

Object Detection Overview

Naeemullah Khan
naeemullah.khan@kaust.edu.sa



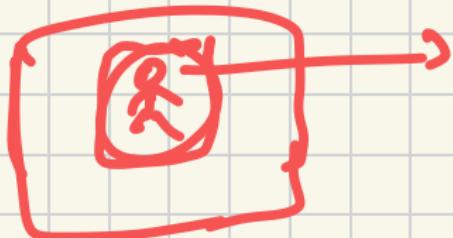
جامعة الملك عبدالله
للعلوم والتكنولوجيا
King Abdullah University of
Science and Technology

KAUST Academy
King Abdullah University of Science and Technology

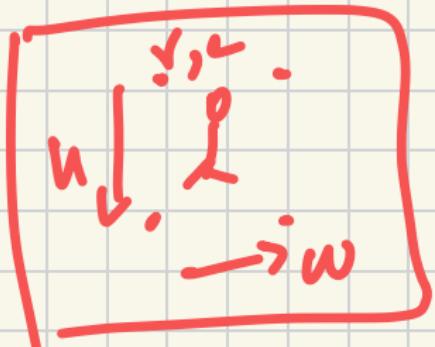
June 3, 2025

Table of Contents

1. Challenges
2. Intersection over Union (IoU)
3. Single Object Detection
4. Multiple Objects Sliding Window
5. Region Proposal Networks (RPN)
6. R-CNN
7. Fast R-CNN
8. Faster R-CNN
9. YOLO
10. Summary



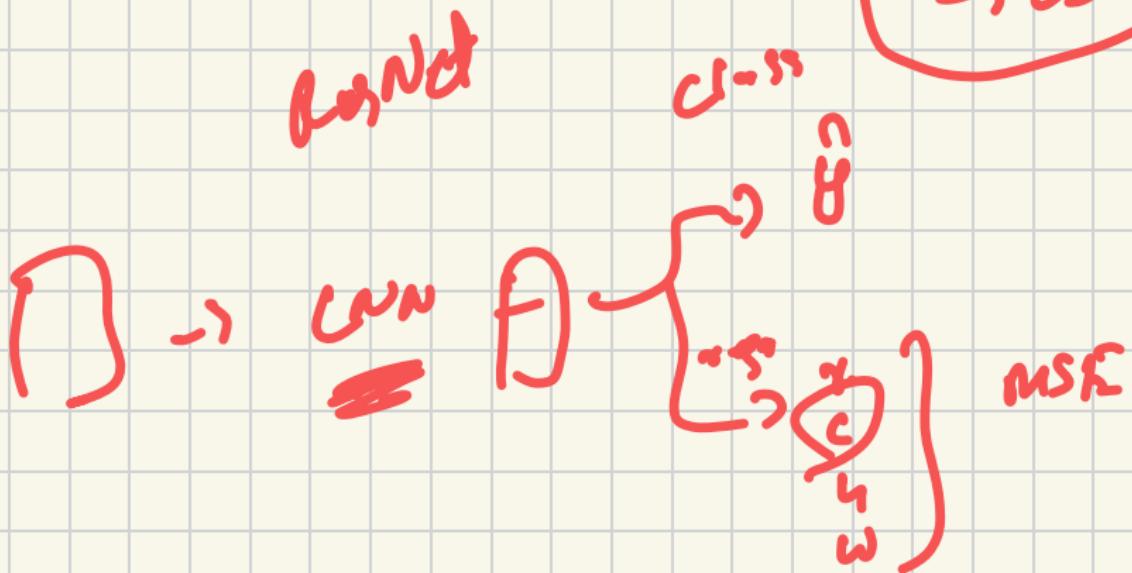
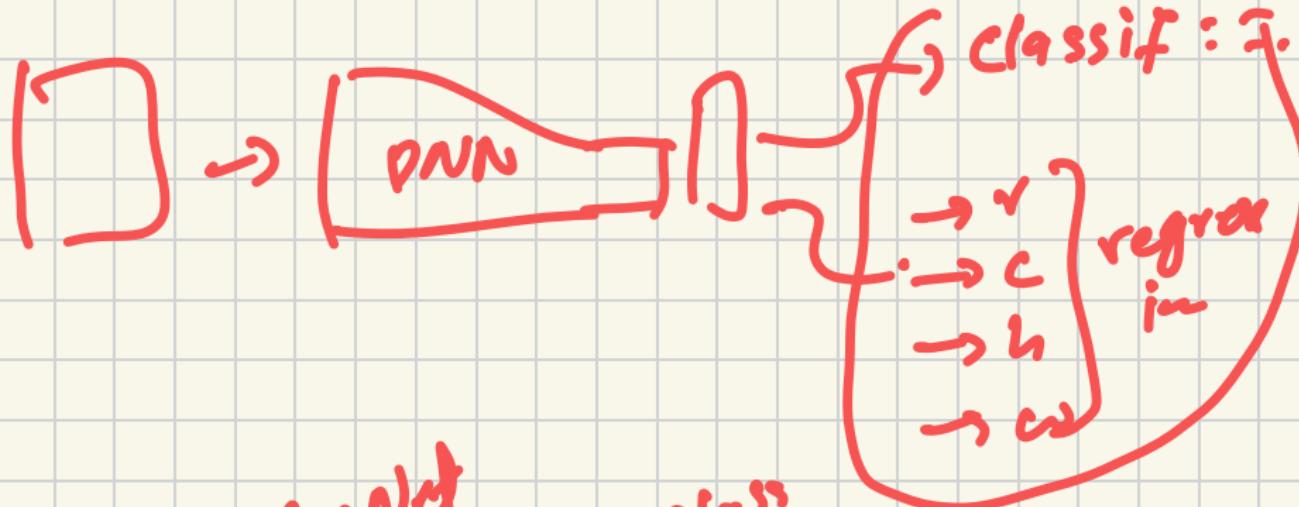
GT
what



four coordinates

x_1, x_2





$$L_{cls} = - \sum_{l=1}^n \hat{y}_l \log \hat{y}_l$$

$$L_{MSE} = \frac{1}{2} \left[(\hat{y} - y)^2 + (c - l)^2 + (\hat{u} - h)^2 + (\hat{v} - w)^2 \right]$$

weights

$$L = L_{cls} + \lambda (L_{obj}) +$$

Object Detection: Challenges

Object detection is a complex task in computer vision that involves identifying and localizing objects within images.

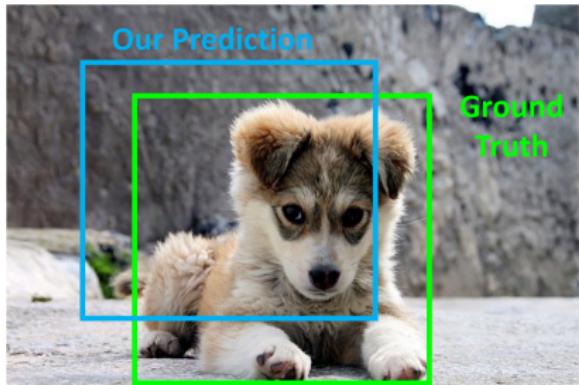
Unlike image classification, which assigns a single label to an entire image, object detection requires both identifying the objects present and determining their locations within the image.

