

25 YEARS ANNIVERSARY

**SOKT**

The graphic consists of the number "25" in large, bold, white font. A curved banner arches over the top of the "2" containing the text "YEARS ANNIVERSARY". Below the "25" is the acronym "SOKT" in a large, bold, white, sans-serif font.

**ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

# Chương 6: Một số ứng dụng học sâu trong thị giác máy (Phần 1)

# Nội dung

1. Giới thiệu tổng quan về thị giác máy và các ứng dụng
2. Giới thiệu về bài toán phát hiện đối tượng
3. Giới thiệu một số mạng đề xuất vùng R-CNN, Fast R-CNN, Faster R-CNN...
4. Giới thiệu một số mạng không đề xuất vùng: SSD, Yolo ...

# Giới thiệu tổng quan về thị giác máy và các ứng dụng

**Physics**

**Biology**

**Psychology**

**Computer Science**

**Mathematics**

# Computer Vision

Machine learning

Information retrieval

Systems, architecture, ...

Algorithms, theory, ...

Cognitive sciences

Neuroscience

optics

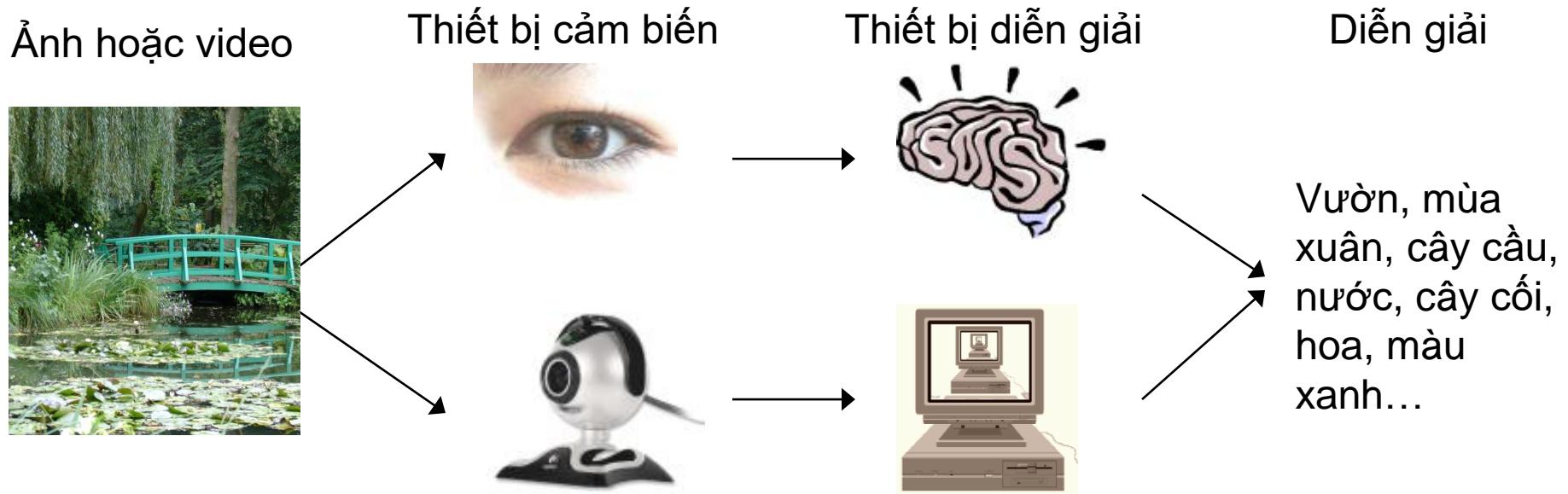
Image processing

Speech, NLP

Robotics

**Engineering**

# Thế nào là Thị giác máy tính?



# Mắt người rất nhạy

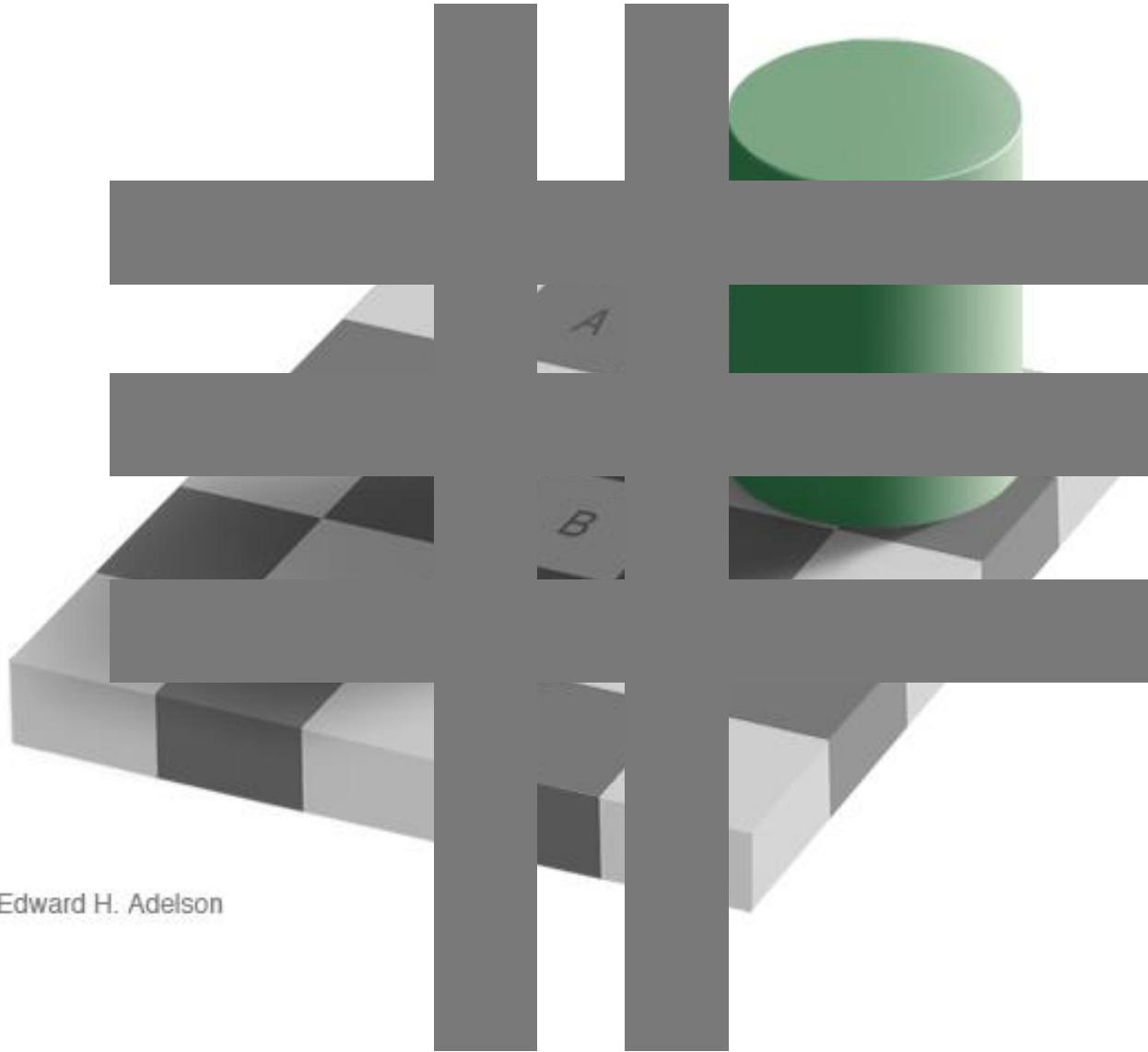


100 ms mỗi khung hình,  
Chưa hề nhìn thấy ảnh và không biết có người  
trong đó  
Nhưng có thể nhận ra dễ dàng

Potter, Biederman, etc. 1970s

# Thị giác con người vẫn có nhiều yếu điểm





Edward H. Adelson

# Mục tiêu của thị giác máy tính

- Cầu nối giữa các điểm ảnh biểu diễn bằng số với nghĩa



La Gare Montparnasse, 1895

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

Source: S. Narasimhan

# Tại sao nên học thi giác máy tính?

- Hữu ích: Ảnh và video khắp nơi!



**Google**  
Image Search

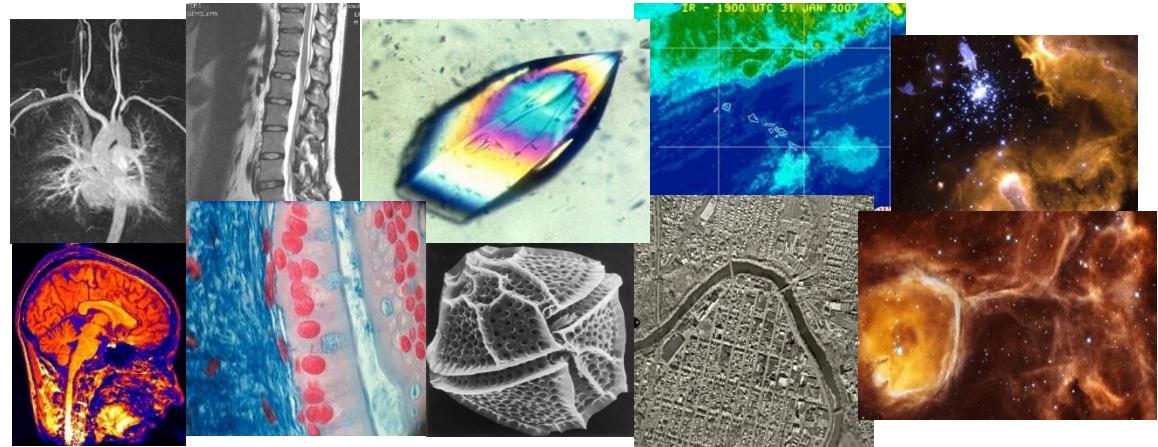
Google Photos

**flickr** GANNA

**webshots** beta

**picsearch**™

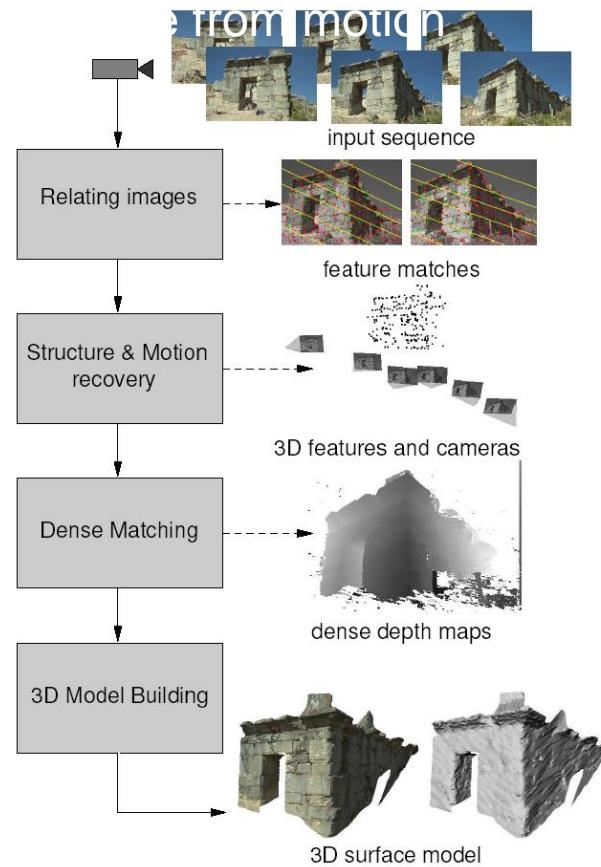
**YouTube**  
Broadcast Yourself™



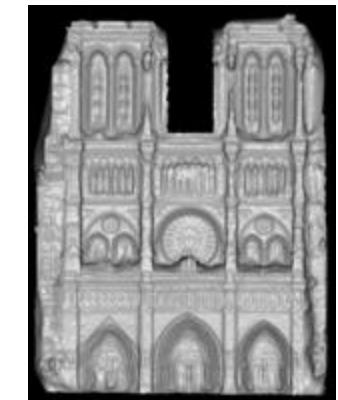
Giám sát và an ninh

# Thị giác máy có thể dùng như thiết bị đo đạc

Real-time stereo

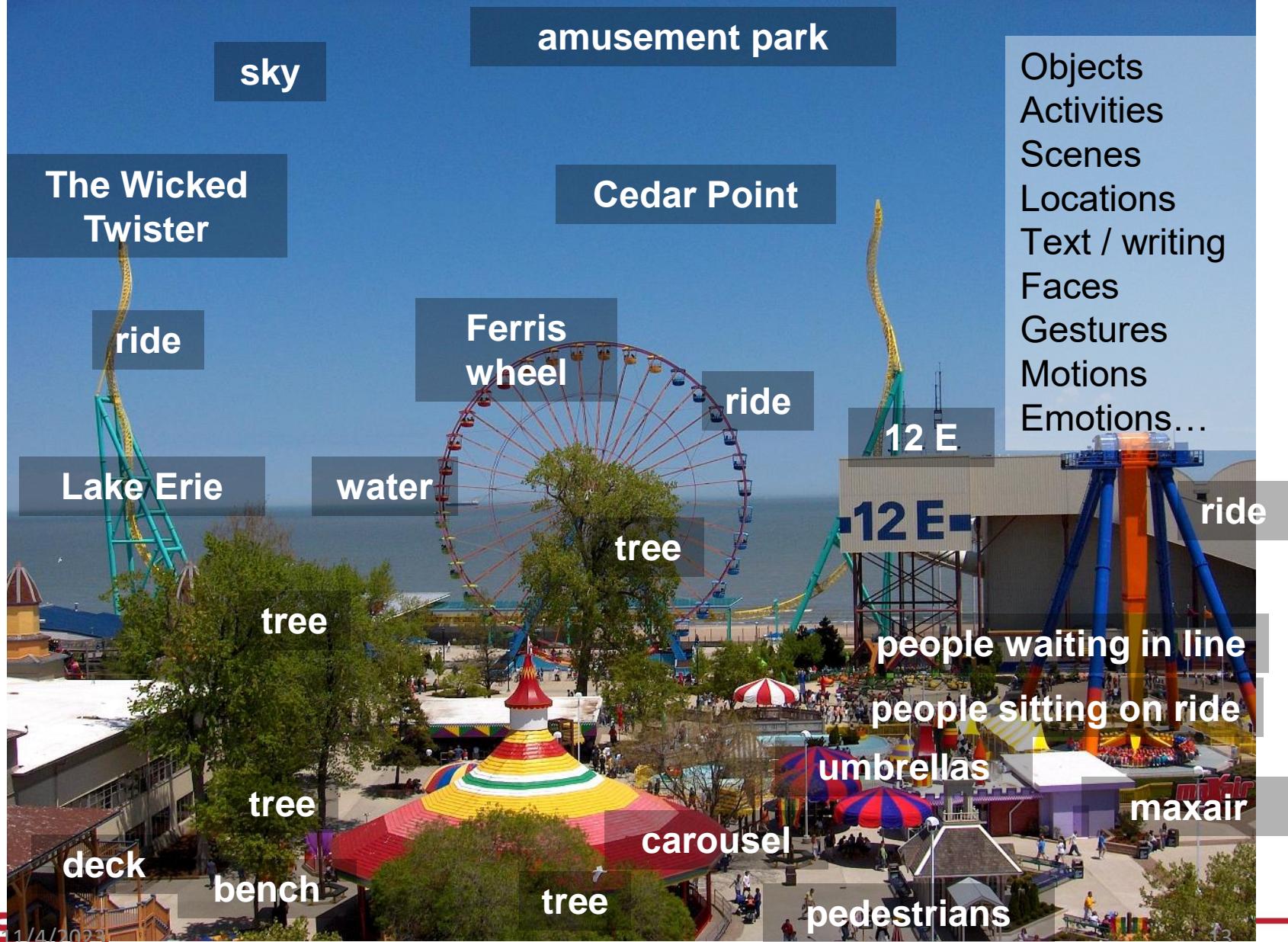


Pollefeys et al.

