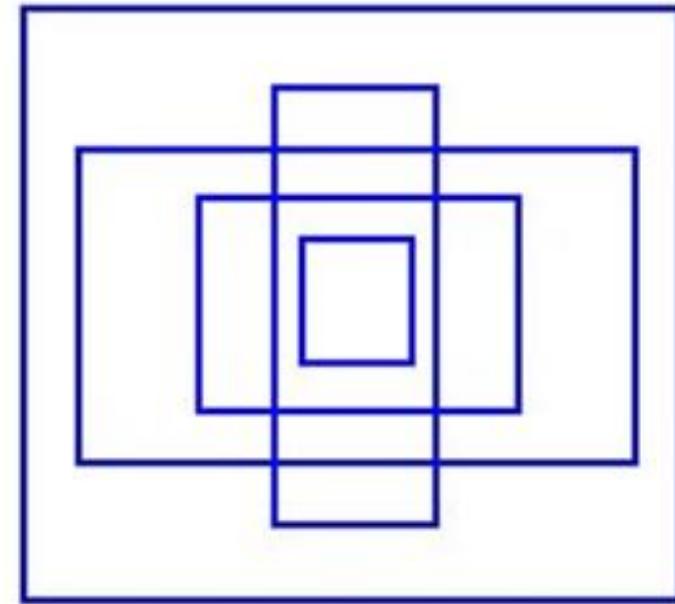
YOLO v2



YOLO v2

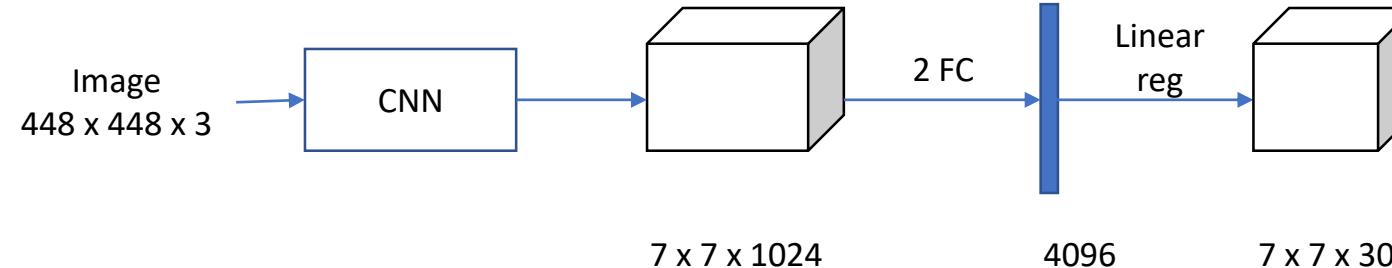
- Mỗi ô có 5 anchor box. Với mỗi anchor mạng sẽ đưa ra các thông tin:
 - offset của box: 4 số thực trong khoảng [0, 1]
 - Độ tin tưởng box đó có khả năng chứa đối tượng (objectness score).
 - Phân bố xác suất của đối tượng trong box đó ứng với các lớp đối tượng khác nhau (class scores).
- Tổng cộng mỗi ô có số đầu ra là:

$$5 * (4 + 1 + 20) = 125$$
 số thực

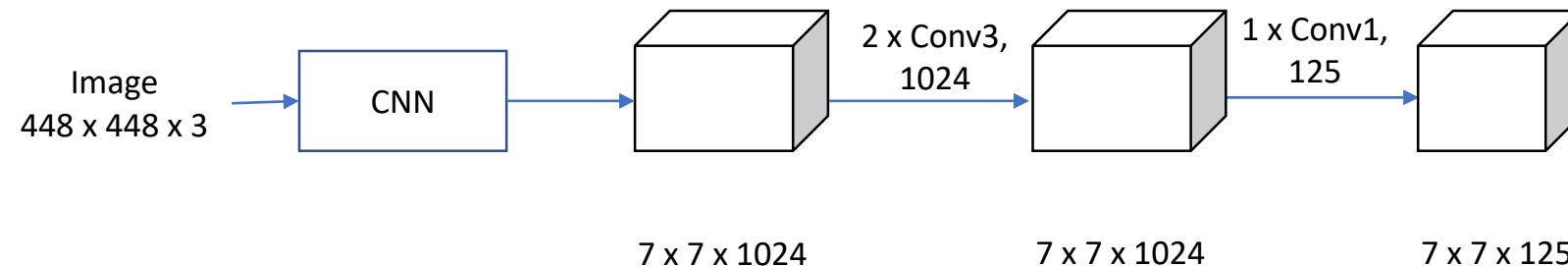


5 anchor box

YOLO v2



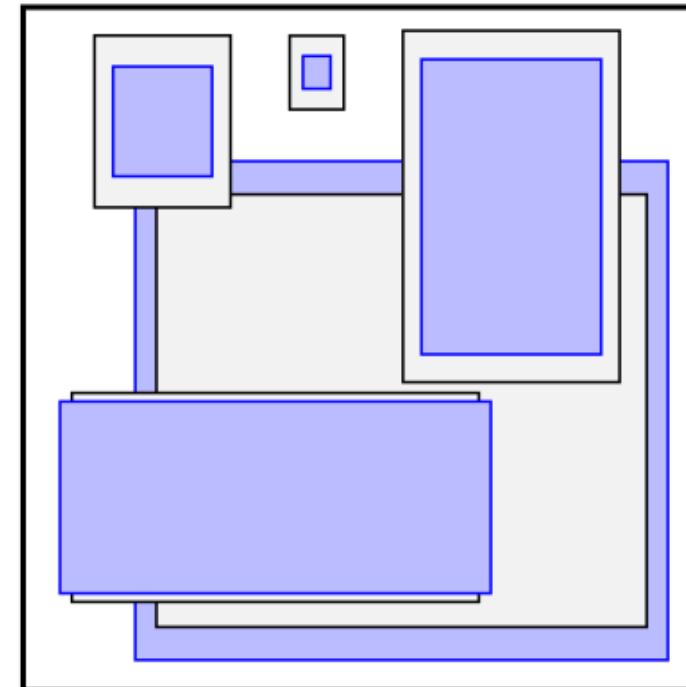
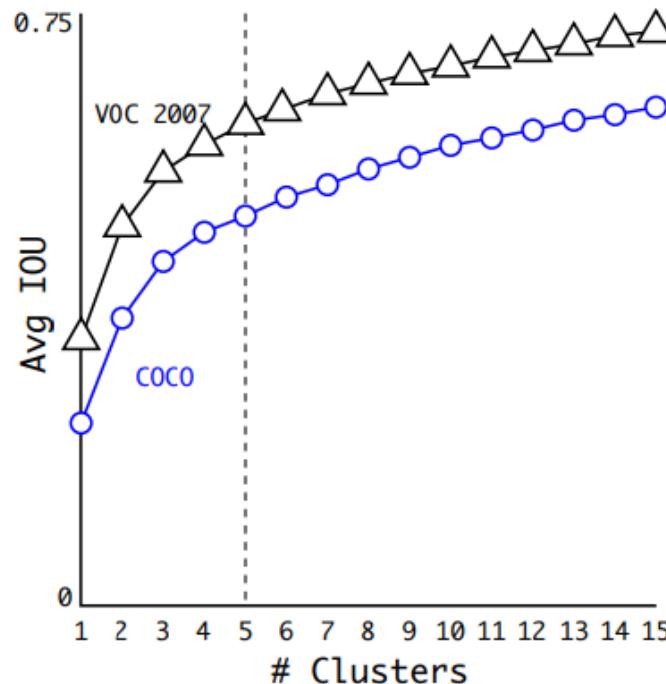
YOLO v1