This task presents several challenges that make it more difficult than classification:

- ▶ **Multiple outputs:** Need to output variable numbers of objects per image
- ▶ **Multiple types of output:** Need to predict "what" (category label) as well as "where" (bounding box)
- ▶ **Large images:** Classification works at 224x224; need higher resolution for detection, often $\sim 800 \times 600$

Comparing Boxes: Intersection over Union (IoU)

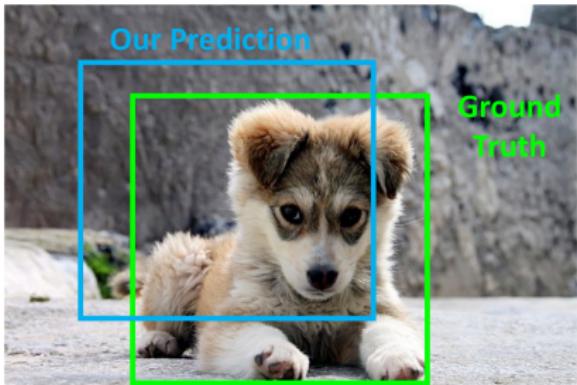
- ▶ How can we compare our prediction to the ground-truth box?



Comparing Boxes: Intersection over Union (IoU)

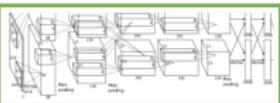
- ▶ How can we compare our prediction to the ground-truth box?
- ▶ **Intersection over Union (IoU)**
(Also called "Jaccard similarity" or "Jaccard index"):

$$\frac{\text{Area of Intersection}}{\text{Area of Union}}$$



Detecting a Single Object

Often pretrained
on ImageNet
(Transfer learning)



This image is CC0 public domain

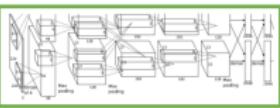
Treat localization as a
regression problem!

Vector:
4096

Detecting a Single Object (cont.)



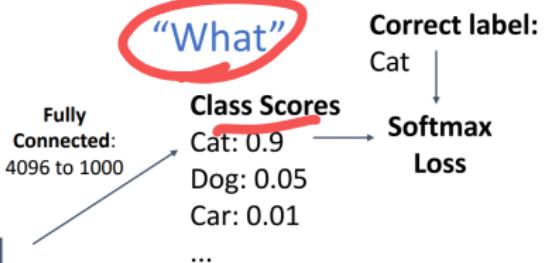
Often pretrained
on ImageNet
(Transfer learning)



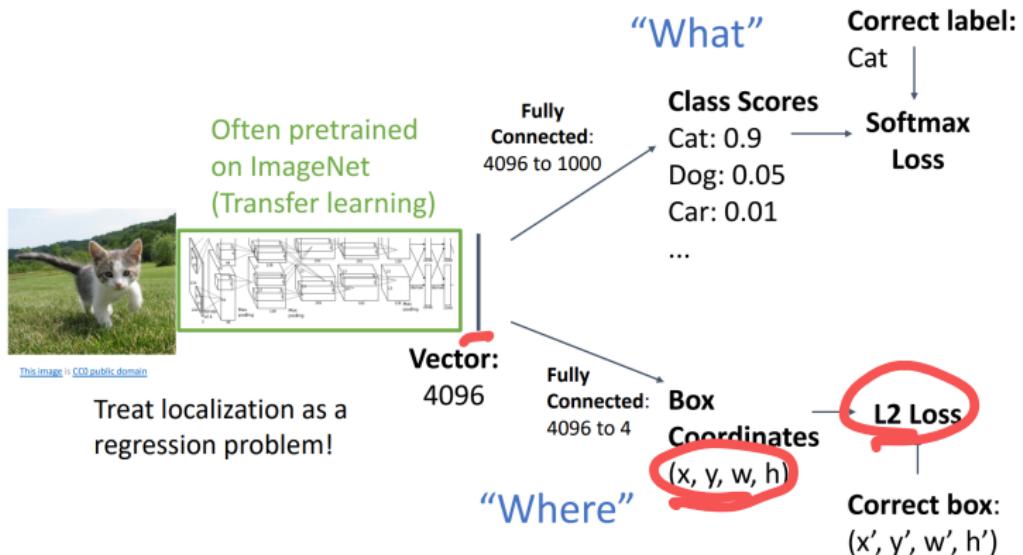
This image is CC0 public domain

Treat localization as a
regression problem!

Vector:
4096



Detecting a Single Object (cont.)



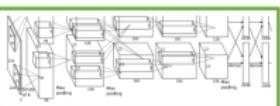
Treat localization as a regression problem!

Detecting a Single Object (cont.)



This image is CC0 public domain

Often pretrained
on ImageNet
(Transfer learning)



Treat localization as a
regression problem!

Vector:
4096

Fully
Connected:
4096 to 1000

“What”

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Fully
Connected:
4096 to 4

Box
Coordinates
(x, y, w, h)

“Where”

Correct label:
Cat

Softmax
Loss

Multitask
Loss

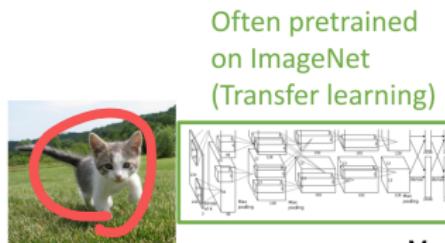
Weighted
Sum

$$\bullet \quad L = L_{cls} + \lambda L_{reg}$$

L2 Loss

Correct box:
(x', y', w', h')

Detecting a Single Object (cont.)



Often pretrained
on ImageNet
(Transfer learning)

Treat localization as a
regression problem!

Problem: Images can have
more than one object!

Vector:
4096

Fully
Connected:
4096 to 1000

“What”

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Fully
Connected:
4096 to 4

“Where”

Box
Coordinates
(x, y, w, h)

Correct label:
Cat

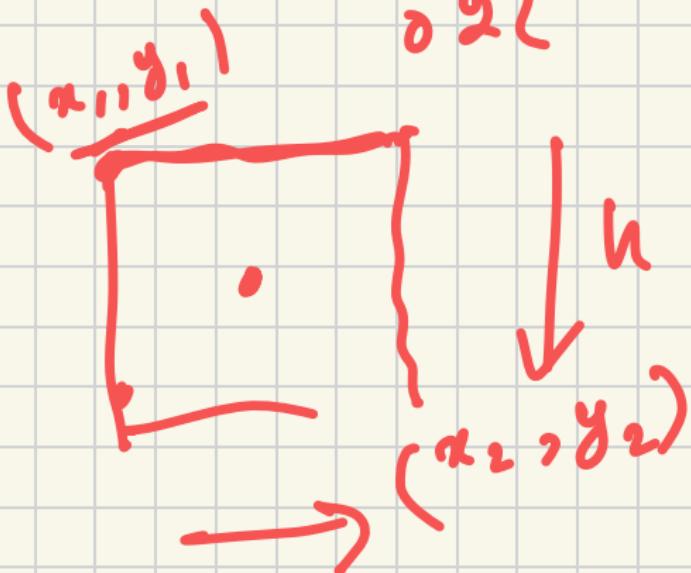
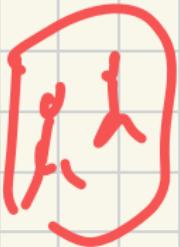
Softmax
Loss