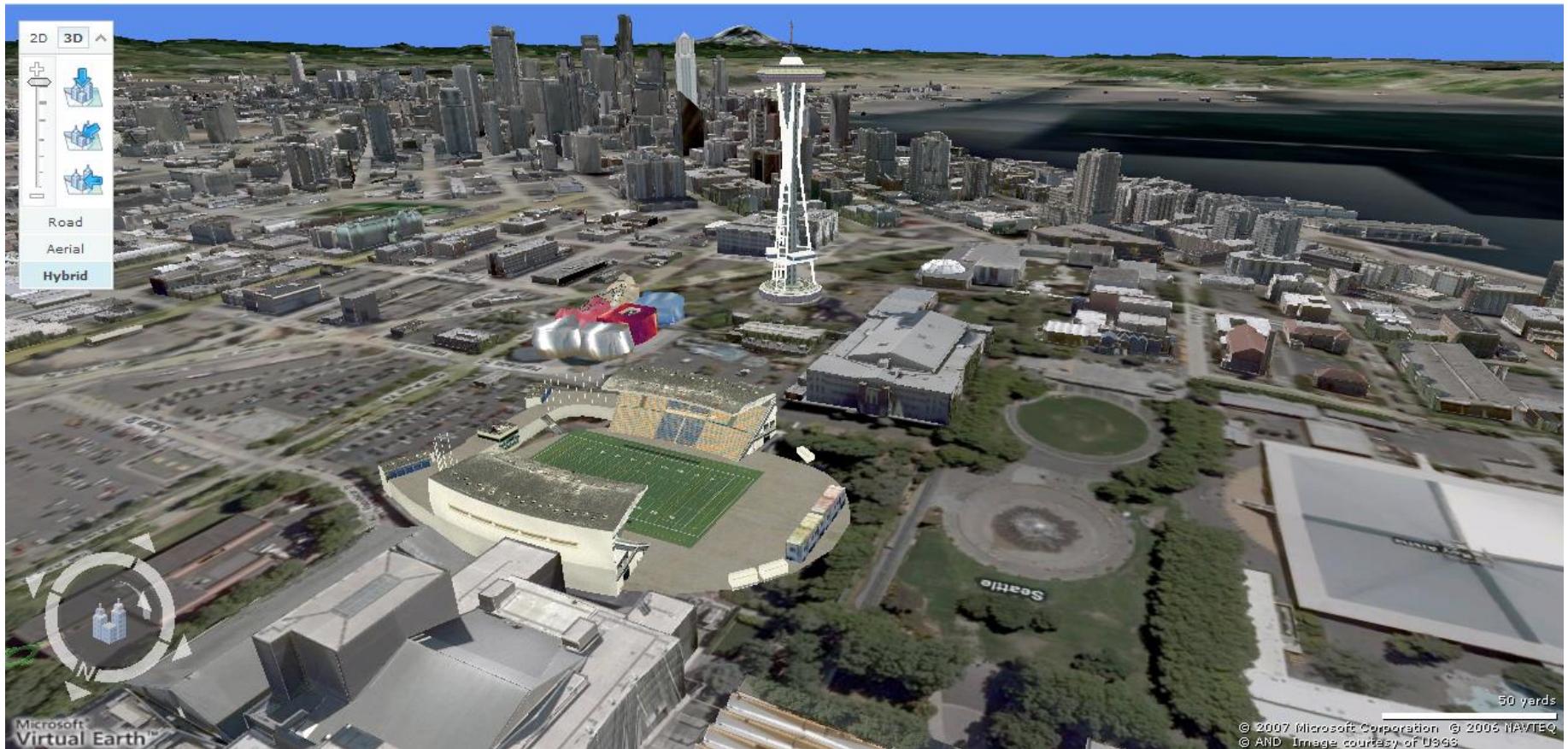


Goesele et al.

# Thị giác máy là nguồn thông tin ngữ nghĩa



# Mô hình hóa 3D thành phố



Bing maps, Google Streetview

Source: S. Seitz

# Phát hiện mặt



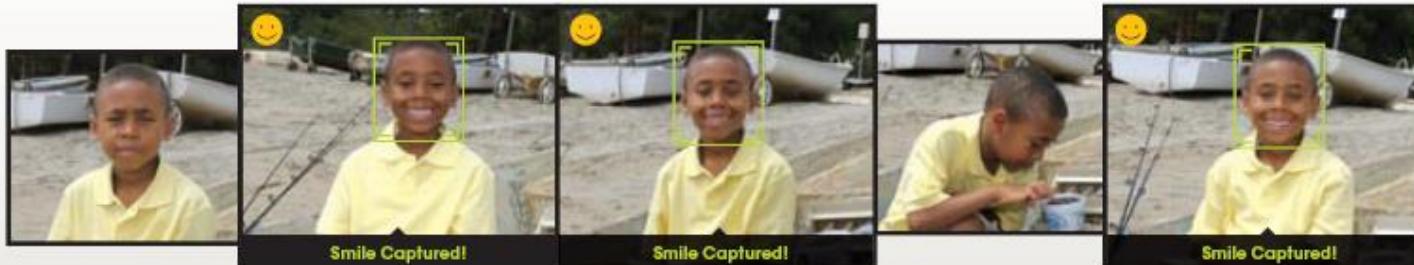
- Nhiều camera kỹ thuật số có khả năng tự động phát hiện khuôn mặt
  - Canon, Sony, Fuji, ...

Source: S. Seitz

# Phát hiện nụ cười

## The Smile Shutter flow

Imagine a camera smart enough to catch every smile! In Smile Shutter Mode, your Cyber-shot® camera can automatically trip the shutter at just the right instant to catch the perfect expression.



[Sony Cyber-shot® T70 Digital Still Camera](#)

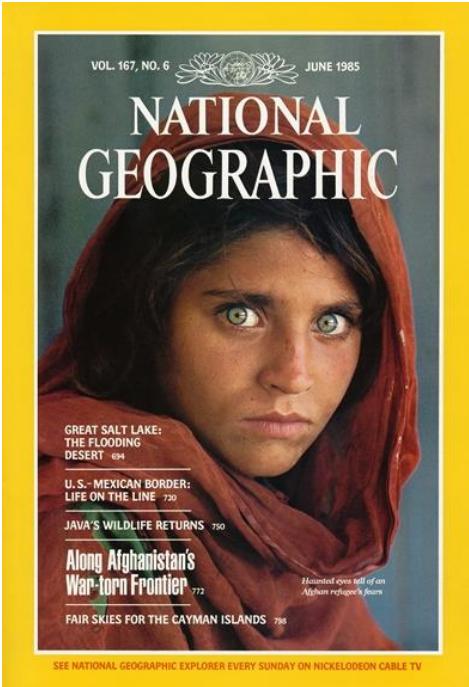
Source: S. Seitz

# Nhận dạng mặt: Apple iPhoto

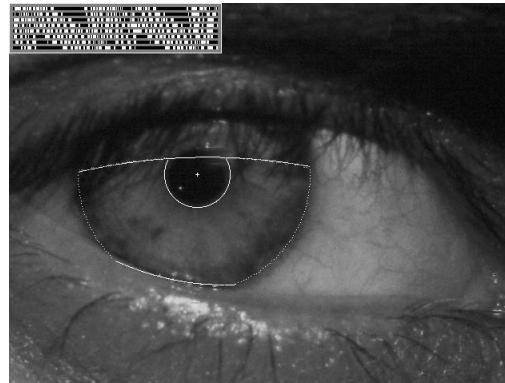
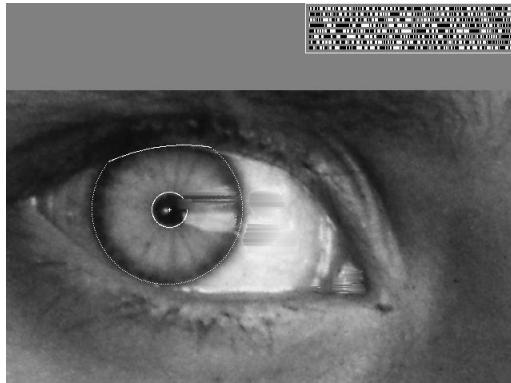


<http://www.apple.com/ilife/iphoto/>

# Sinh trắc học



## How the Afghan Girl was Identified by Her Iris Patterns



Source: S. Seitz

# Sinh trắc học

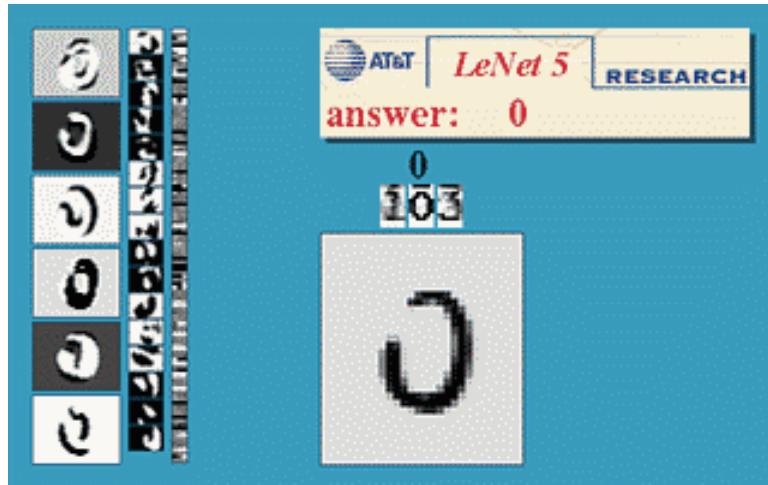


Nhận dạng vân tay (Fingerprint scanners) trên nhiều laptop và thiết bị



Hệ thống nhận dạng mặt xuất hiện ngày càng nhiều  
Ví dụ: iphone X vừa giới thiệu  
faceID

# Nhận dạng văn bản (OCR)



Nhận dạng chữ số viết tay, AT&T labs



Đọc biển số xe

[http://en.wikipedia.org/wiki/Automatic\\_number\\_plate\\_recognition](http://en.wikipedia.org/wiki/Automatic_number_plate_recognition)

# Tương tác người máy và games



Microsoft's Kinect



Sony EyeToy



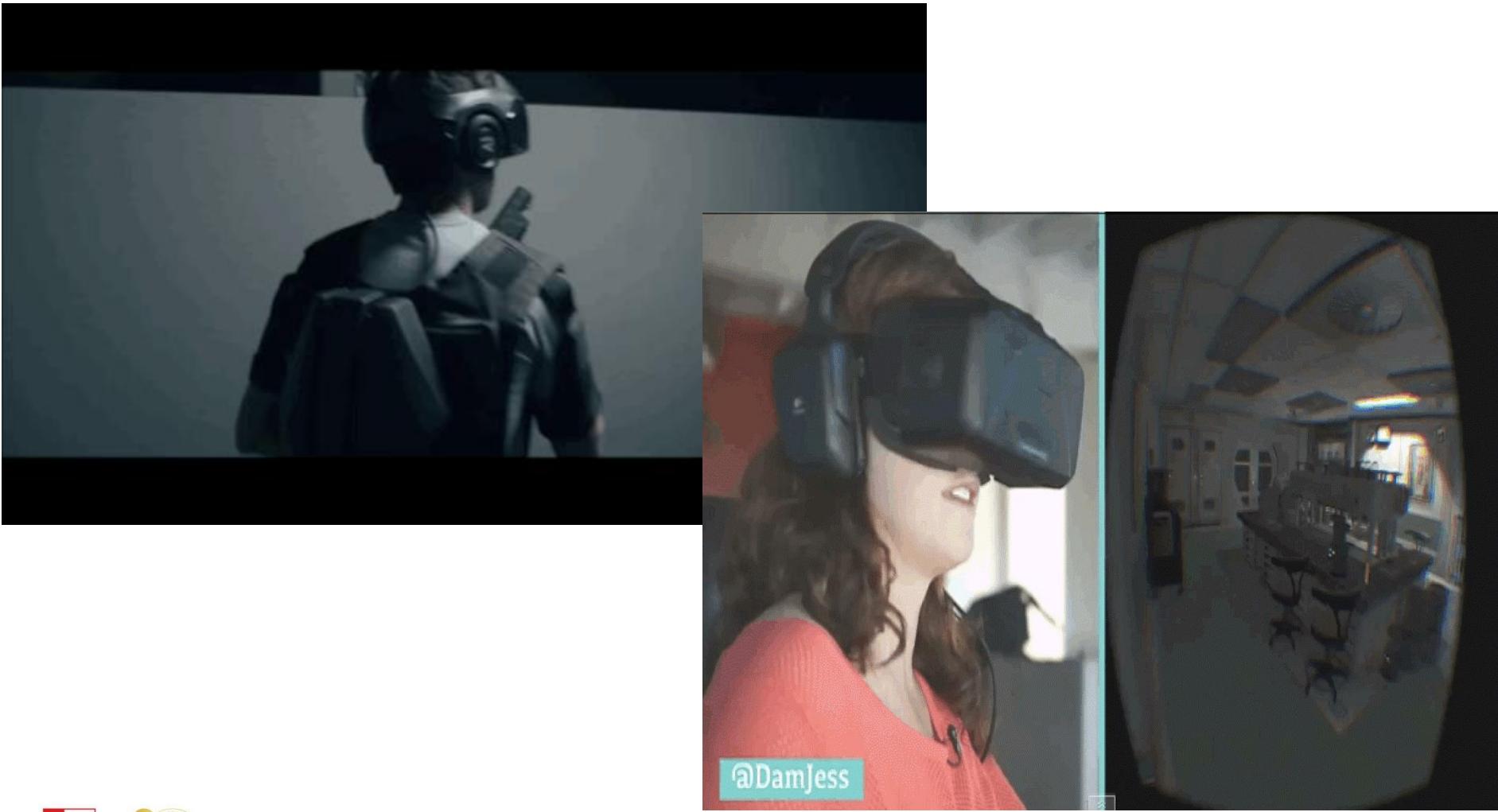
Assistive technologies

Source: S. Seitz

# Thực tại tăng cường



# Thực tại ảo



@DamJess

# Ứng dụng trong robotics và thám hiểm vũ trụ



[NASA's Mars Exploration Rover Spirit](#) captured this westward view from atop a low plateau where Spirit spent the closing months of 2007.