YOLO v2

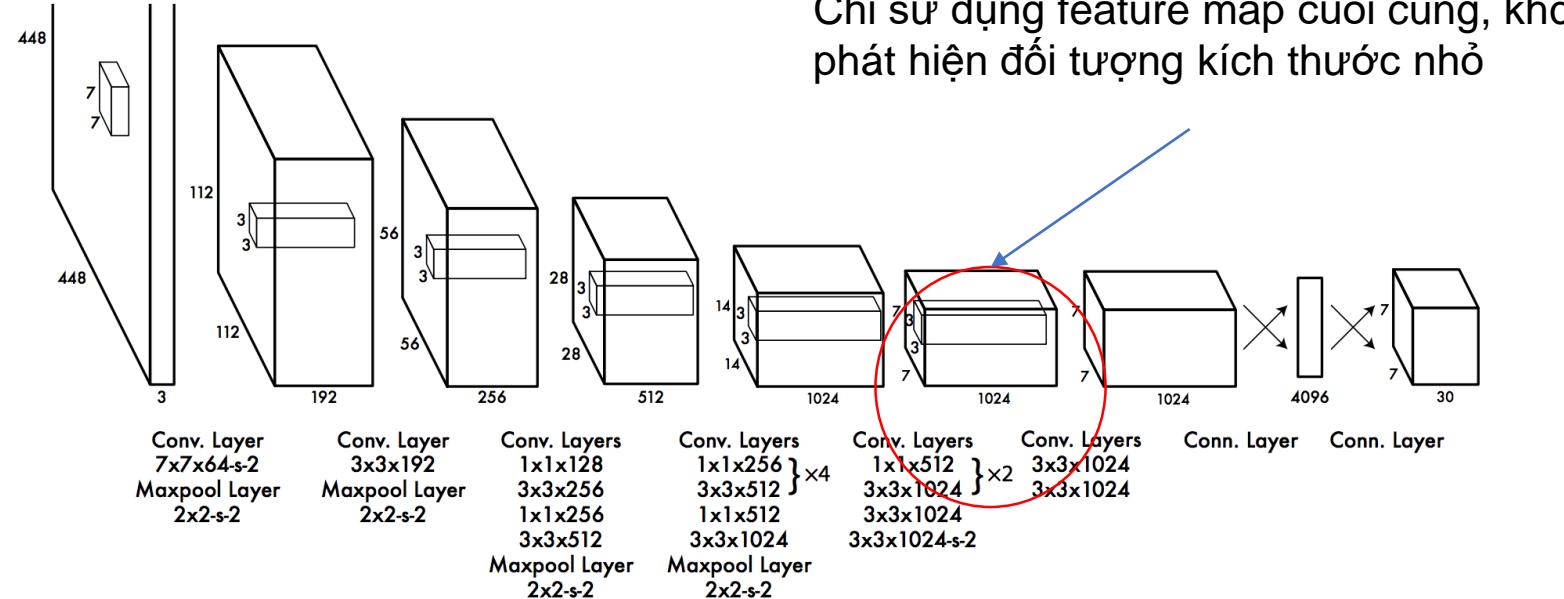
YOLO v2

- Xác định kích thước mặc định của các anchor bằng cách áp dụng k-means trên tập box các đối tượng đã được đánh nhãn trong tập huấn luyện