Multitask
Loss

Weighted
Sum

$$L = L_{cls} + \lambda L_{reg}$$

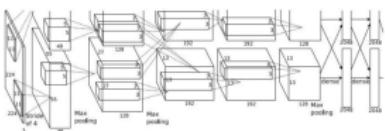
L2 Loss
Correct box:
(x', y', w', h')



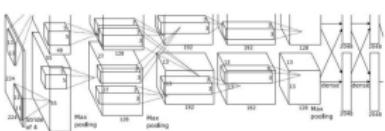
01 [
02]

]
}

Multiple Objects



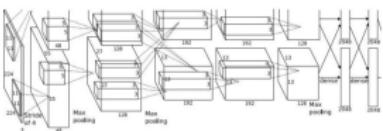
CAT: (x, y, w, h)



DOG: (x, y, w, h)

DOG: (x, y, w, h)

CAT: (x, y, w, h)

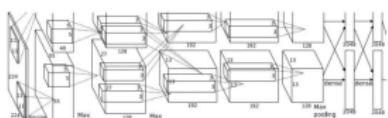


DUCK: (x, y, w, h)

DUCK: (x, y, w, h)

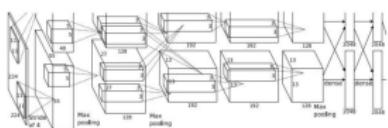
....

Multiple Objects (cont.)

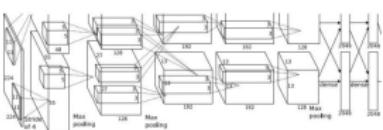


Each image needs a
different number of outputs!

CAT: (x, y, w, h) 4 numbers



DOG: (x, y, w, h)
DOG: (x, y, w, h) 12 numbers
CAT: (x, y, w, h)

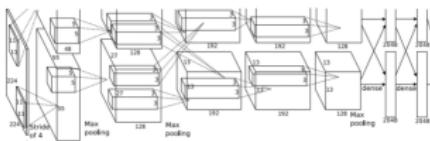


DUCK: (x, y, w, h) Many
DUCK: (x, y, w, h) numbers!
....

Detecting Multiple Objects: Sliding Window

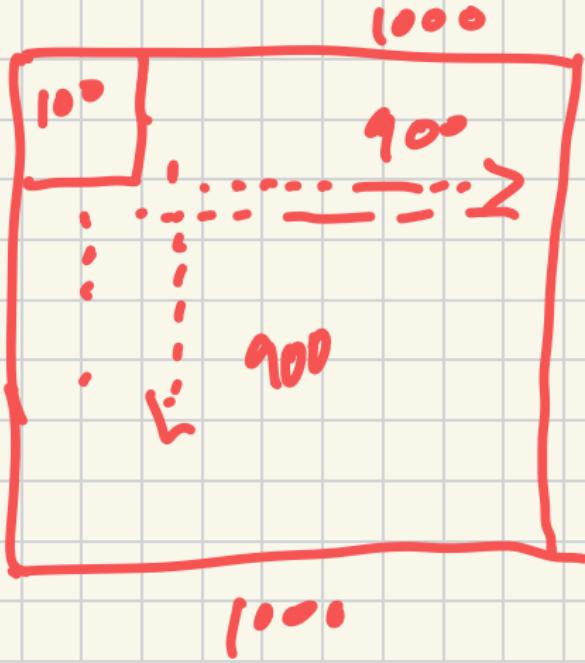


Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO ↘
Cat? NO

Background? YES

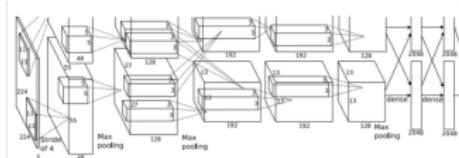


$$900 \times 900$$

Detecting Multiple Objects: Sliding Window (cont.)



Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

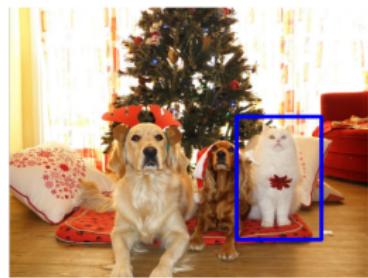


Dog? YES

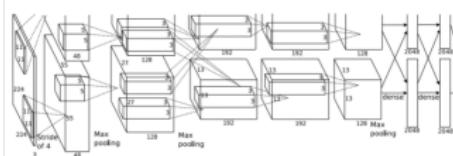
Cat? NO

Background? NO

Detecting Multiple Objects: Sliding Window (cont.)



Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO

Cat? YES

Background? NO

Detecting Multiple Objects: Sliding Window (cont.)

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

Question: How many possible boxes are there in an image of size $H \times W$?



Detecting Multiple Objects: Sliding Window (cont.)



Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

Question: How many possible boxes are there in an image of size $H \times W$?

Consider a box of size $h \times w$:

Possible x positions: $W - w + 1$

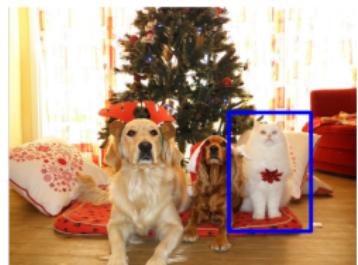
Possible y positions: $H - h + 1$

Possible positions: ?

$$(W - w + 1) * (H - h + 1)$$

700 x 900

Detecting Multiple Objects: Sliding Window (cont.)



Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

Question: How many possible boxes are there in an image of size $H \times W$?

Consider a box of size $h \times w$:

Possible x positions: $W - w + 1$

Possible y positions: $H - h + 1$

Possible positions:

$$(W - w + 1) * (H - h + 1)$$

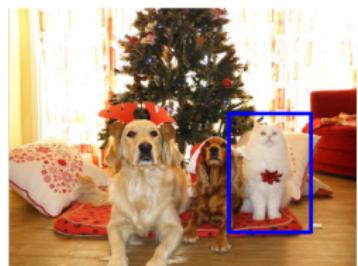
Total possible boxes:

$$\sum_{h=1}^H \sum_{w=1}^W (W - w + 1)(H - h + 1)$$

$$= \frac{H(H + 1)}{2} \frac{W(W + 1)}{2}$$



Detecting Multiple Objects: Sliding Window (cont.)



Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

800 x 600 image
Has ~58M boxes!
No way we can evaluate them all

Question: How many possible boxes are there in an image of size $H \times W$?

Consider a box of size $h \times w$:

Possible x positions: $W - w + 1$

Possible y positions: $H - h + 1$

Possible positions:

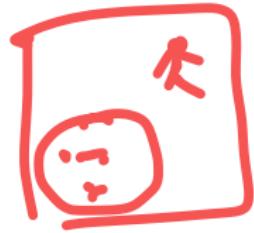
$$(W - w + 1) * (H - h + 1)$$

Total possible boxes:

$$\sum_{h=1}^H \sum_{w=1}^W (W - w + 1)(H - h + 1)$$

$$= \frac{H(H+1)}{2} \frac{W(W+1)}{2}$$

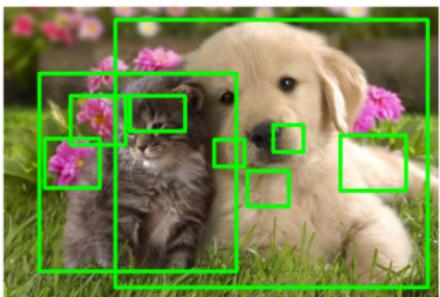
Detecting Multiple Objects: Sliding Window (cont.)