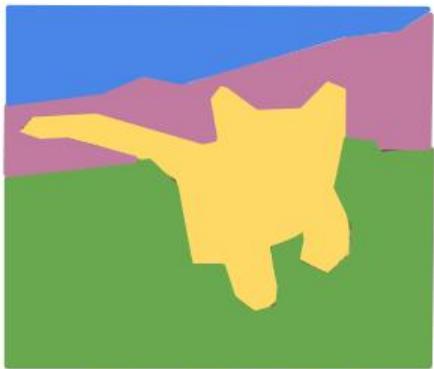
Thị giác sử dụng cho nhiều tác vụ khác nhau:

- Chụp Panorama
- Mô hình hóa 3D bề mặt sao hỏa
- Phát hiện vật cản, bám vết vị trí
- Chi tiết xem "[Computer Vision on Mars](#)" của Matthies et al.

# Giới thiệu về bài toán phát hiện đối tượng

# Các bài toán thị giác máy

Semantic Segmentation



No objects, just pixels

Classification + Localization



Single Object

Object Detection



Multiple Object

Instance Segmentation

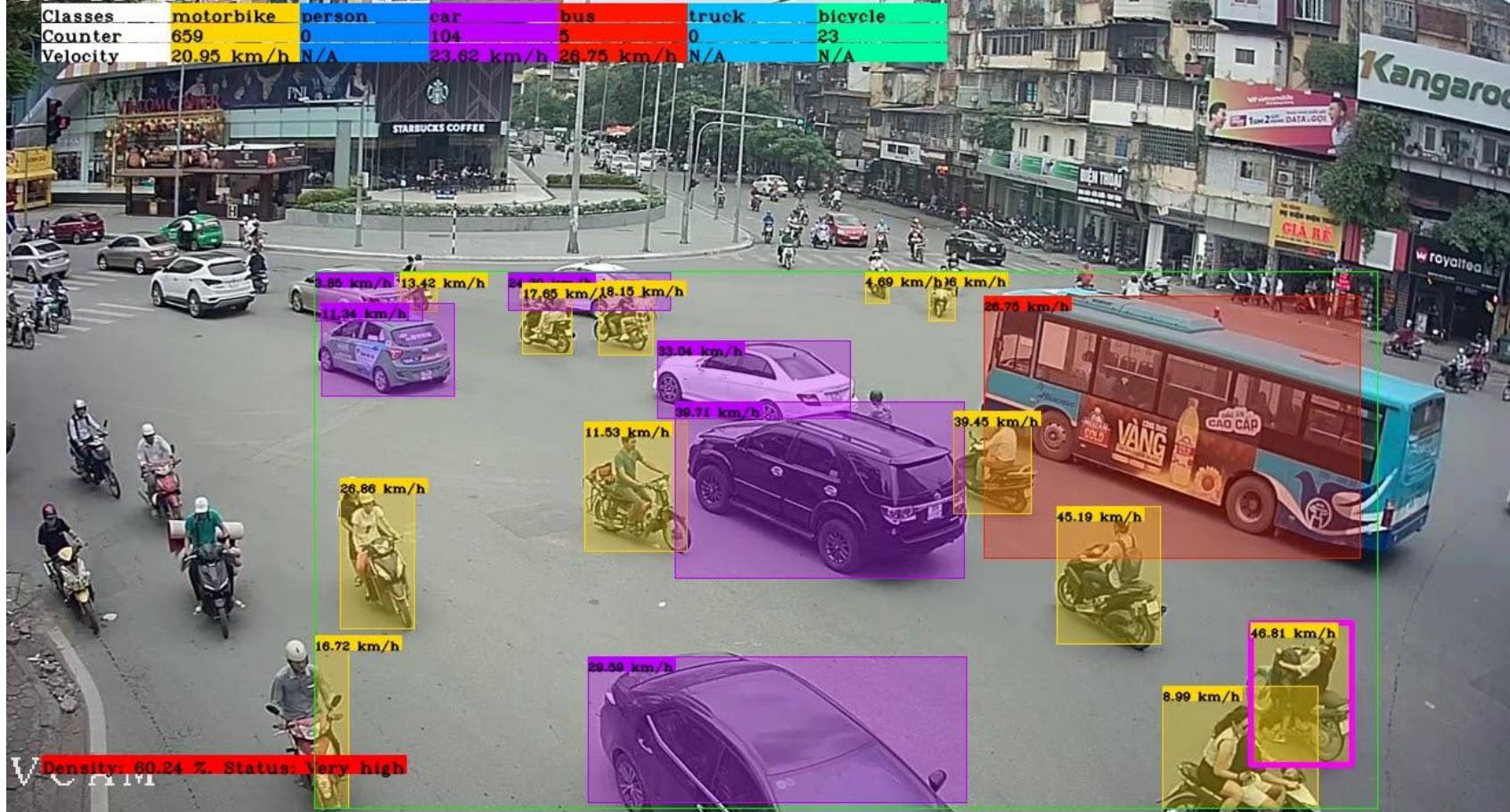


[This image is CC0 public domain](#)

# Một số ứng dụng bài toán phát hiện đối tượng

- Giao thông thông minh

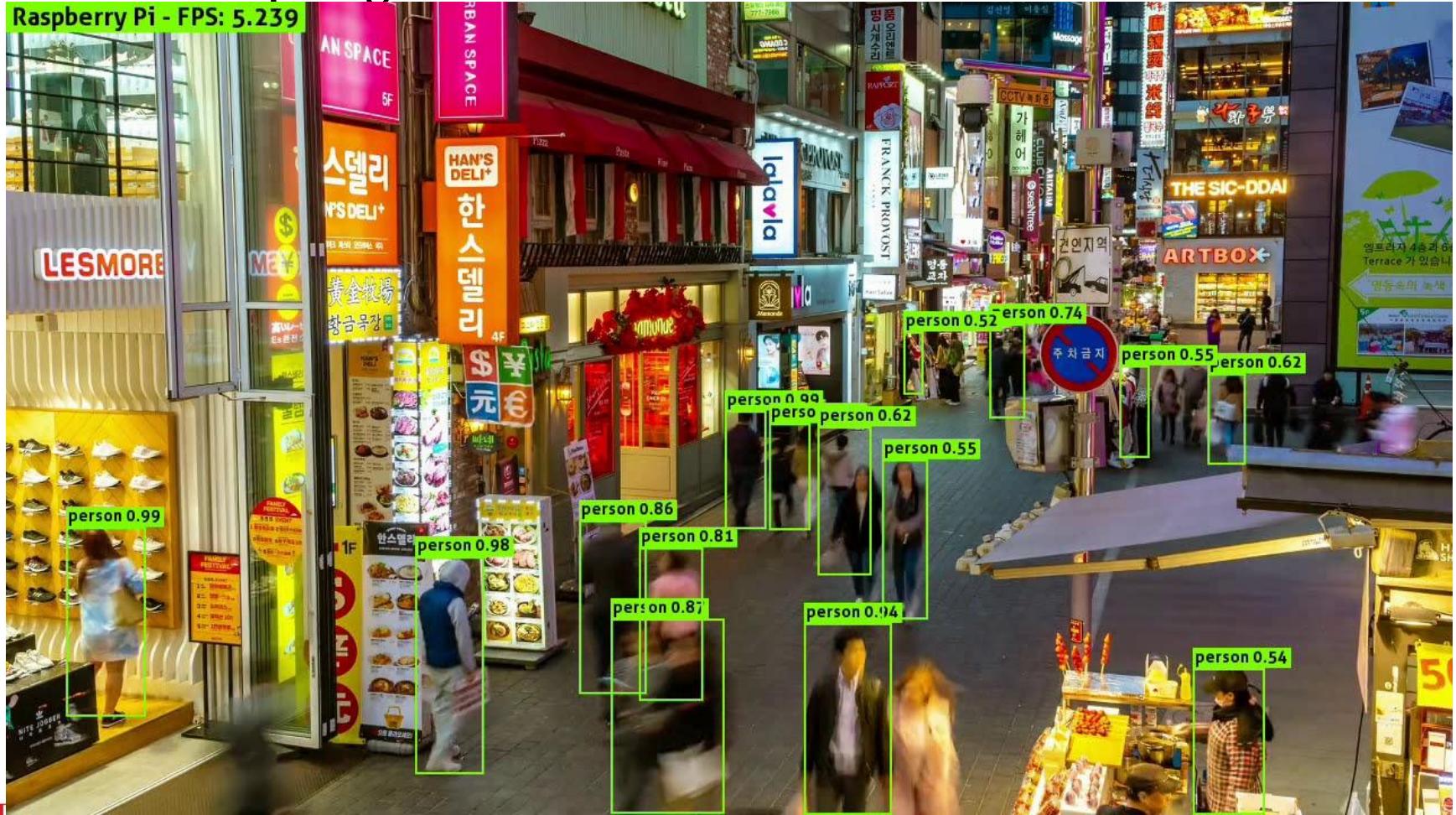
TRAFFIC MONITORING SYSTEM



# Một số ứng dụng bài toán phát hiện đối tượng

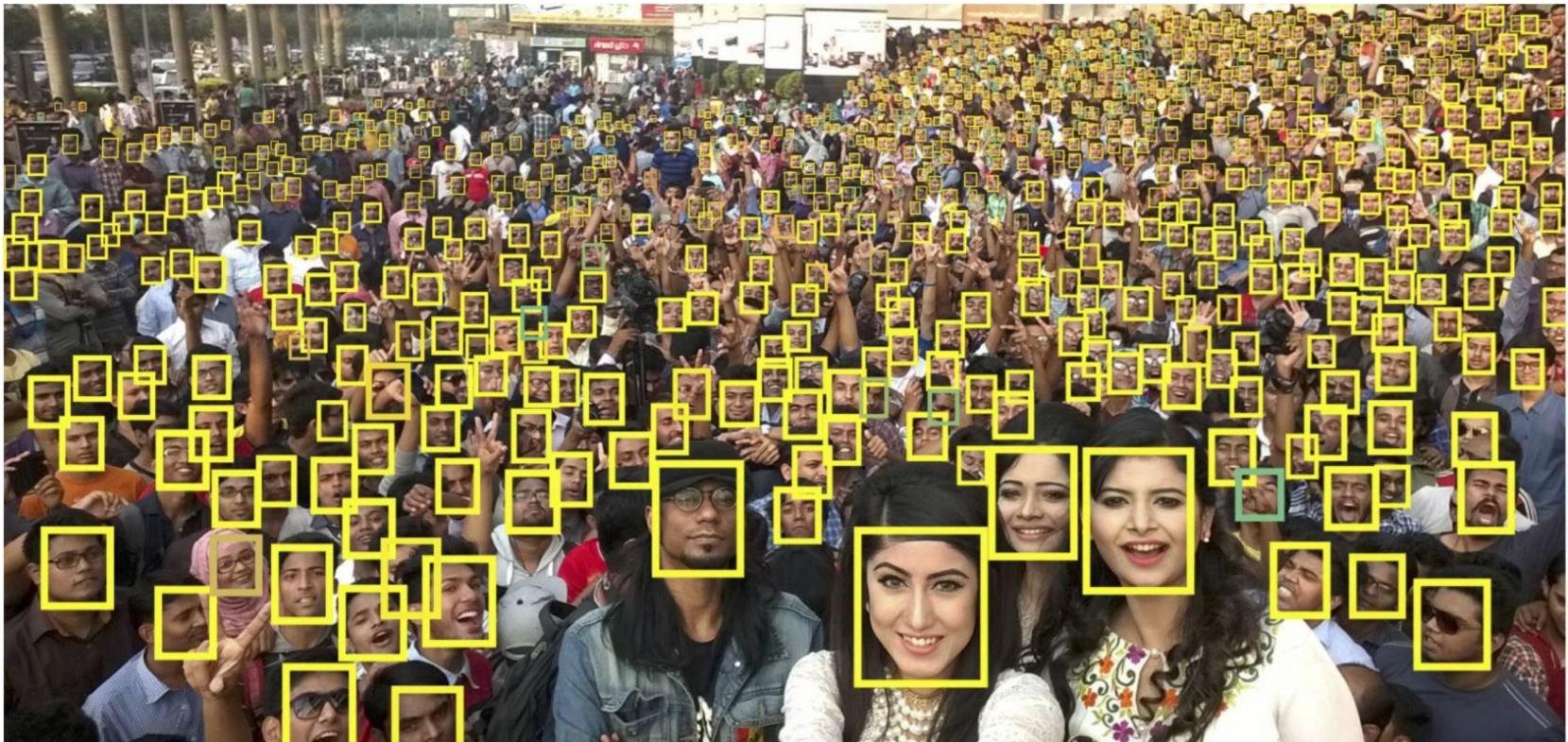
- Phát hiện người

Raspberry Pi - FPS: 5.239



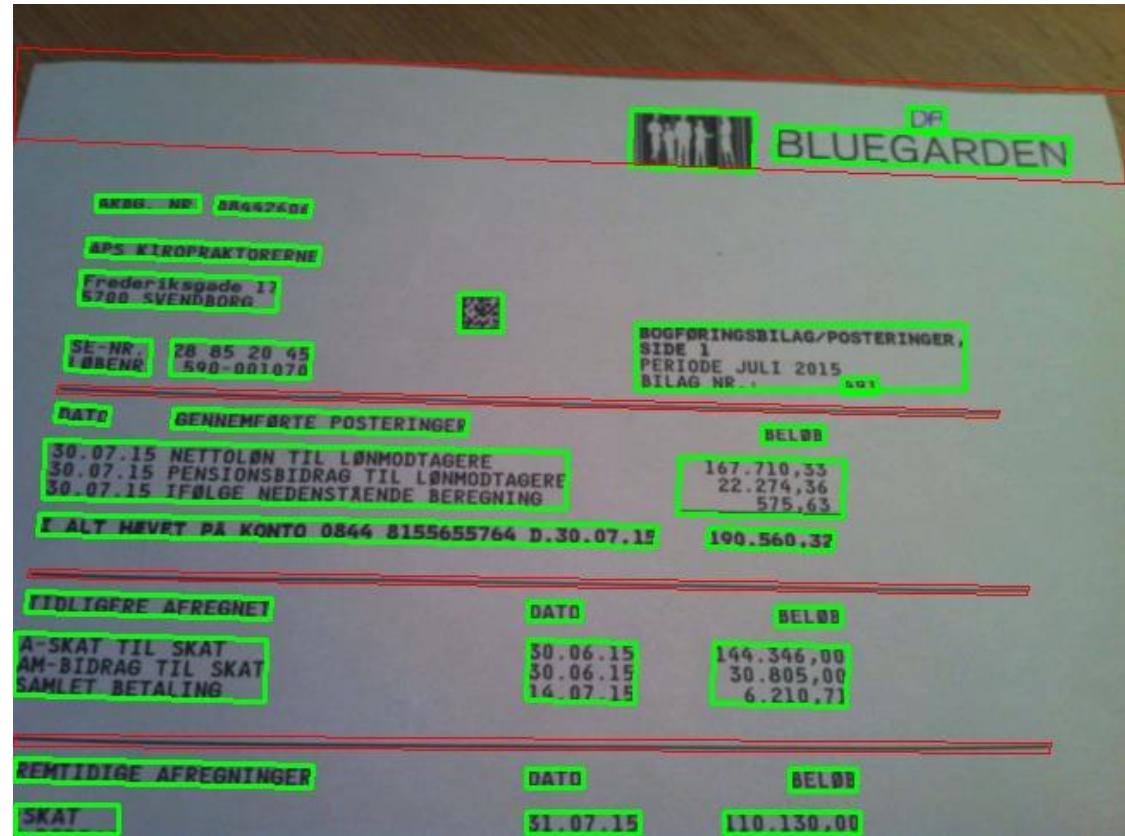
# Một số ứng dụng bài toán phát hiện đối tượng