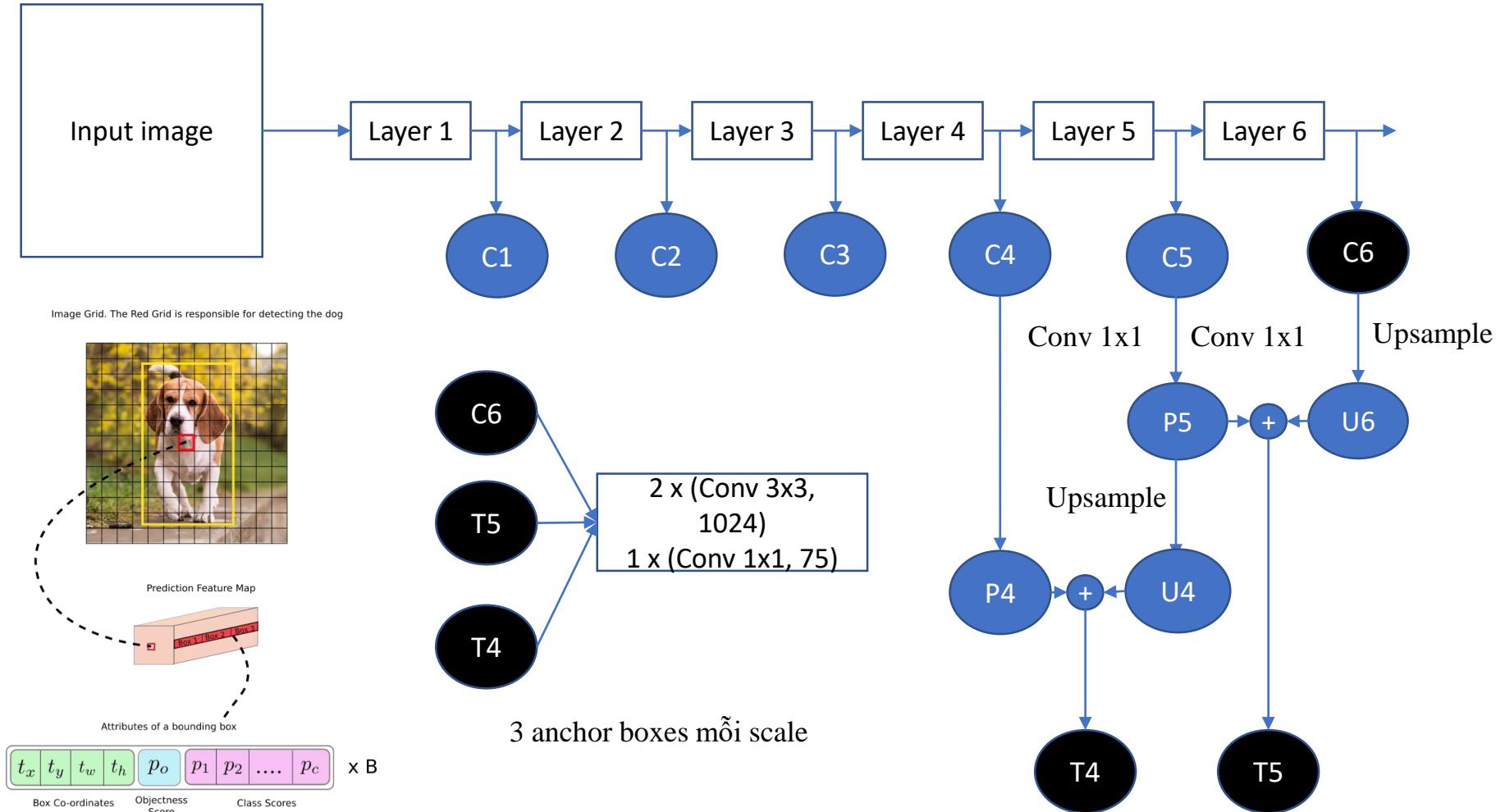


YOLO v2

- Nhược điểm của YOLO v1 và v2:

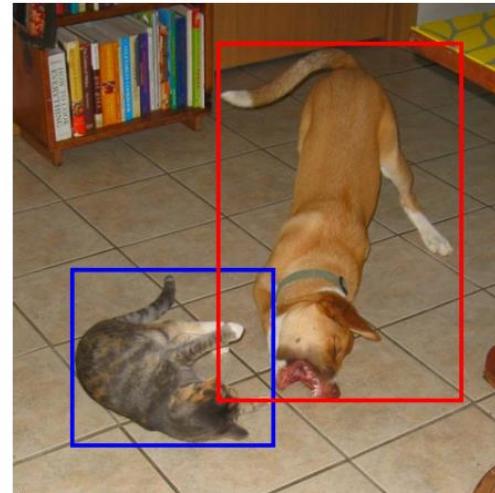


YOLO v3

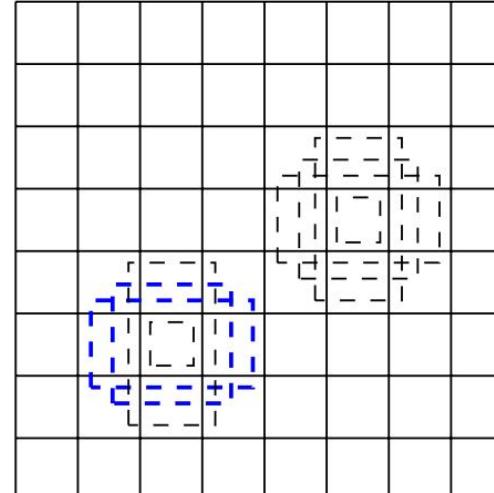


SSD: Single Shot Detector

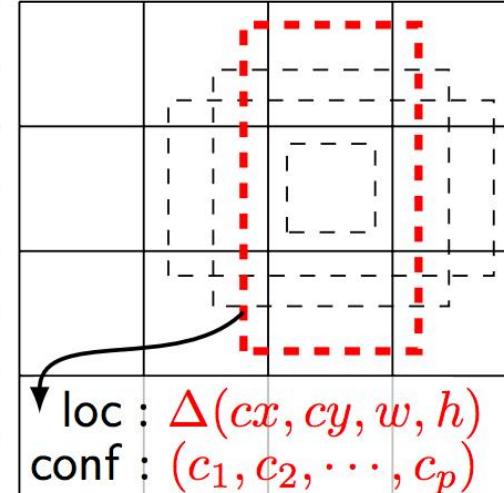
- Tương tự YOLO nhưng lưới box dày đặc hơn, có nhiều lưới với các kích thước box khác nhau
- Kiến trúc mạng backbone khác với YOLO
- Data augmentation + Hard negative mining