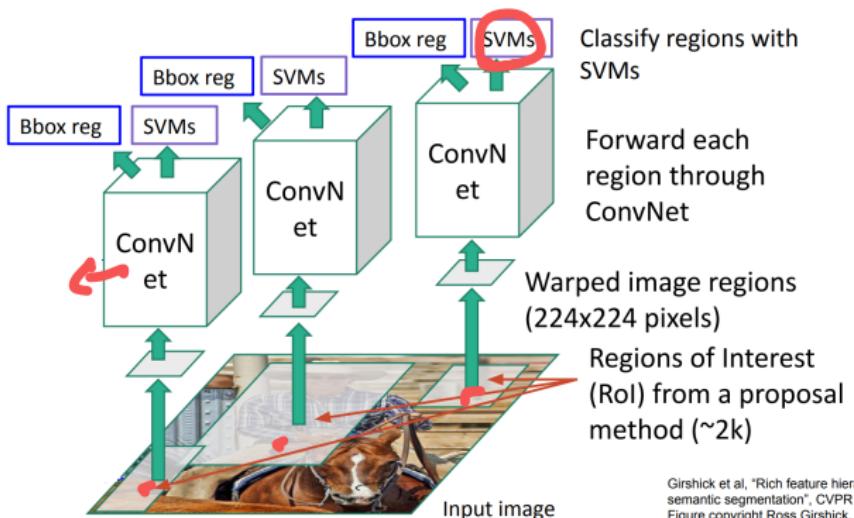
preprocessing.

Region Proposal

- ▶ Find a small set of boxes that are likely to cover all objects
- ▶ Often based on heuristics: e.g. look for “blob-like” image regions
- ▶ Relatively fast to run; e.g. Selective Search gives 2000 region proposals in a few seconds on CPU



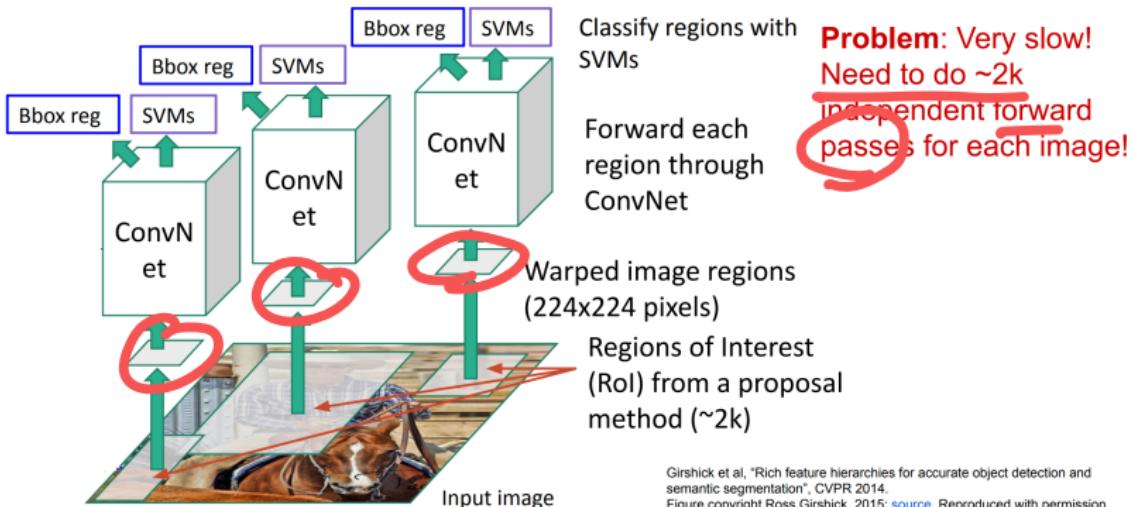
Predict “corrections” to the RoI: 4 numbers: (dx, dy, dw, dh)



Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

R-CNN (cont.)

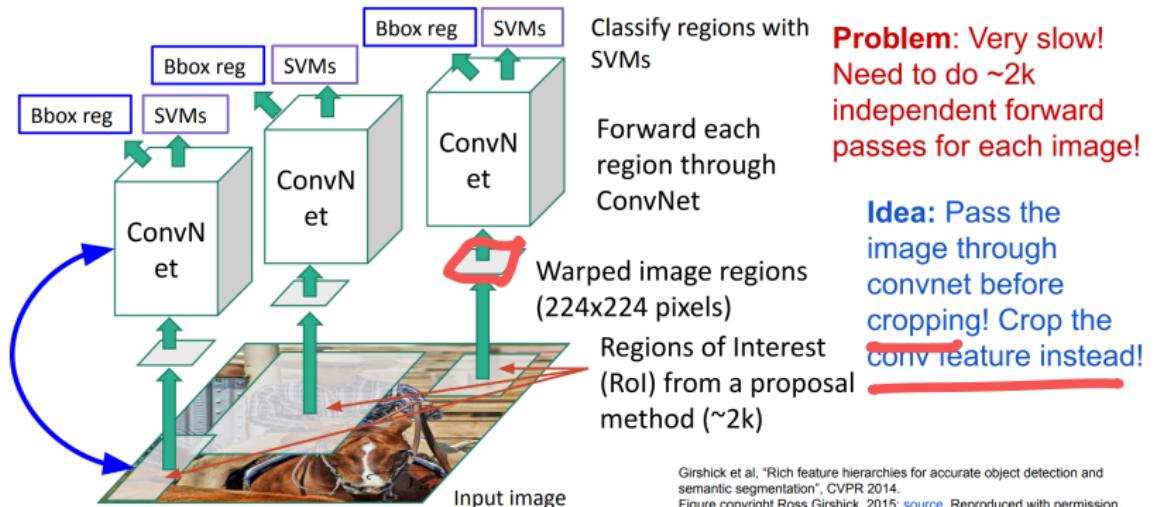
Predict “corrections” to the RoI: 4 numbers: (dx, dy, dw, dh)