- Phát hiện khuôn mặt



# Một số ứng dụng bài toán phát hiện đối tượng

- Phát hiện văn bản



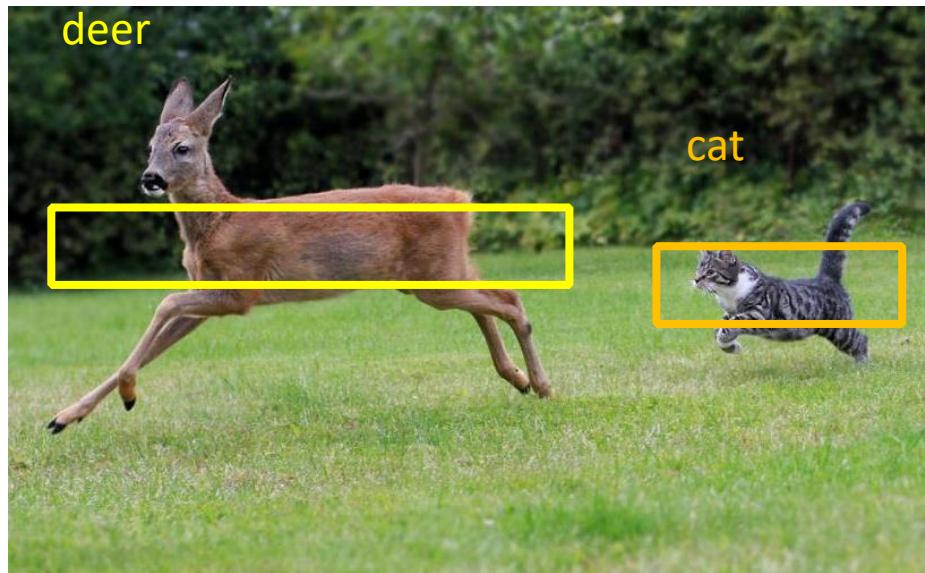
# Một số ứng dụng bài toán phát hiện đối tượng

- Robot tự động hái dâu



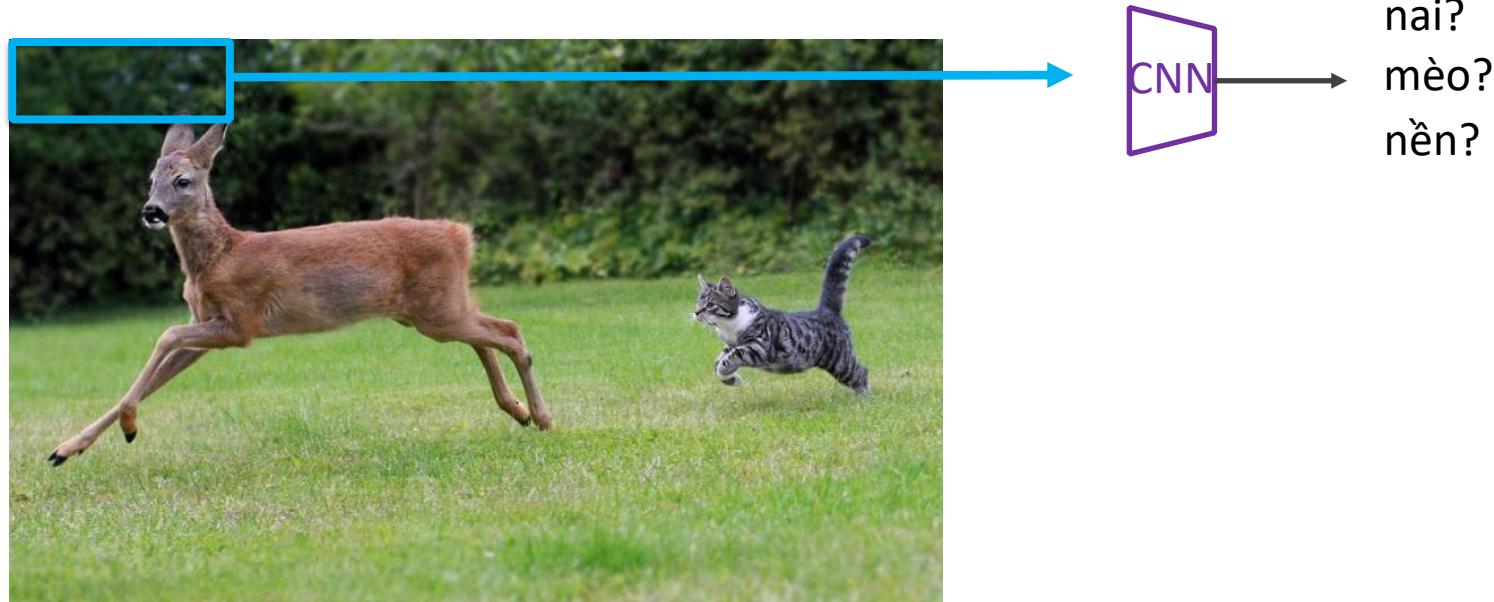
# Giới thiệu một số mạng đề xuất vùng (two-stage object detectors)

# Tiếp cận quét cửa sổ (sliding windows)



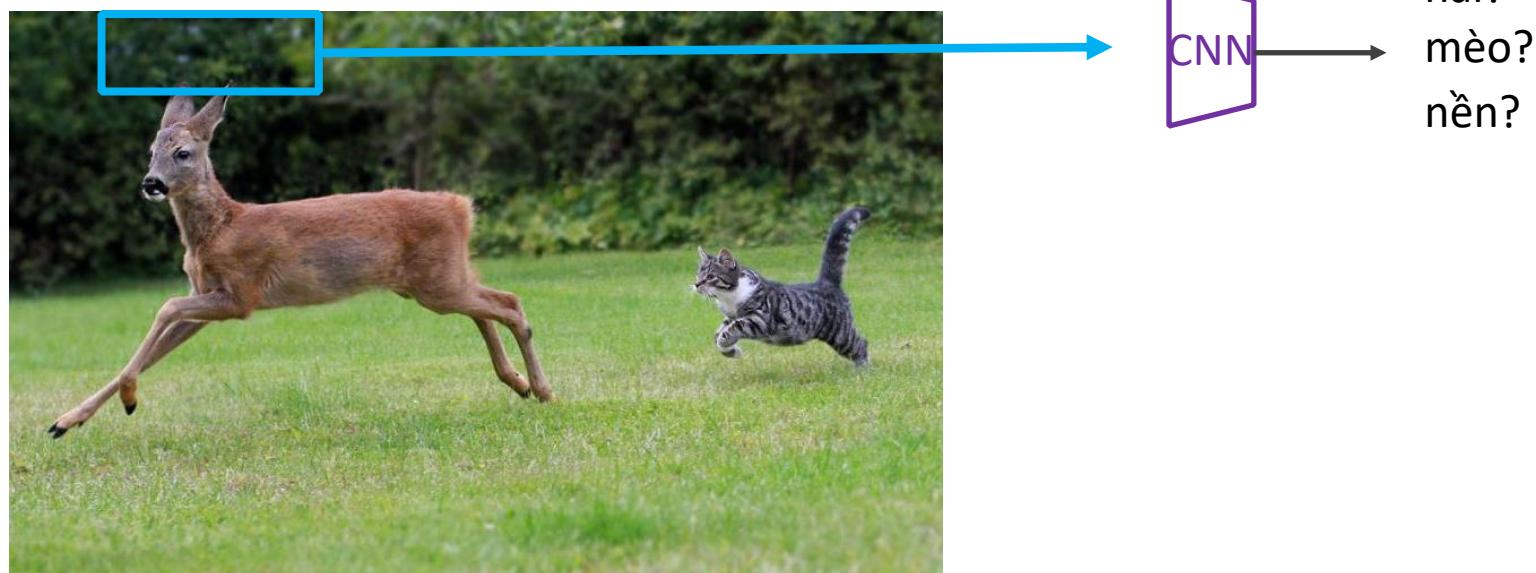
# Tiếp cận quét cửa sổ (sliding windows)

- 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.



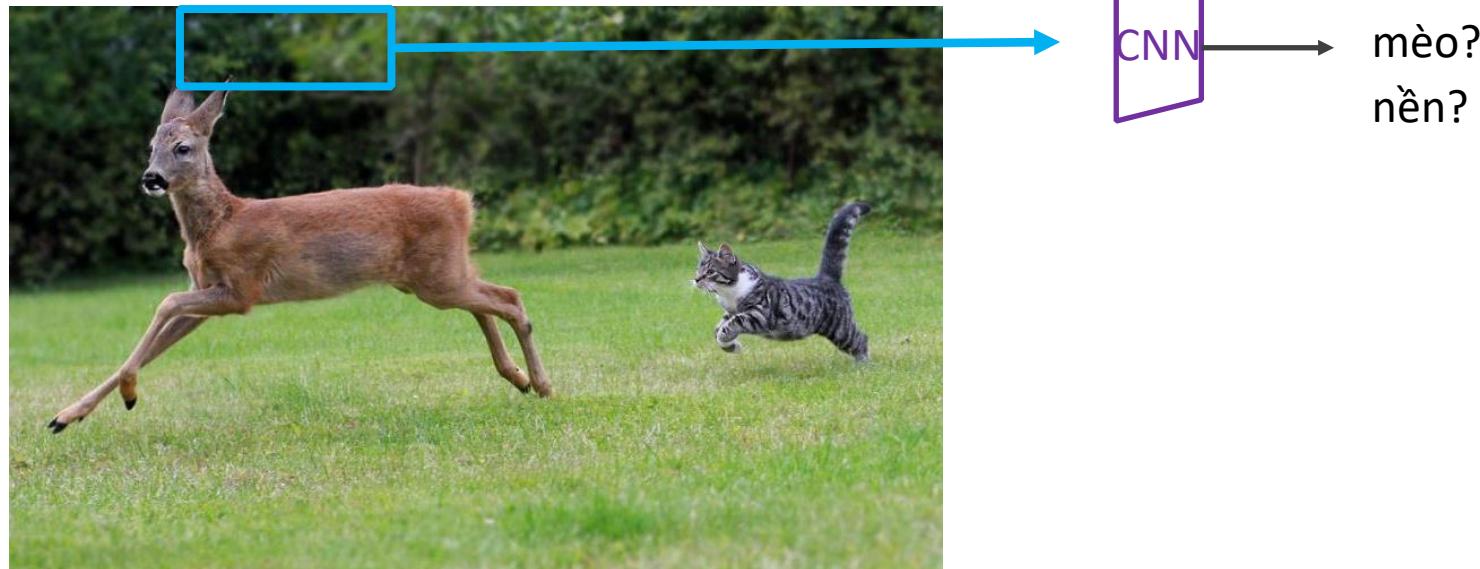
# Tiếp cận quét cửa sổ (sliding windows)

- 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.



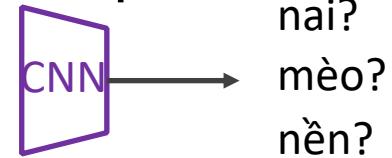
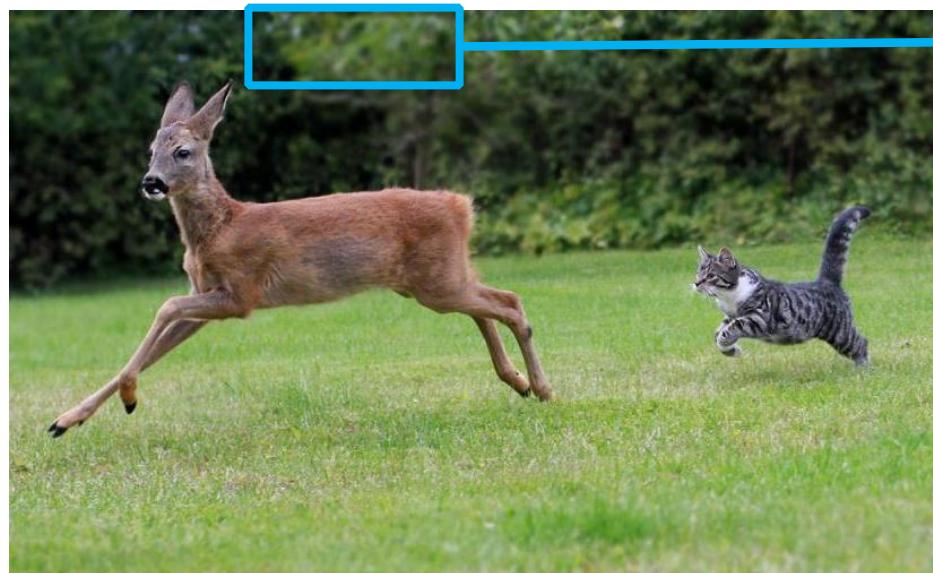
# Tiếp cận quét cửa sổ (sliding windows)

- 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.



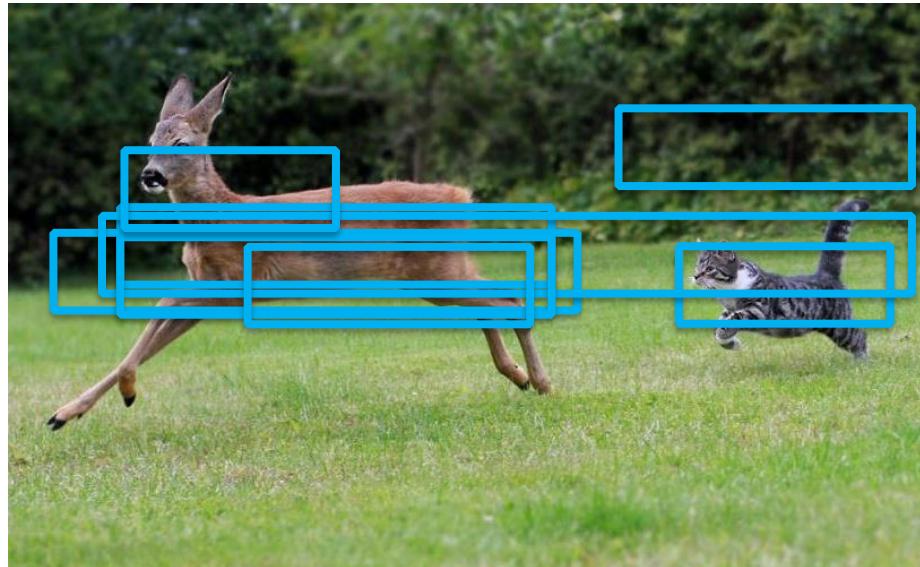
# Tiếp cận quét cửa sổ (sliding windows)

- 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.