(a) Image with GT boxes



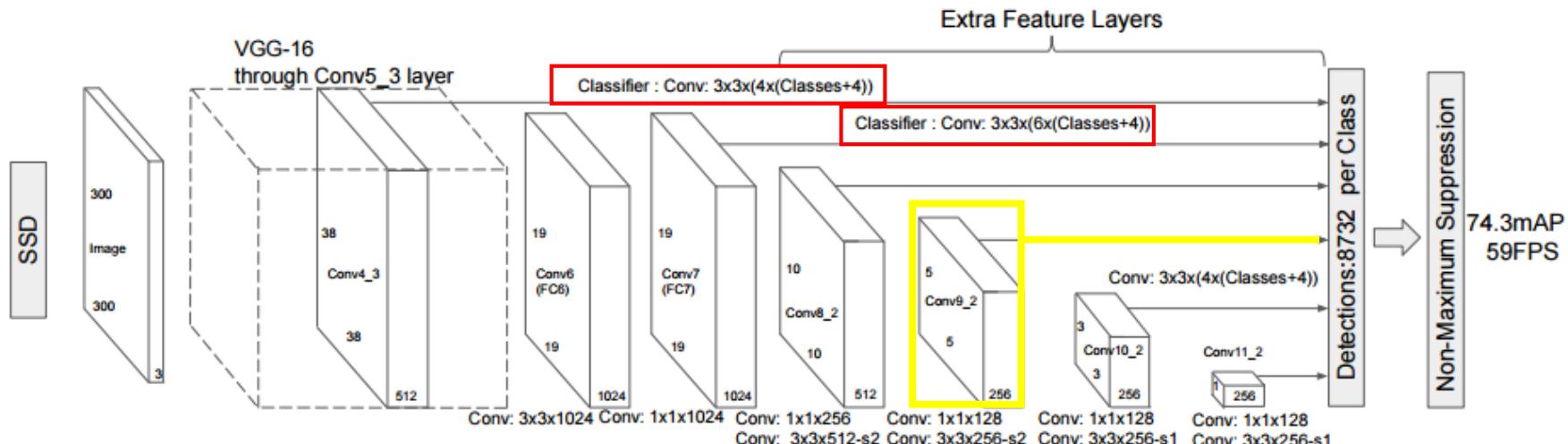
(b) 8×8 feature map



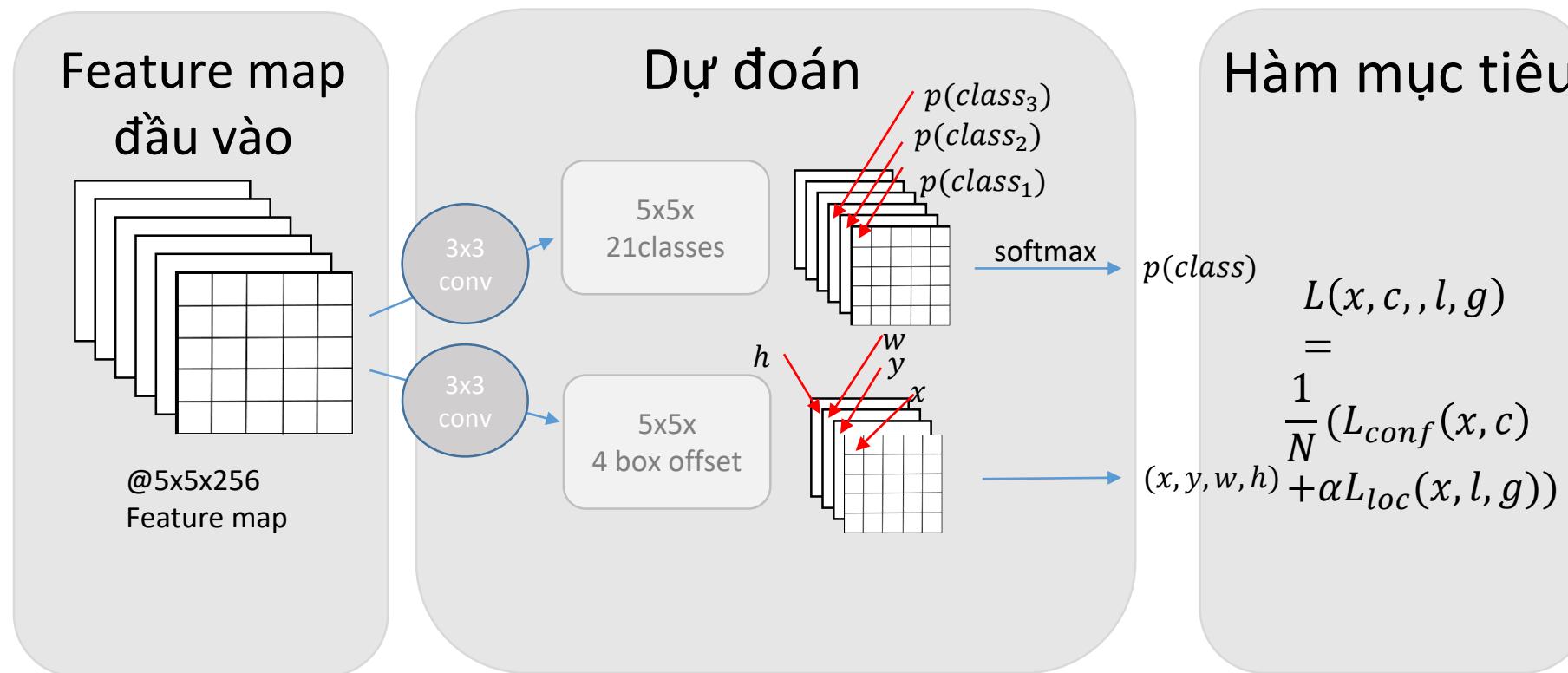
(c) 4×4 feature map

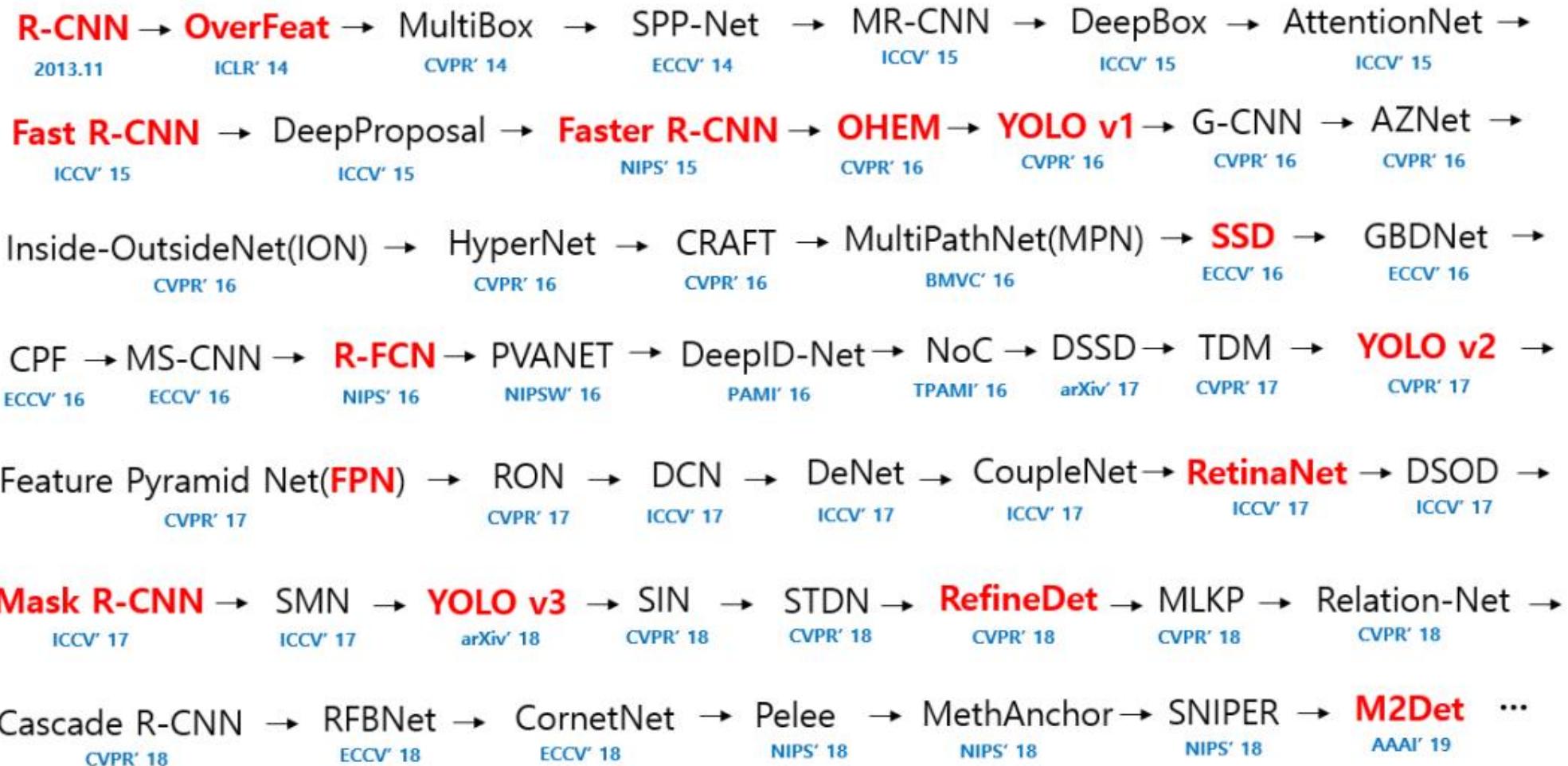
SSD: Single Shot Detector

- Mạng backbone: VGG-16
- Thêm các lớp tích chập phụ phía sau các lớp của mạng backbone
- Phát hiện đối tượng ở nhiều mức khác nhau trong mạng (Multi-scale)