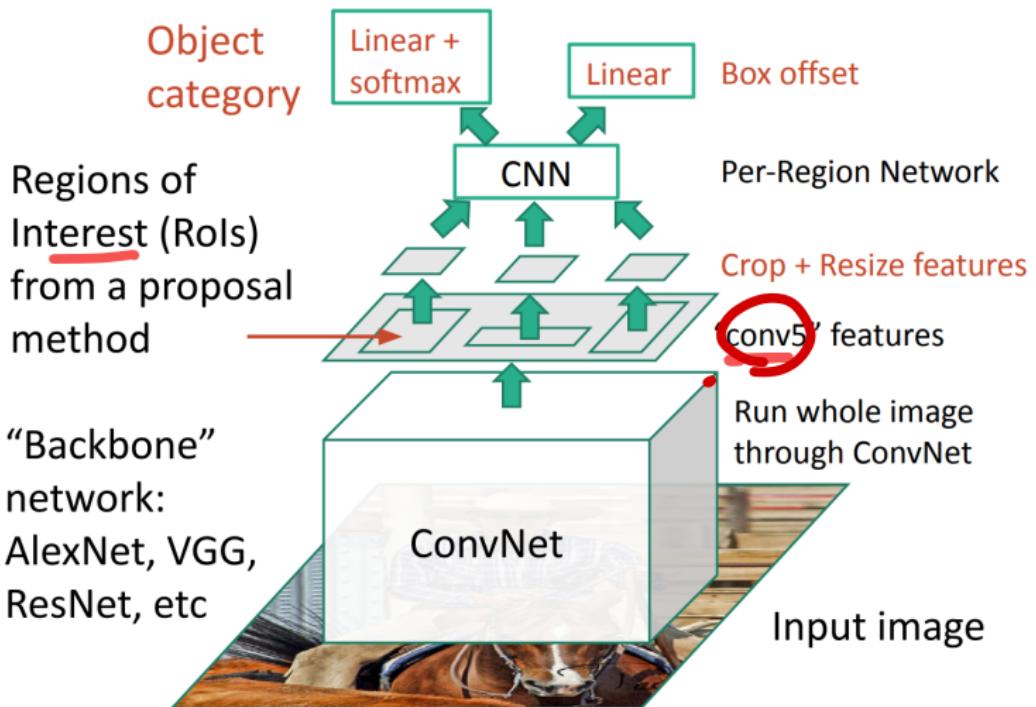
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

R-CNN (cont.)

Predict “corrections” to the RoI: 4 numbers: (dx, dy, dw, dh)



Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

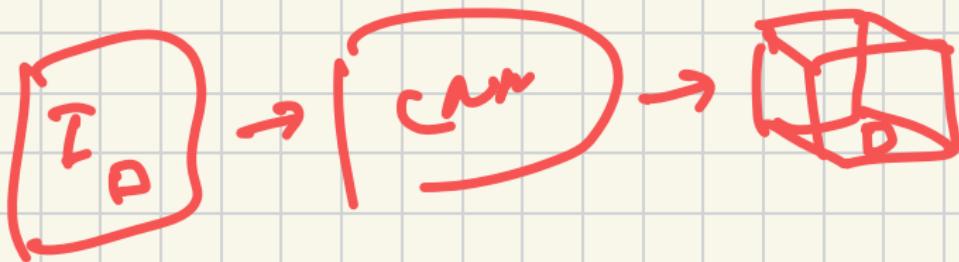
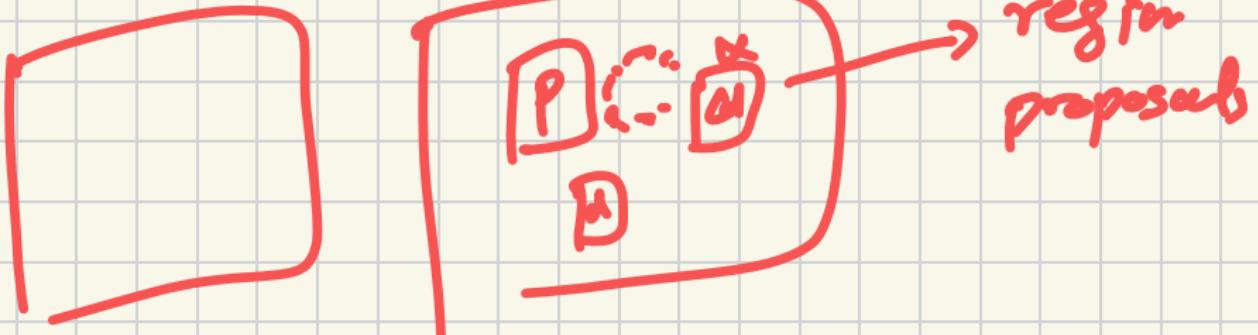


Girshick, "Fast R-CNN", ICCV 2015. Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

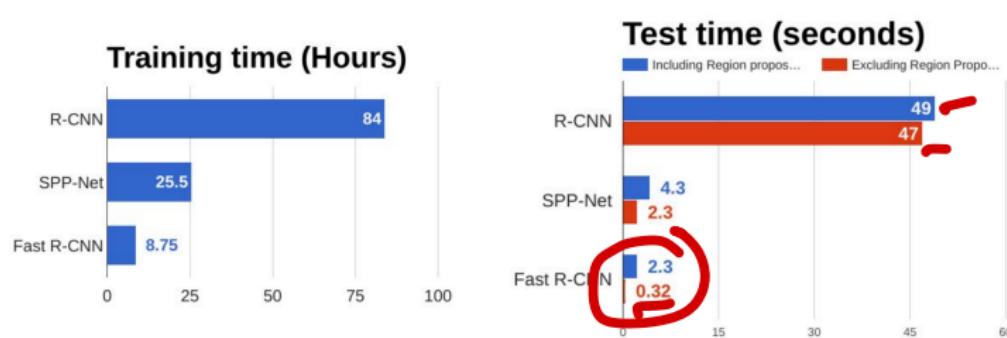
1. D

$\xrightarrow{64, 64} (64)$ $\xrightarrow{32, 32} \dots$

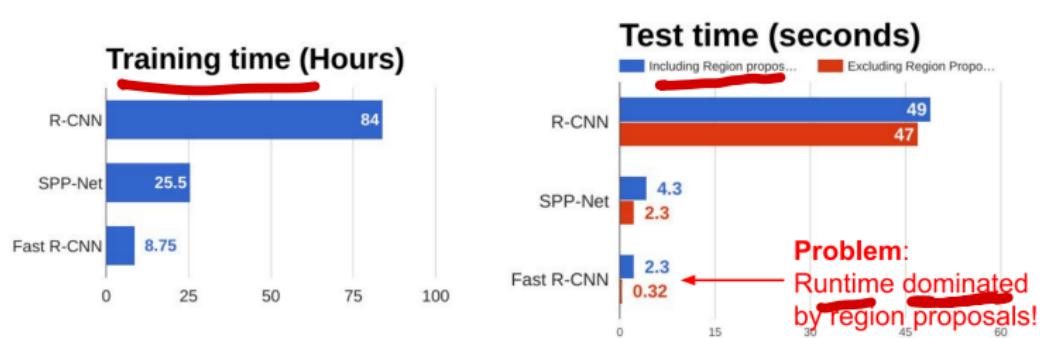
$128, 128$
 $64, 37$



Fast R-CNN (cont.)