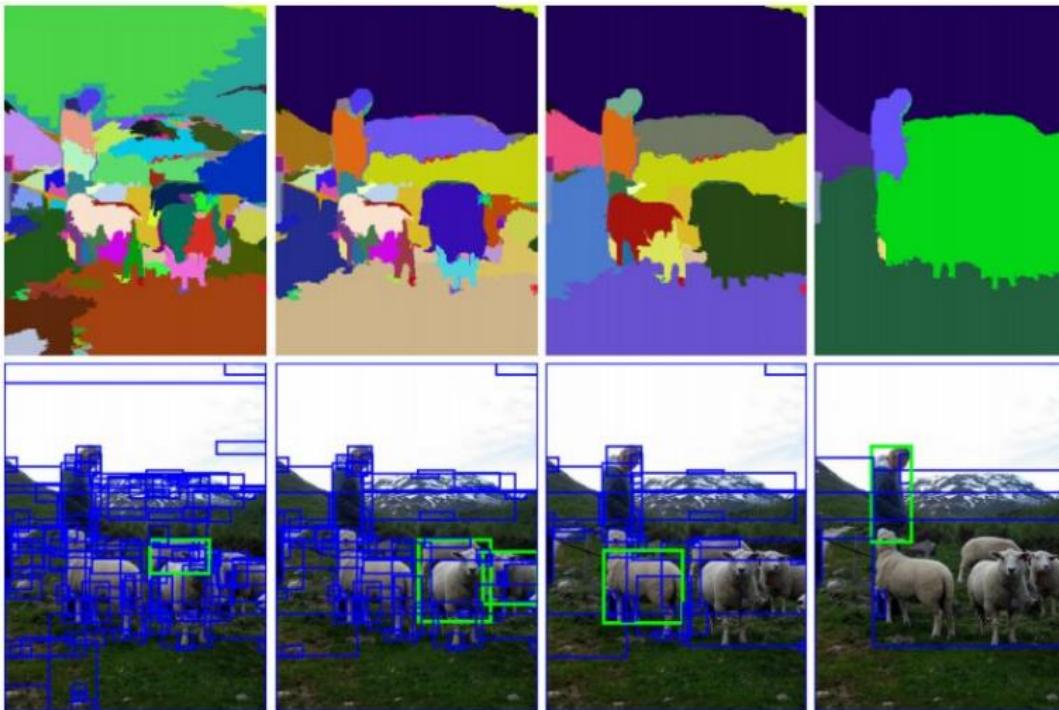
# Tiếp cận dựa trên đề xuất vùng

- 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



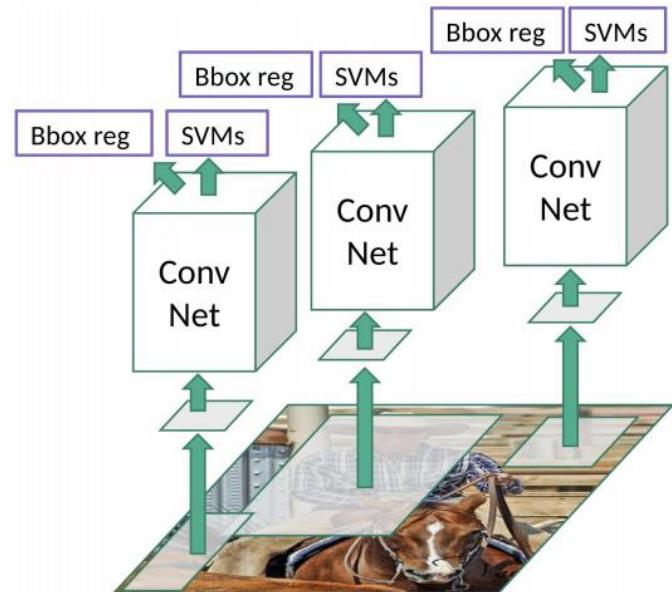
# SS: Selective Search

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

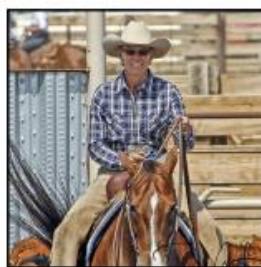


# R-CNN (Region-based ConvNet)

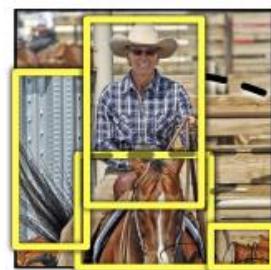
- Đề xuất một số vùng tiềm năng bằng thuật toán khác, chẳng hạn selective search
- Dùng mạng CNN trích xuất đặc trưng từng vùng rồi phân loại bằng SVM



**R-CNN: *Regions with CNN features***

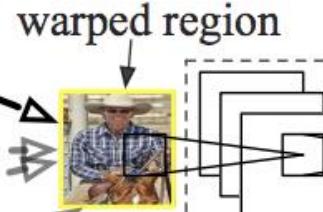


1. Input image

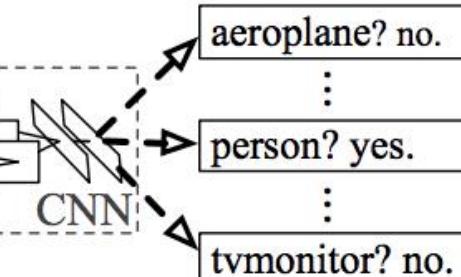


2. Extract region proposals (~2k)

warped region



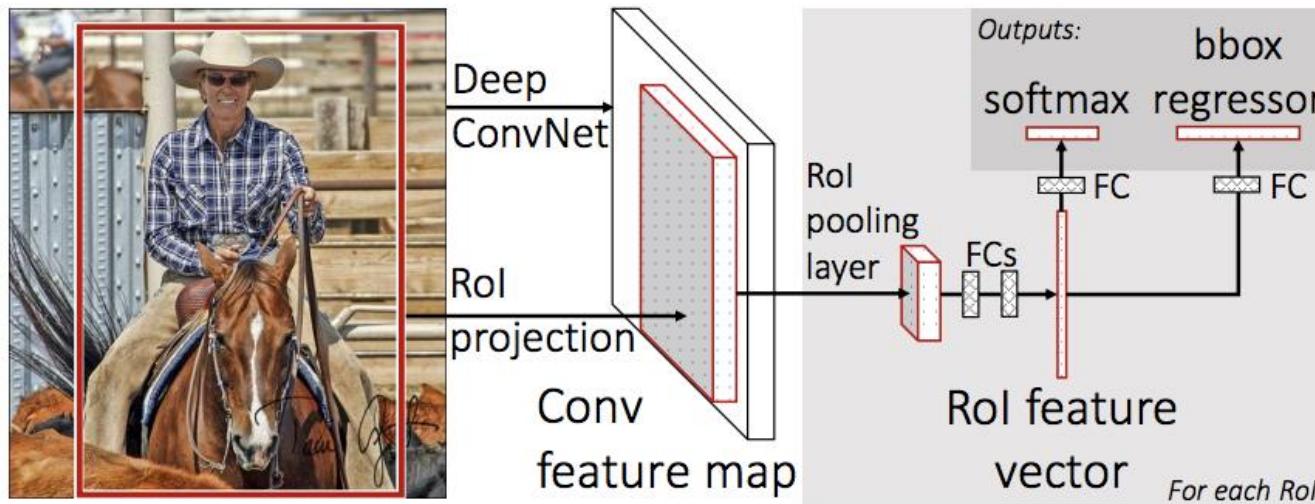
3. Compute CNN features



4. Classify regions

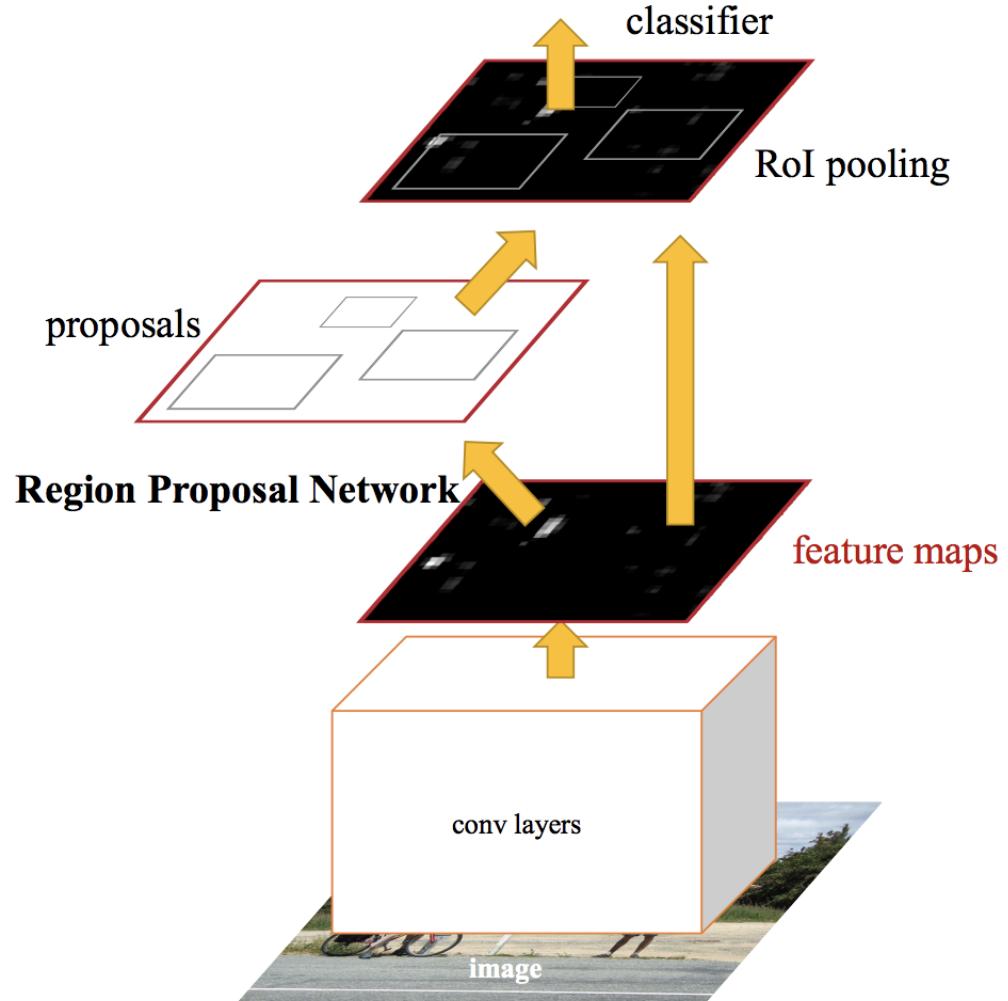
# Fast-RCNN

- 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)



# Giới thiệu một số mạng không đề xuất vùng (one-stage object detectors)

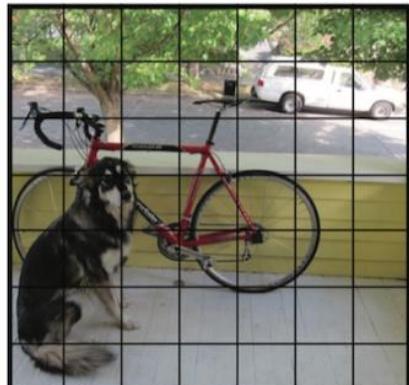
# Đặc điểm các mạng không đề xuất vùng

- 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.

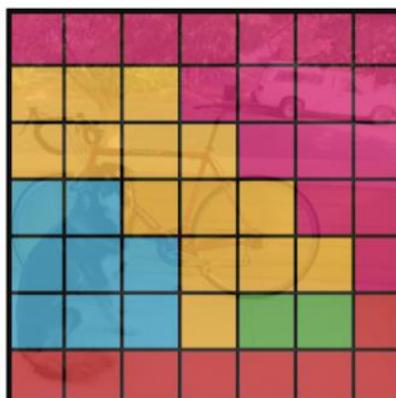
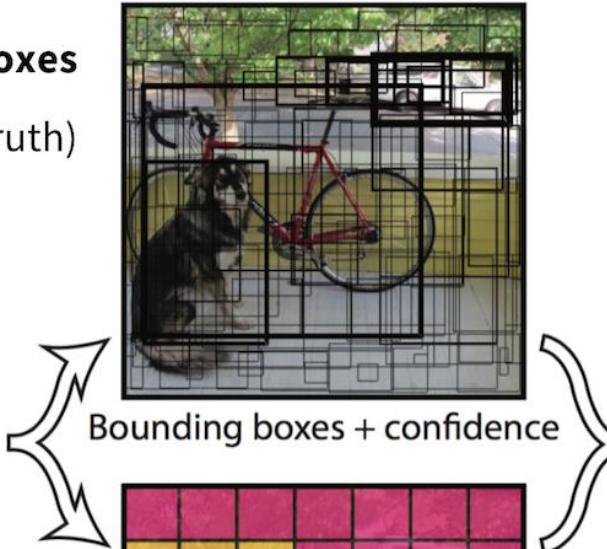
# YOLO- You Only Look Once

$S \times S \times B$  bounding boxes

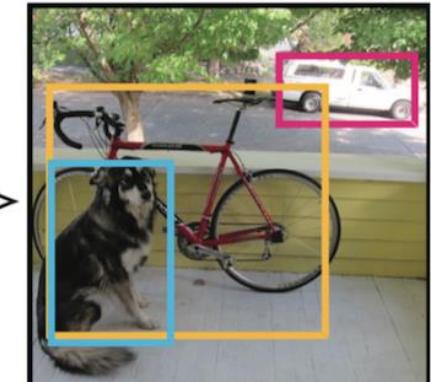
**confidence** =  $Pr(\text{object}) \times \text{IoU}(\text{pred}, \text{truth})$