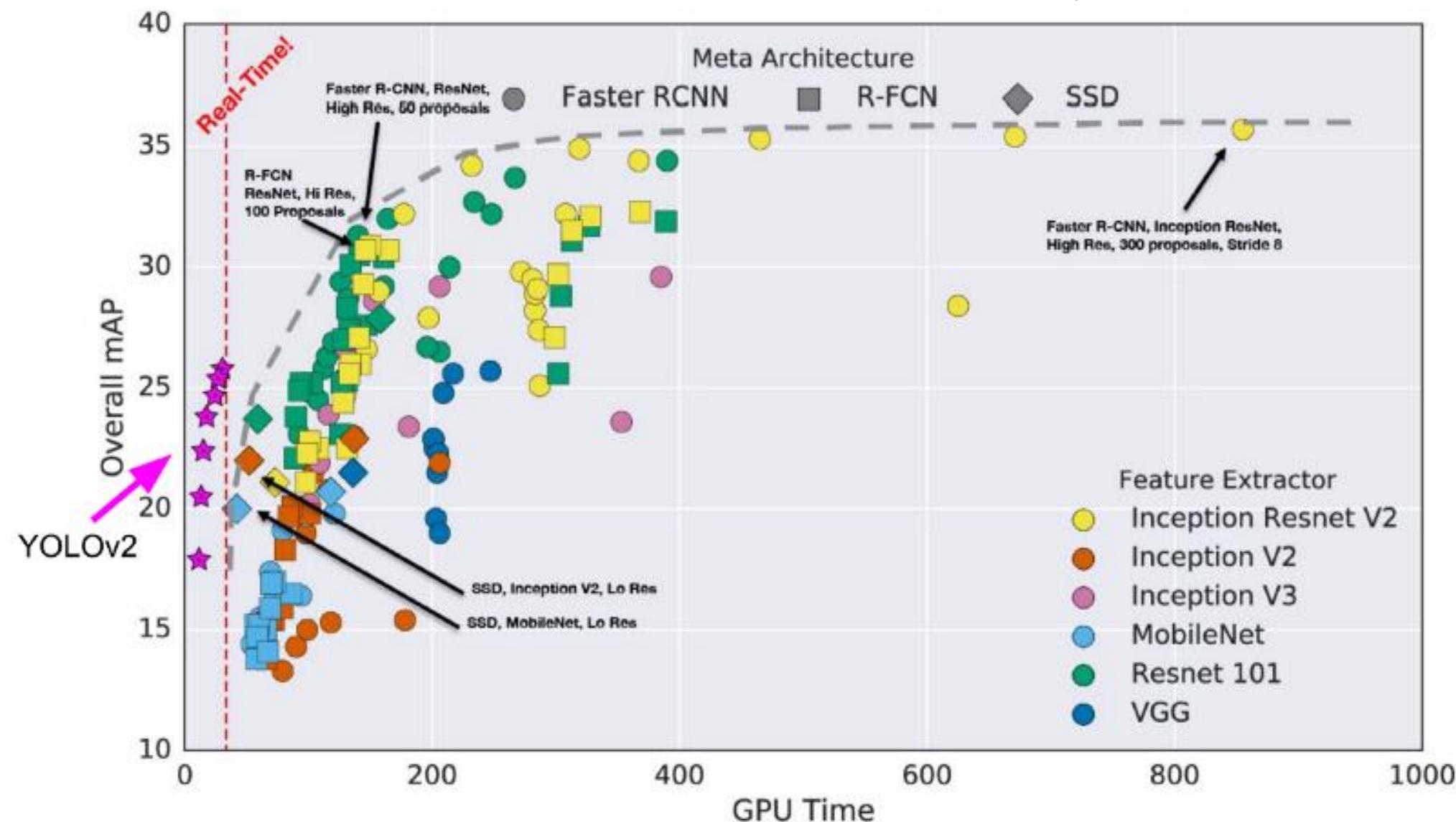
SSD: Single Shot Detector

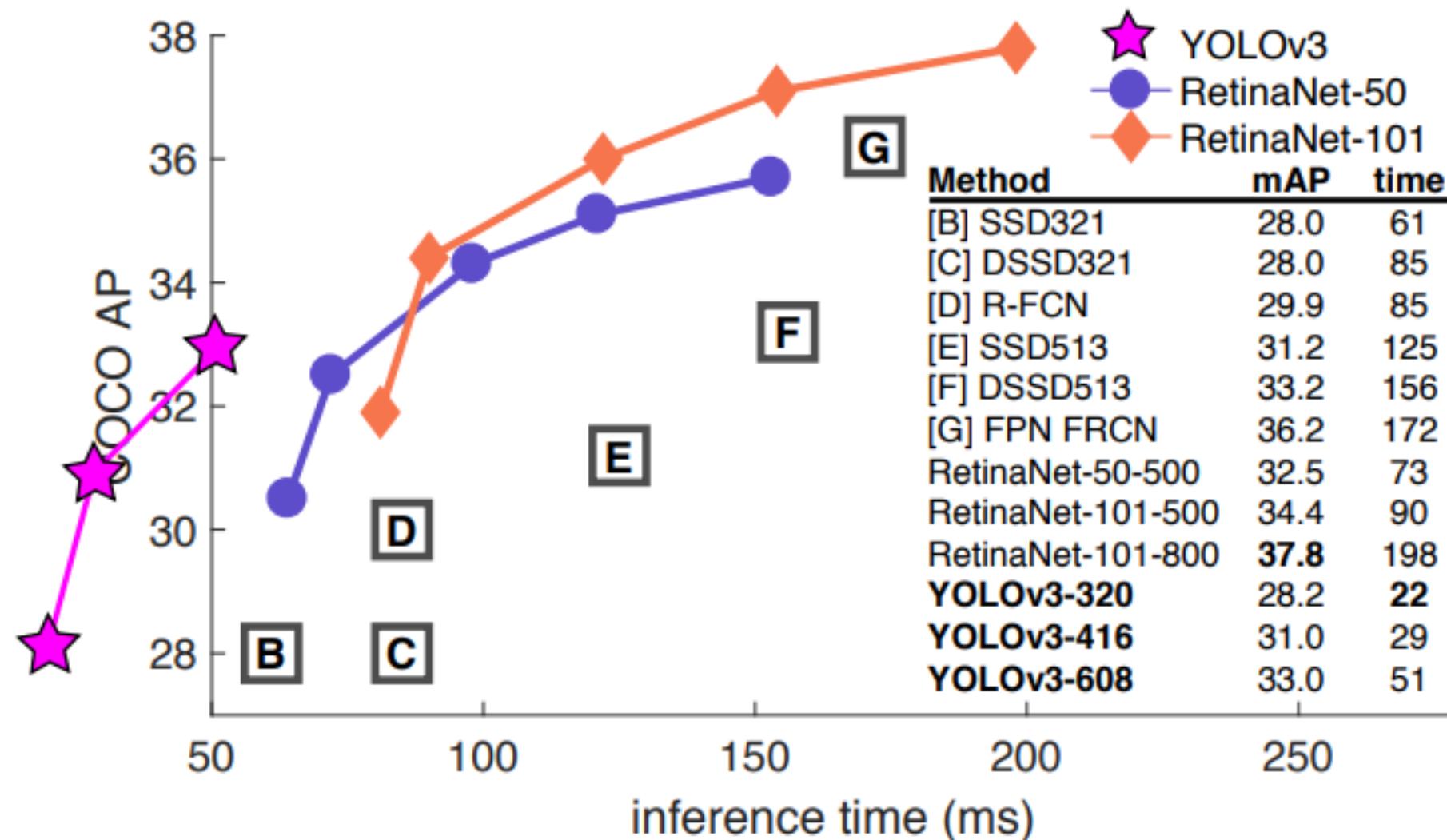




One-stage vs two-stage

<p>Faster and simpler <i>one-stage object detector</i> (dense sampling of object locations, scales, and aspect ratios)</p>			<p>More accurate <i>two-stage object detector</i> (proposal-driven mechanism)</p>
YOLO	YOLO-v2	YOLO-v3	R-CNN
SSD	DSSD		Fast R-CNN
MDCN	SqueezeNet		Faster R-CNN
RetinaNet	RedefineDet		Feature Pyramid Network (FPN)
CornetNet	CenterNet		Mask R-CNN
EfficientDet			







Q&A

Thank you!