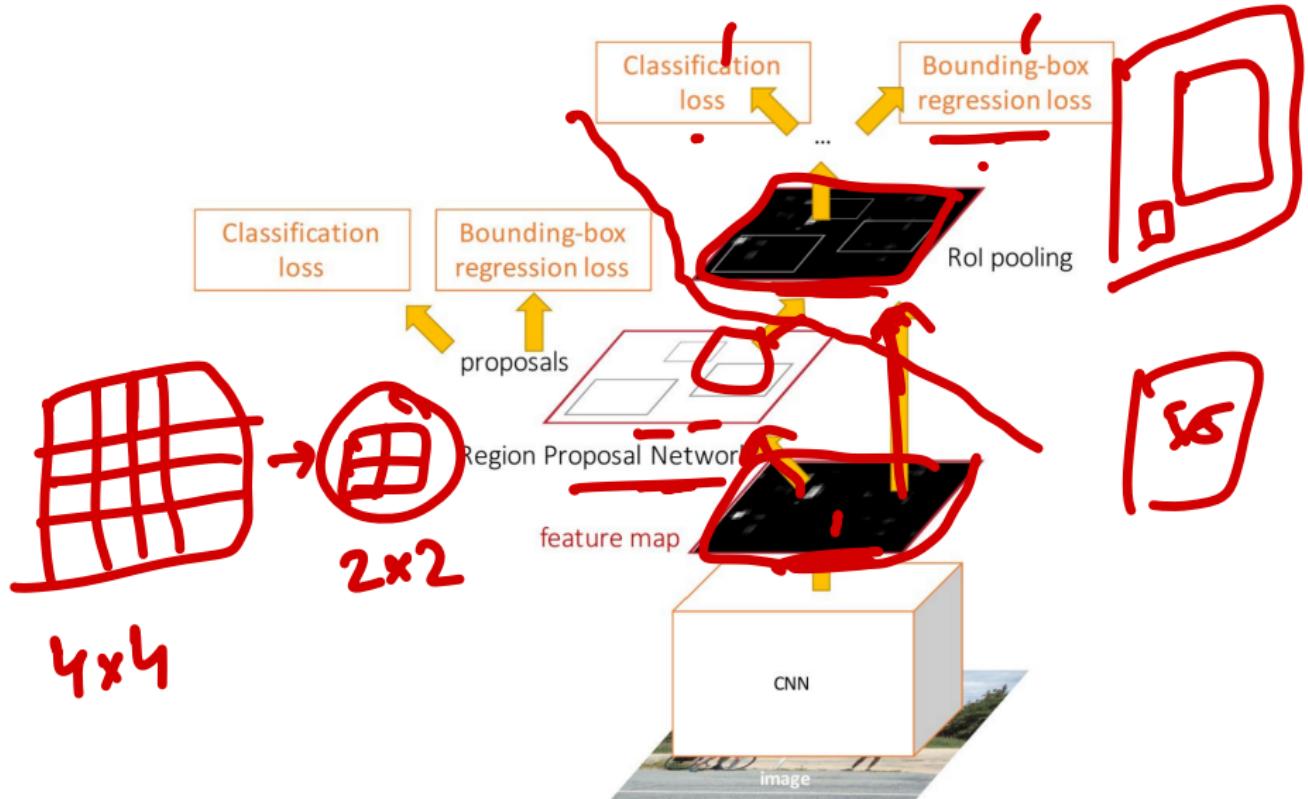
Fast R-CNN (cont.)

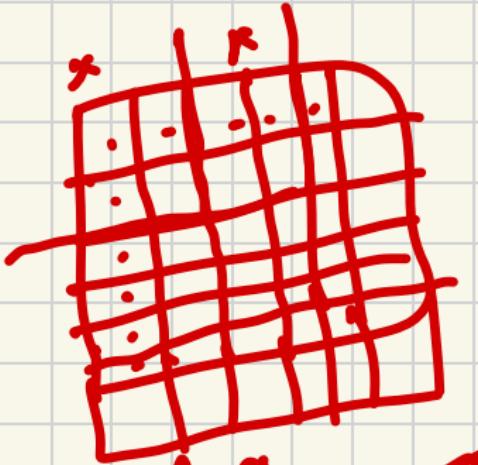


- ▶ Make CNN do proposals!
- ▶ Insert Region Proposal Network (RPN) to predict proposals from features

- ▶ Make CNN do proposals!
- ▶ Insert Region Proposal Network (RPN) to predict proposals from features
- ▶ Jointly train on 4 losses:
 - **RPN classification:** anchor box is object / not an object
 - **RPN regression:** predict transform from anchor box to proposal box
 - **Object classification:** classify proposals as background / object class
 - **Object regression:** predict transform from proposal box to object box

Faster R-CNN

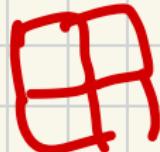




7×6



2×2

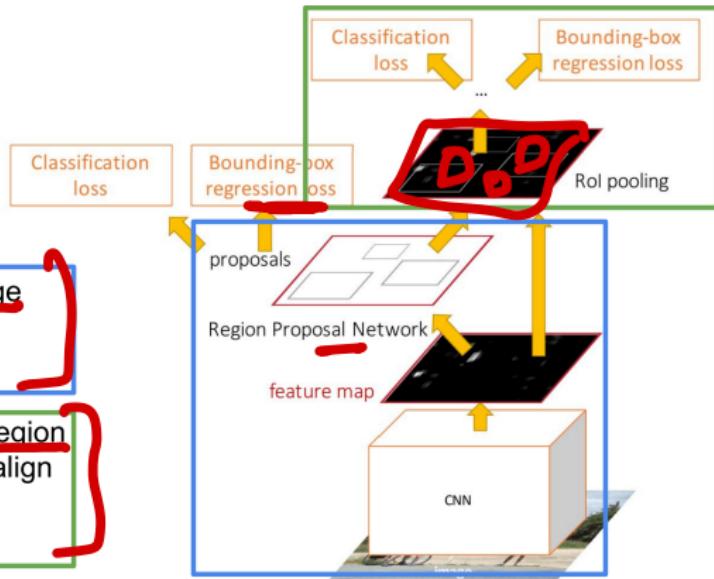


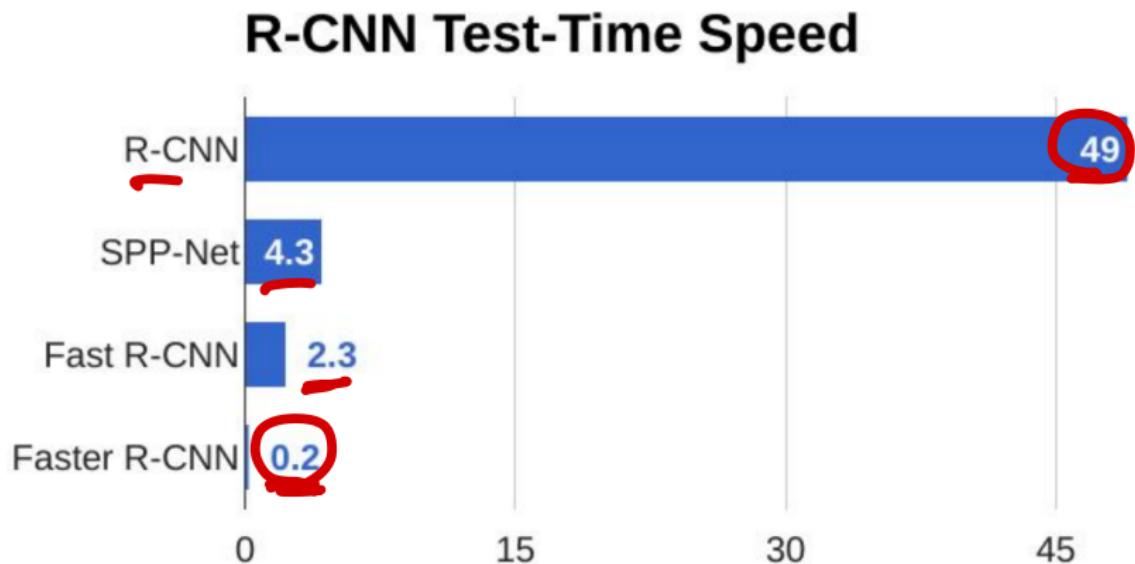
Faster R-CNN: Make CNN do proposals!