$S \times S$  grid on input



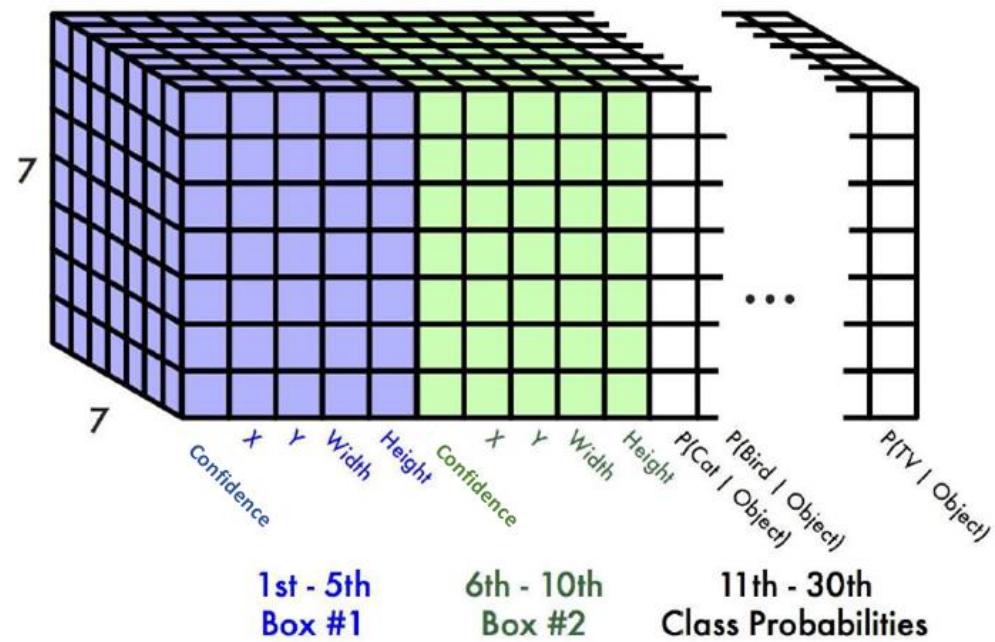
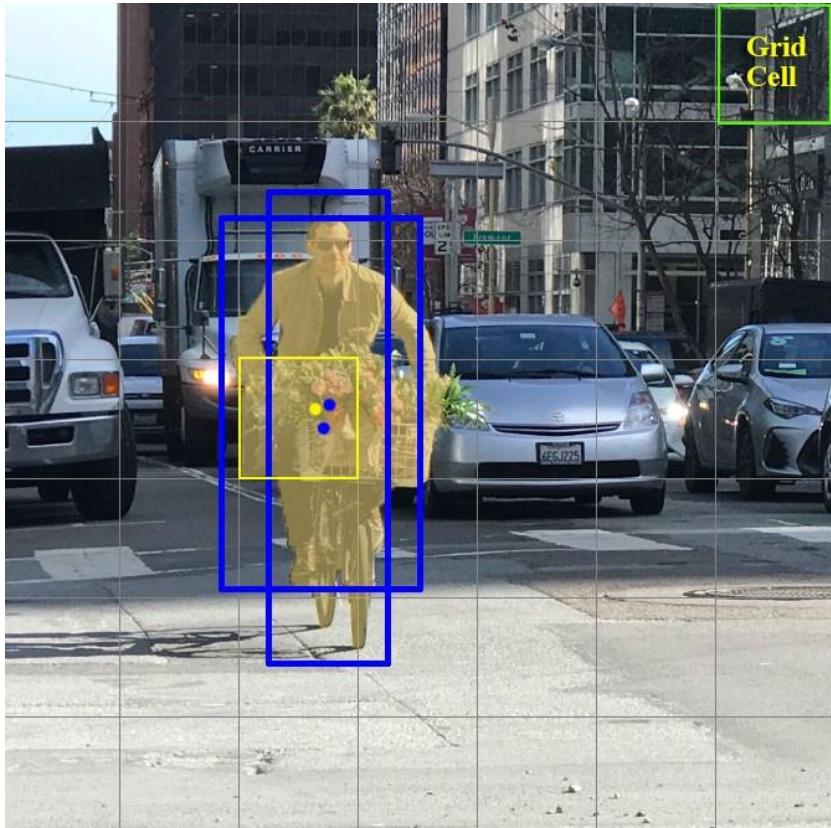
$Pr(\text{Class}_i | \text{object})$



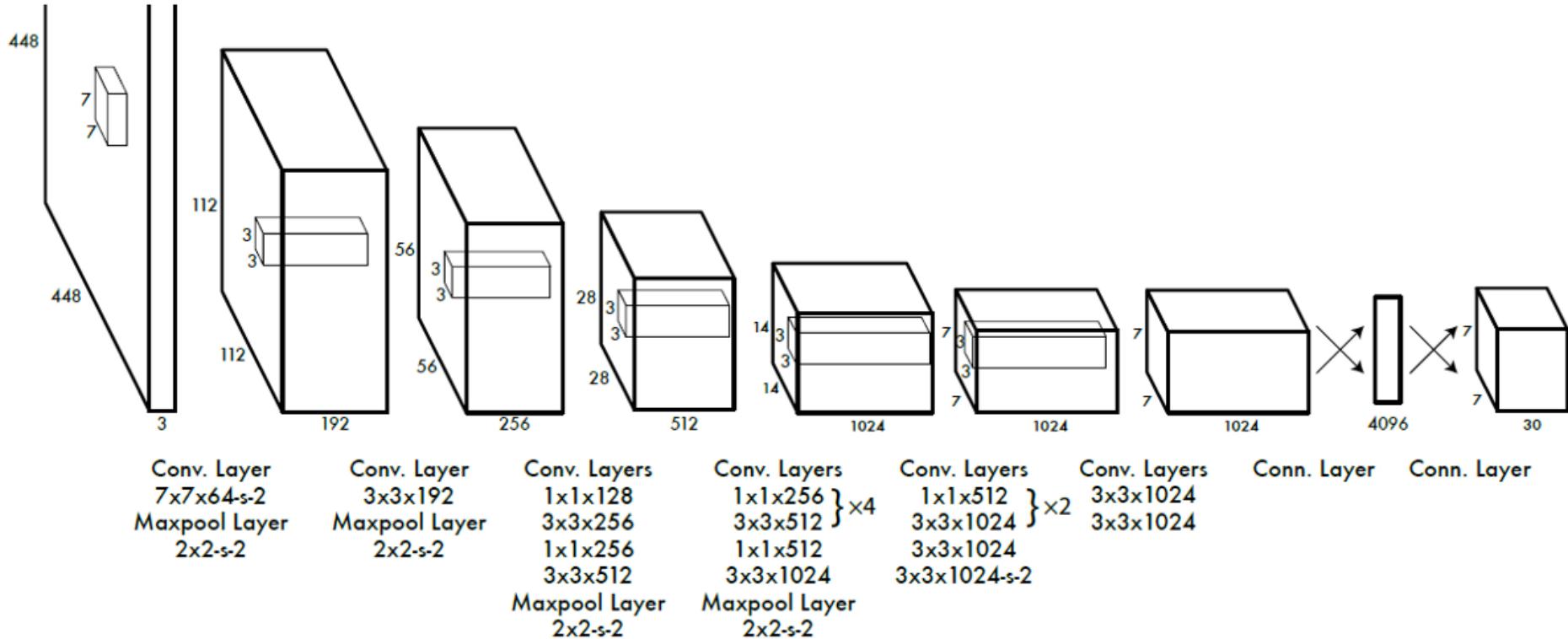
Final detections

$$\Pr(\text{Class}_i | \text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

# YOLO- You Only Look Once



# YOLO- You Only Look Once

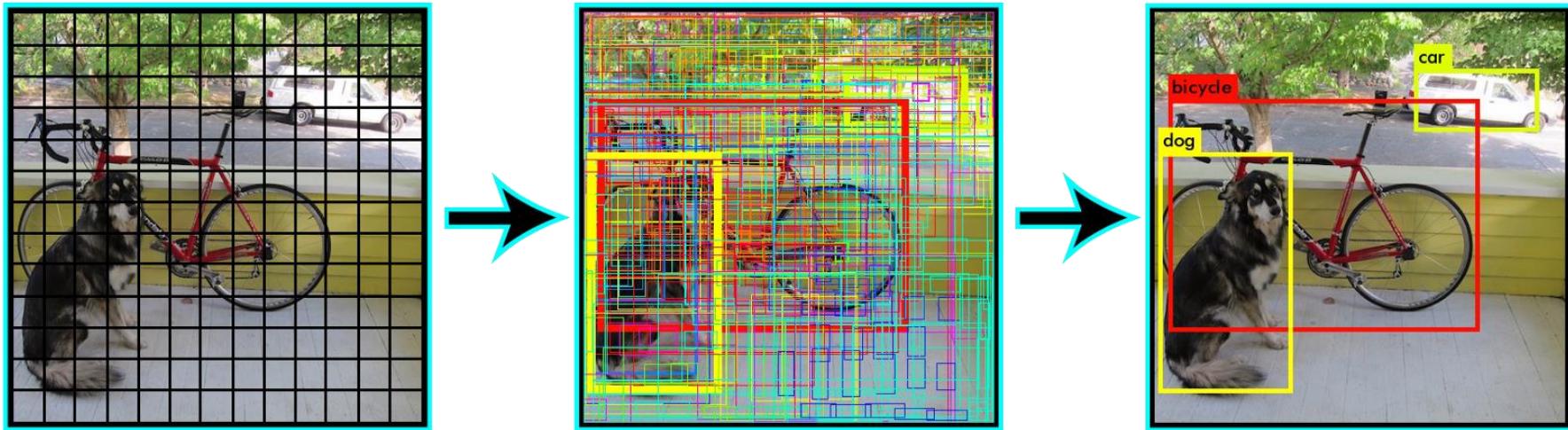


# YOLO- You Only Look Once

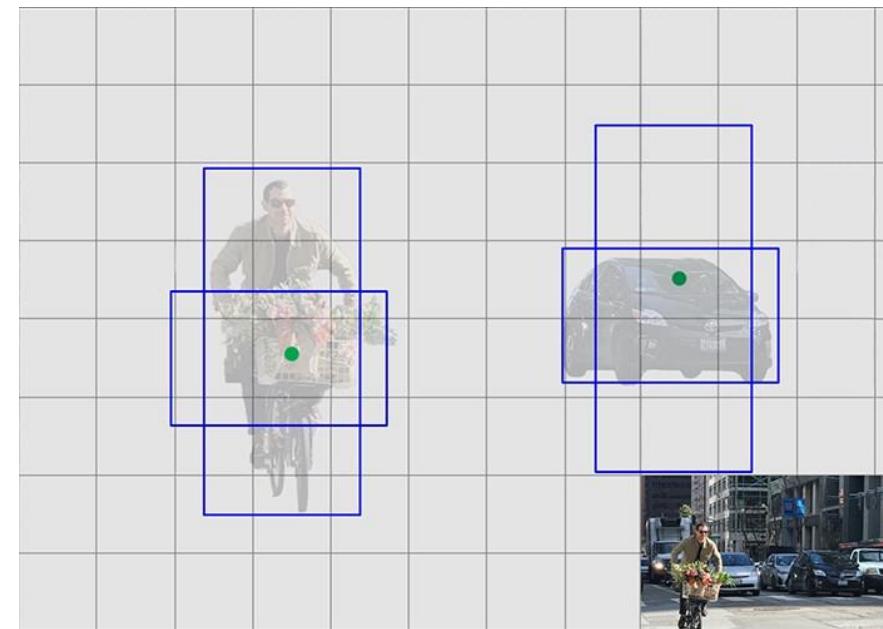
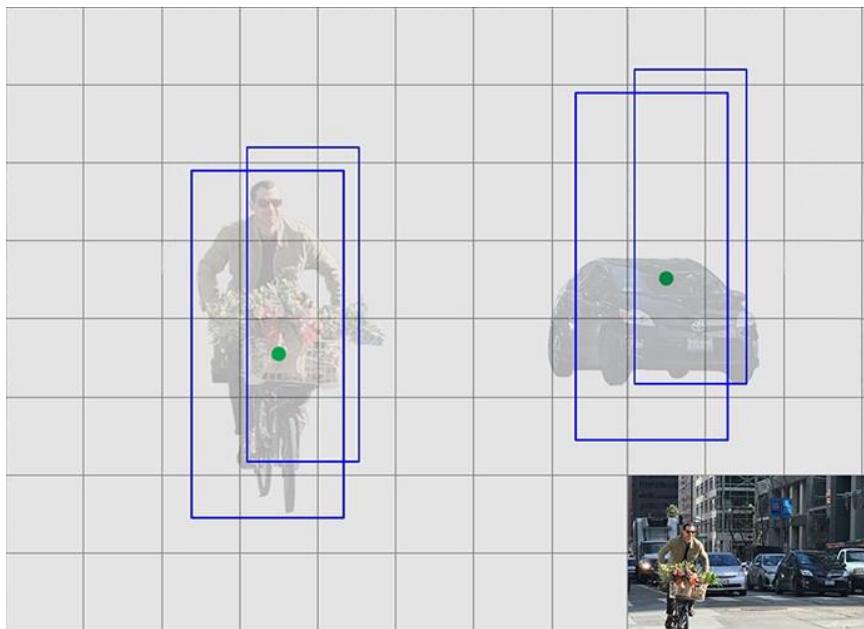
$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \quad \begin{array}{l} \text{1 when there is object, 0 when there is no object} \\ \text{Bounding Box Location (x, y) when there is object} \end{array}$$
$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \quad \begin{array}{l} \text{Bounding Box size (w, h)} \\ \text{when there is object} \end{array}$$
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left( C_i - \hat{C}_i \right)^2 \quad \text{Confidence when there is object}$$
$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} \left( C_i - \hat{C}_i \right)^2 \quad \begin{array}{l} \text{1 when there is no object, 0 when there is object} \\ \text{Confidence when there is no object} \end{array}$$
$$+ \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \quad \text{Class probabilities when there is object}$$