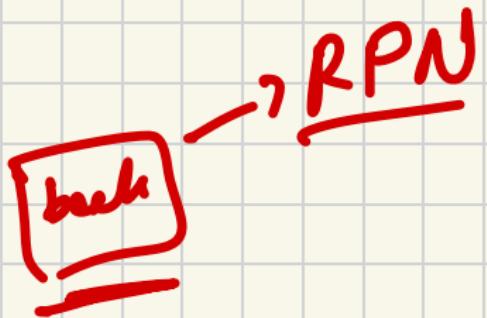
Faster R-CNN is a
Two-stage object detector

First stage: Run once per image
- Backbone network
- Region proposal network

Second stage: Run once per region
- Crop features: RoI pool / align
- Predict object class
- Prediction bbox offset







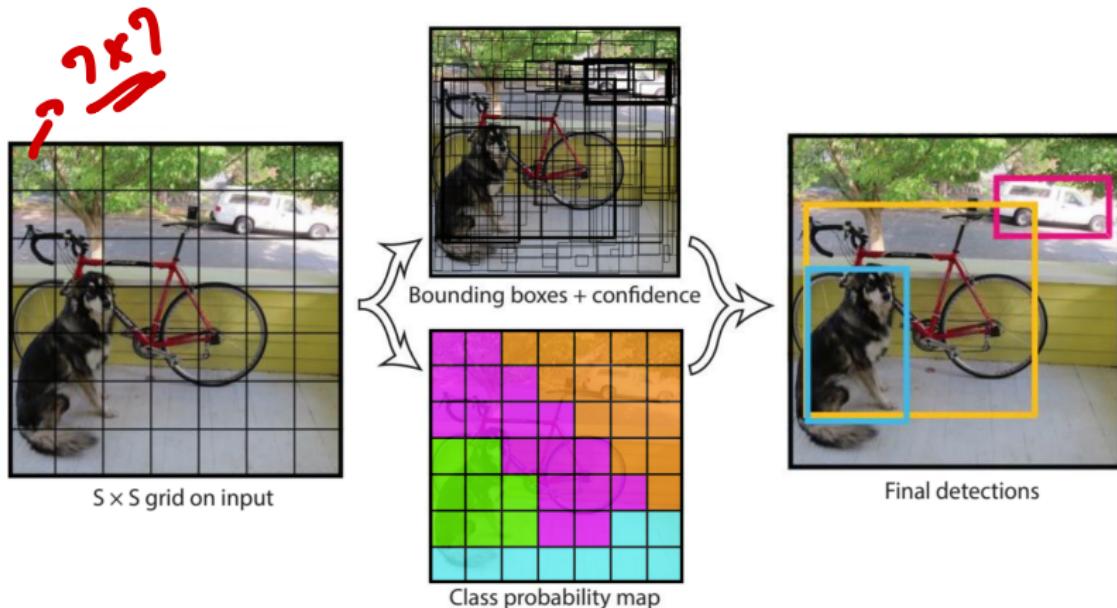
Single Shot Object Detection



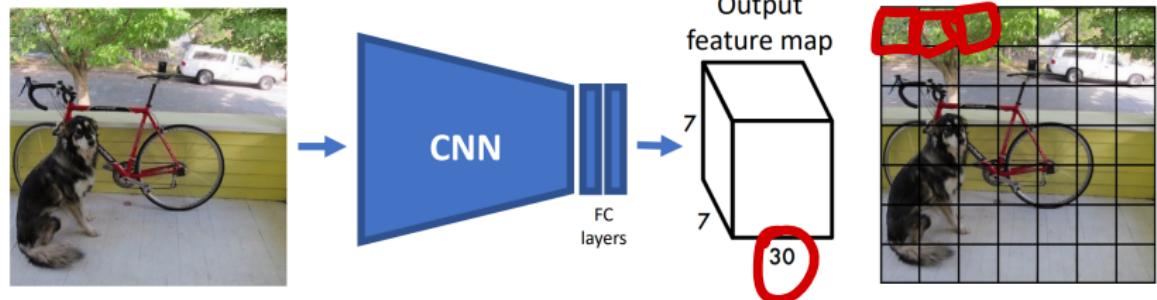
Single Shot:
SSD, YOLO ...