# YOLO- You Only Look Once

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

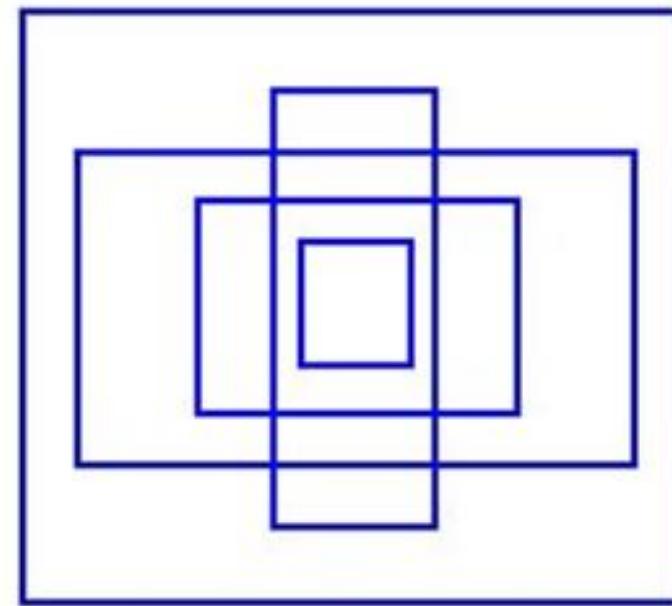


# YOLO v2



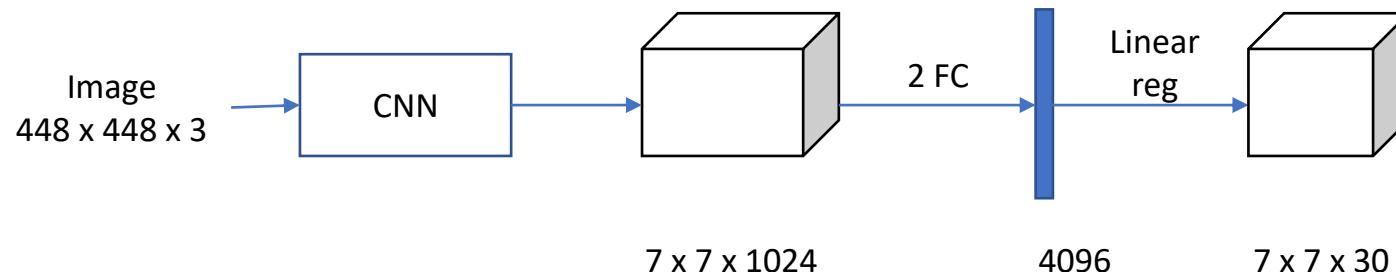
# YOLO v2

- 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

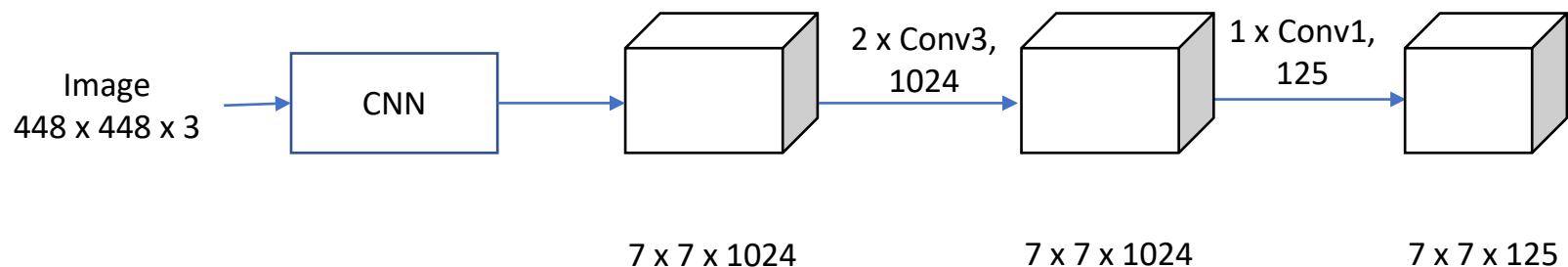


5 anchor box

# YOLO v2



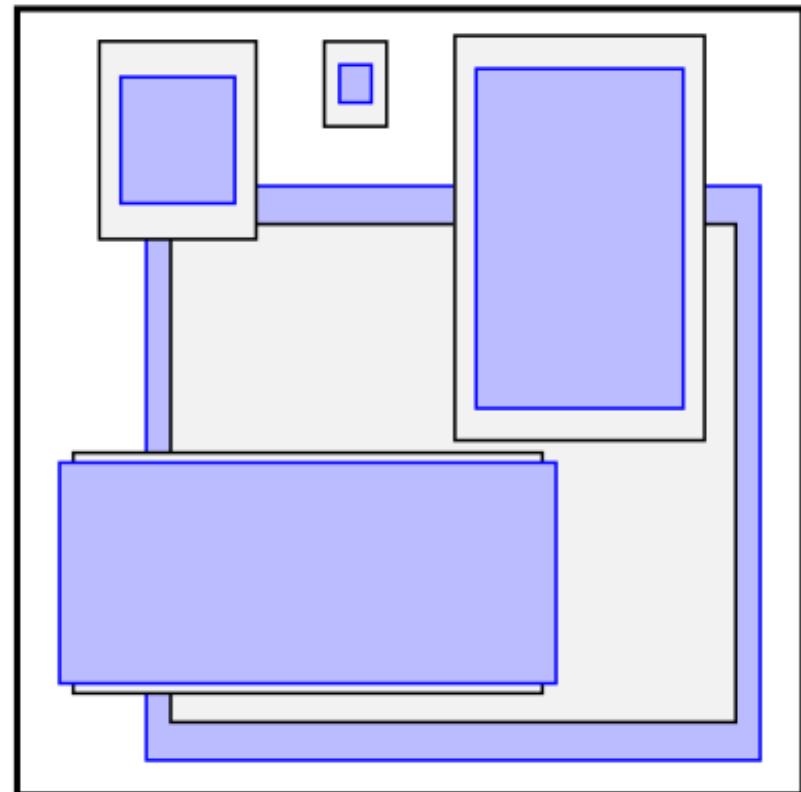
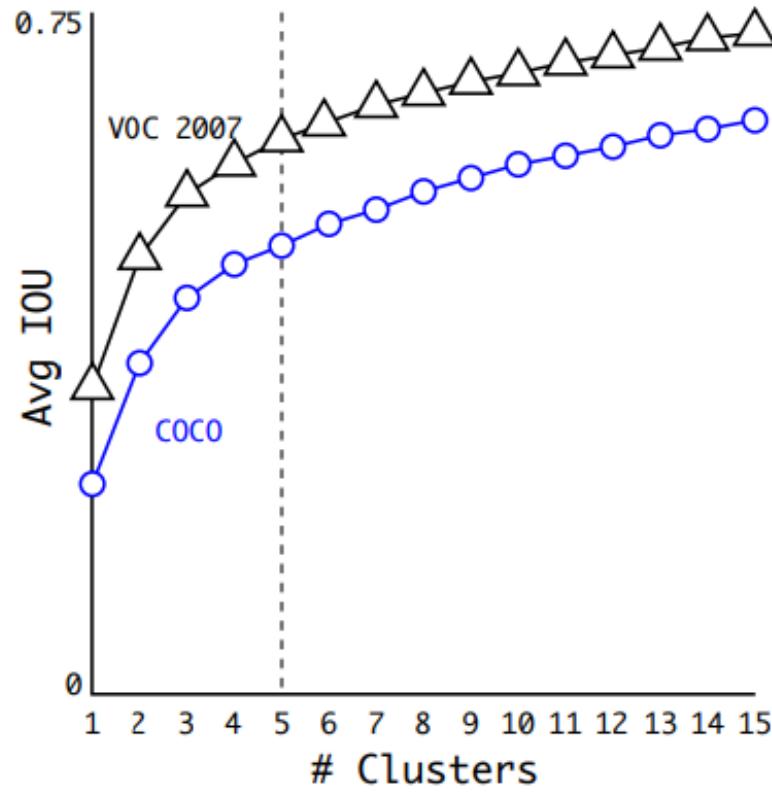
YOLO v1



YOLO v2

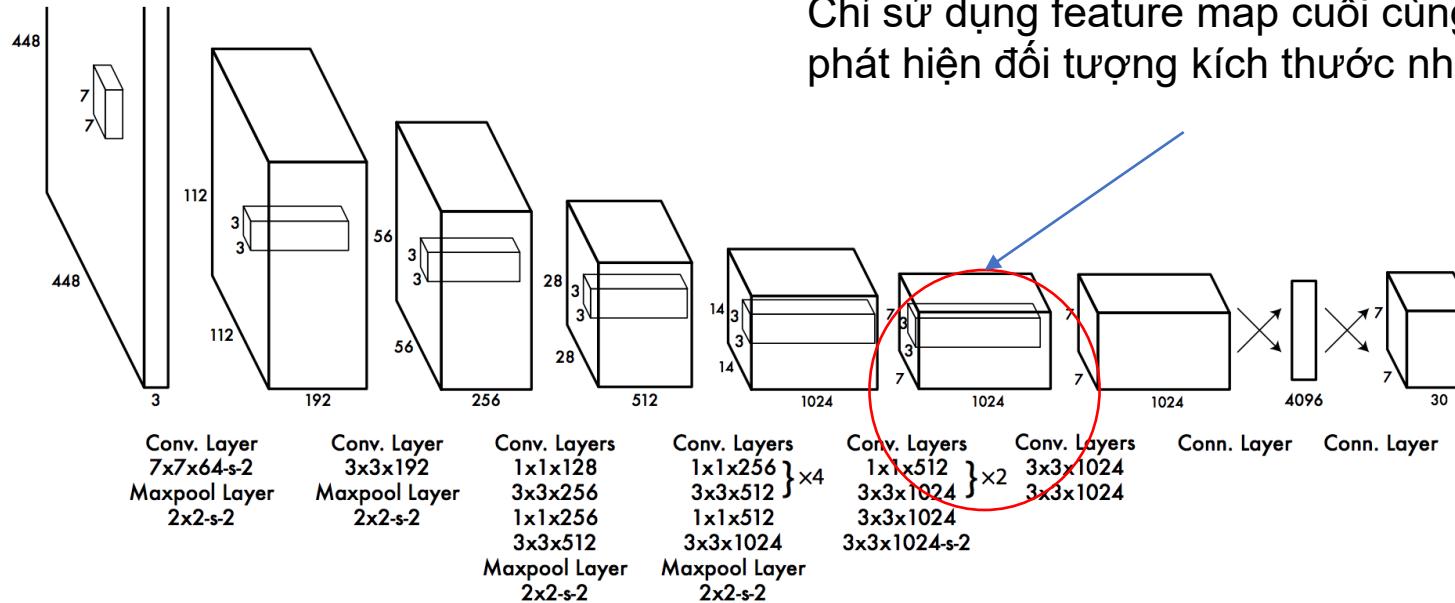
# YOLO v2

- 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

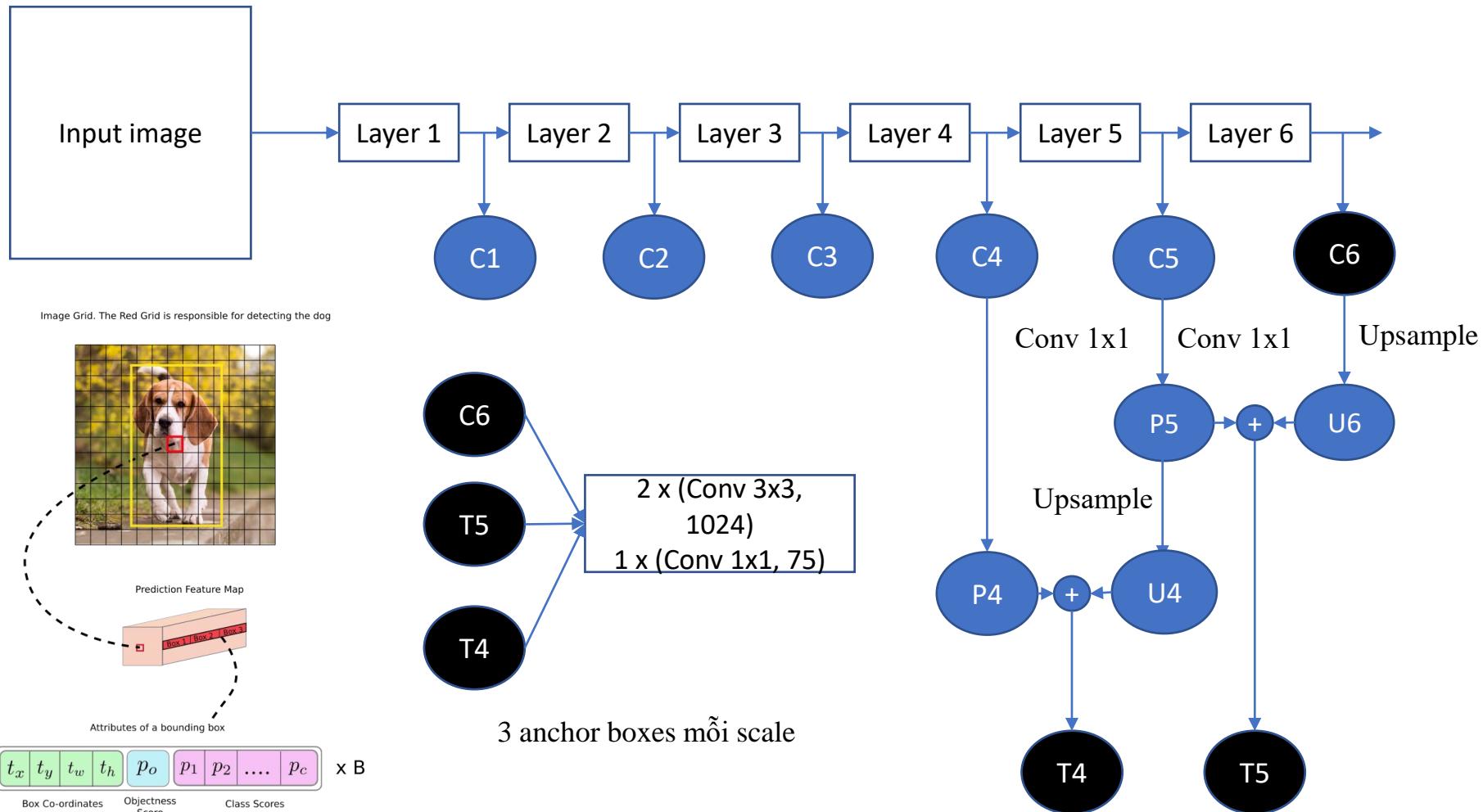


# YOLO v2

- Nhược điểm của YOLO v1 và v2:

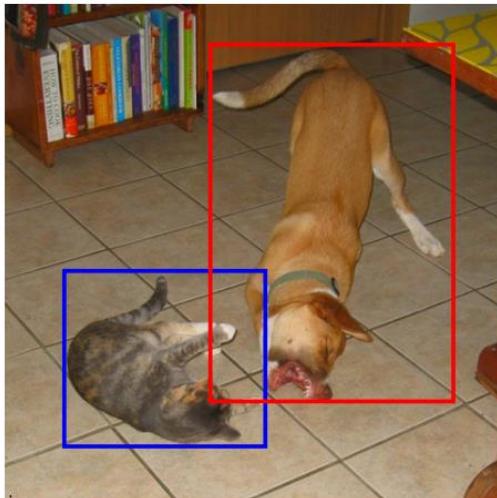


# YOLO v3

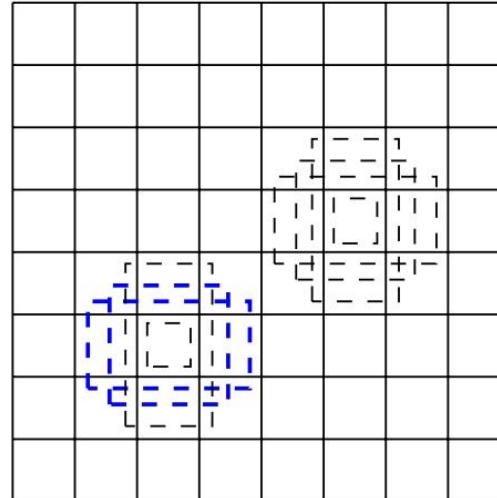


# SSD: Single Shot Detector

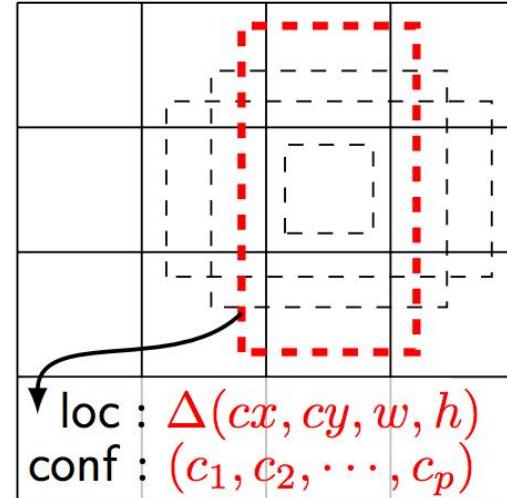
- Tương tự YOLO nhưng lưới box dày đặc hơn, có nhiều lưới với các kích thước box khác nhau
- Kiến trúc mạng backbone khác với YOLO
- Data augmentation + Hard negative mining



(a) Image with GT boxes



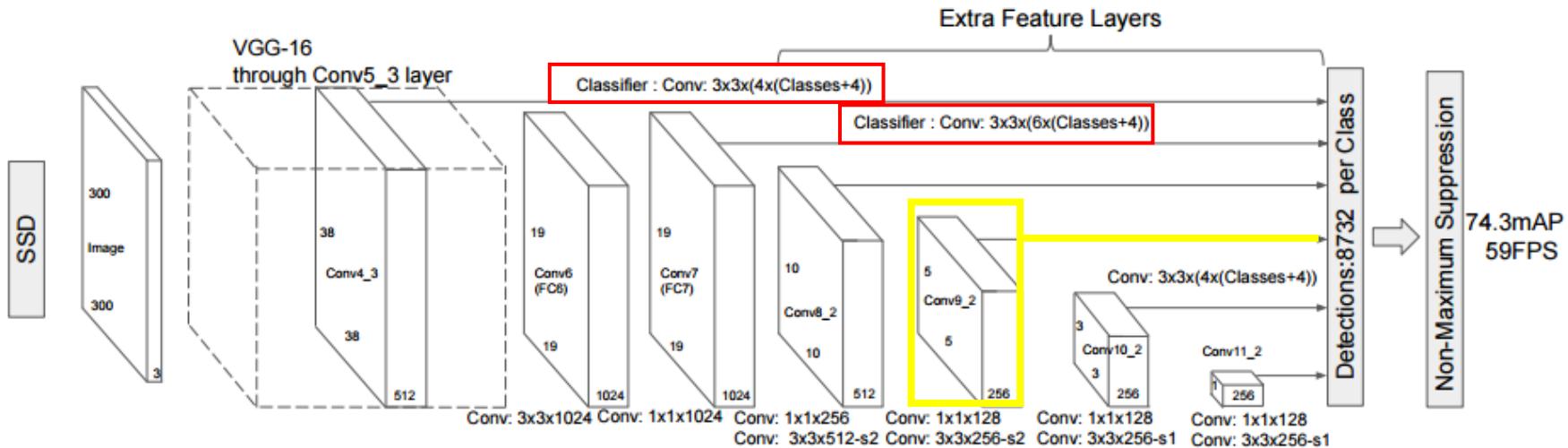
(b)  $8 \times 8$  feature map



(c)  $4 \times 4$  feature map

# SSD: Single Shot Detector

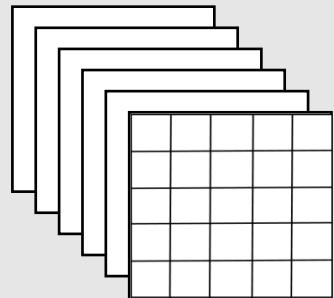
- Mạng backbone: VGG-16
- Thêm các lớp tích chập phụ phía sau các lớp của mạng backbone
- Phát hiện đối tượng ở nhiều mức khác nhau trong mạng (Multi-scale)



Liu et al. ECCV 2016.

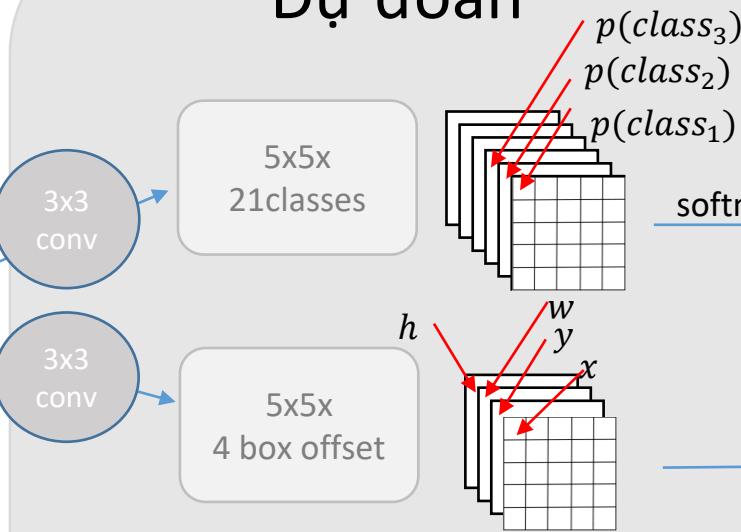
# SSD: Single Shot Detector

Feature map  
đầu vào



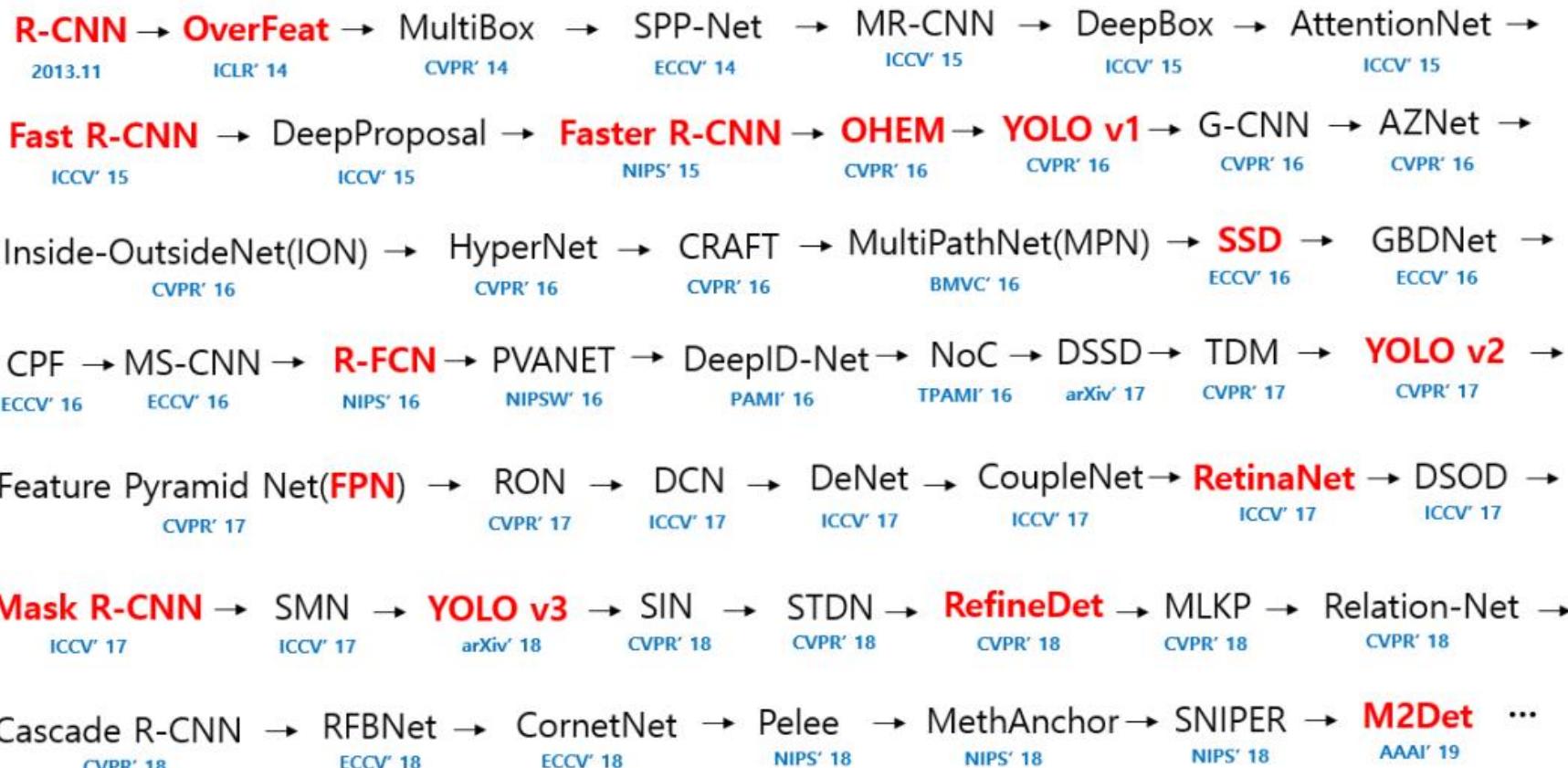
@5x5x256  
Feature map

Dự đoán



Hàm mục tiêu

$$L(x, c, l, g) = \frac{1}{N} (L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$



# One-stage vs two-stage

Faster and simpler

*one-stage object detector*

(dense sampling of object locations,  
scales, and aspect ratios)

YOLO

YOLO-v2

YOLO-v3

SSD

DSSD

MDCN

SqueezeNet

RetinaNet

RedefineDet

CornetNet

CenterNet

**EfficientDet**



More accurate

*two-stage object detector*

(proposal-driven mechanism)

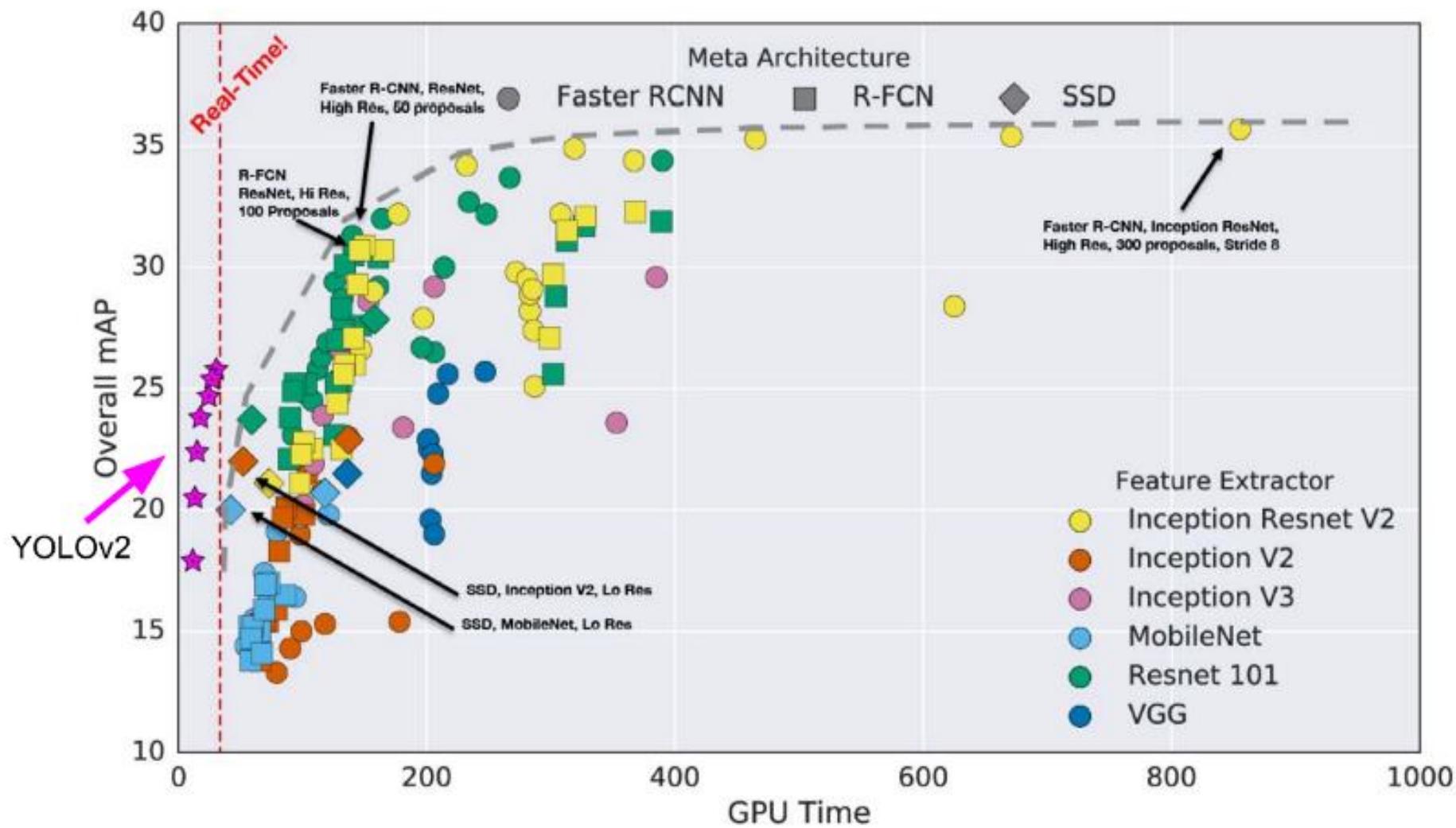
R-CNN

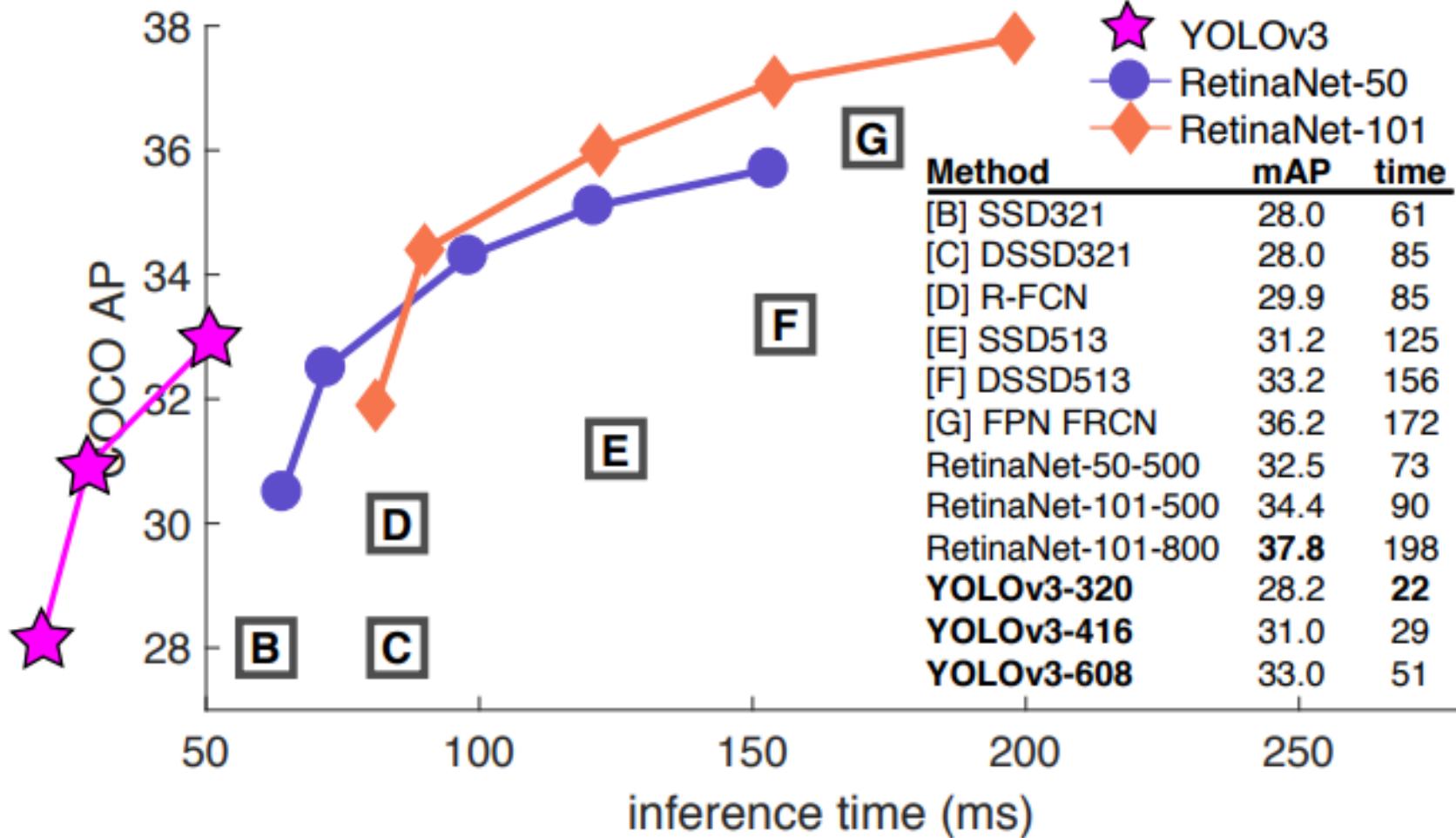
Fast R-CNN

Faster R-CNN

Feature Pyramid Network (FPN)

Mask R-CNN







25  
YEARS ANNIVERSARY  
**SOICT**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you  
for your  
attentions!**