Fast
High false rate

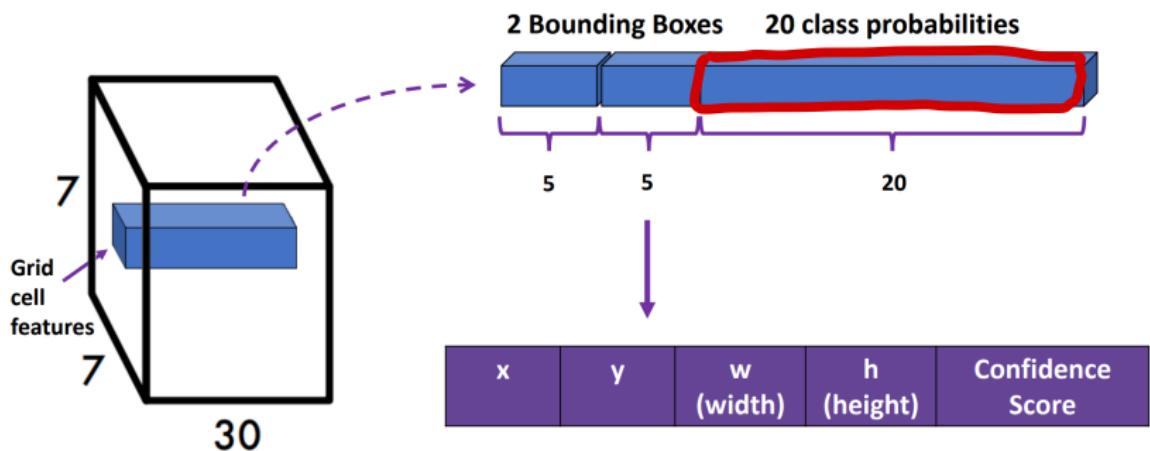
YOLO - Overview

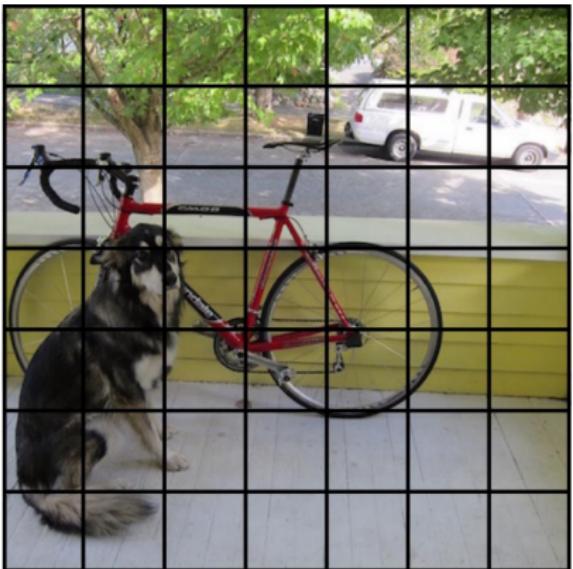


YOLO - Overview

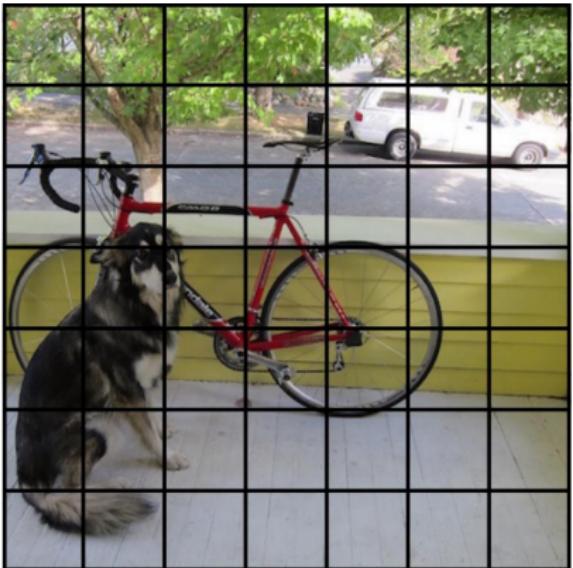


YOLO - Overview



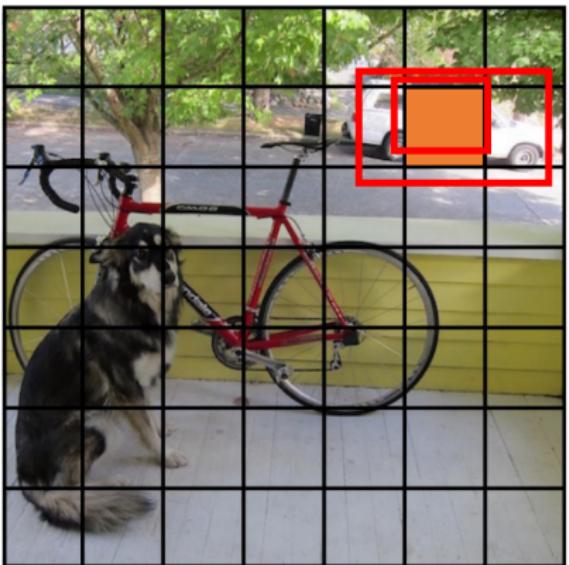


Each cell predicts



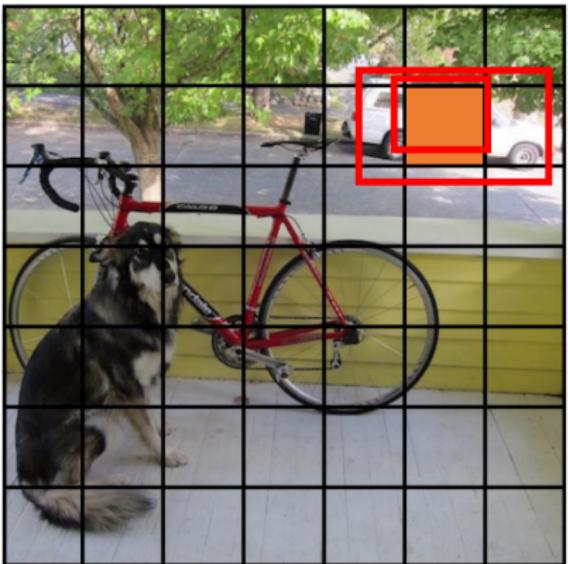
Each cell predicts

- ▶ $B = 2$ bounding boxes
 $(x, y, w, h) + \text{confidence score}$



Each cell predicts

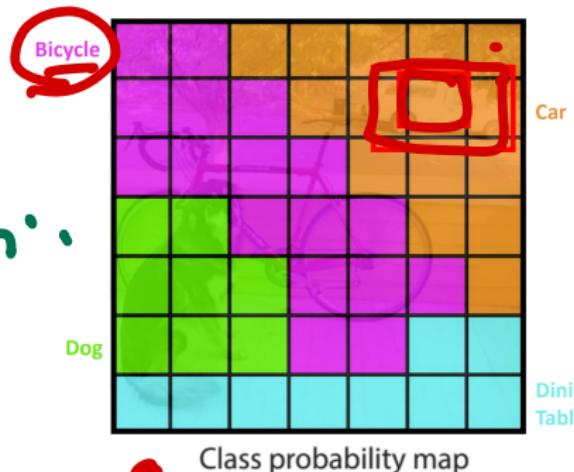
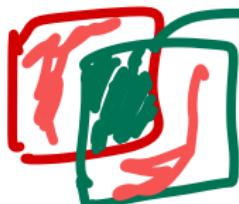
- ▶ $B = 2$ bounding boxes
 $(x, y, w, h) +$ confidence score
 - ▶ $C = 20$ class probabilities
-



Each cell predicts

- ▶ $B = 2$ bounding boxes (x, y, w, h) + confidence score
- ▶ $C = 20$ class probabilities

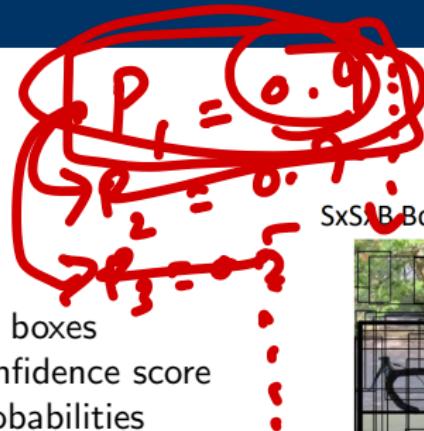
↑ intersection



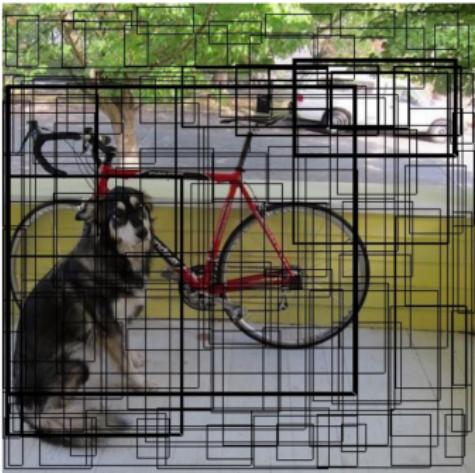
$$P_1 = 0.9, P_2 = 0.7, P_3 = 0.5$$

Each cell predicts

- ▶ $B = 2$ bounding boxes (x, y, w, h) + confidence score
- ▶ $C = 20$ class probabilities



SxS Bounding-Boxes ($S=7, B=2 \rightarrow 96$ Bboxs)



$S \times S$ grid on input

Each cell predicts

- ▶ $B = 2$ bounding boxes (x, y, w, h) + confidence score
- ▶ $C = 20$ class probabilities

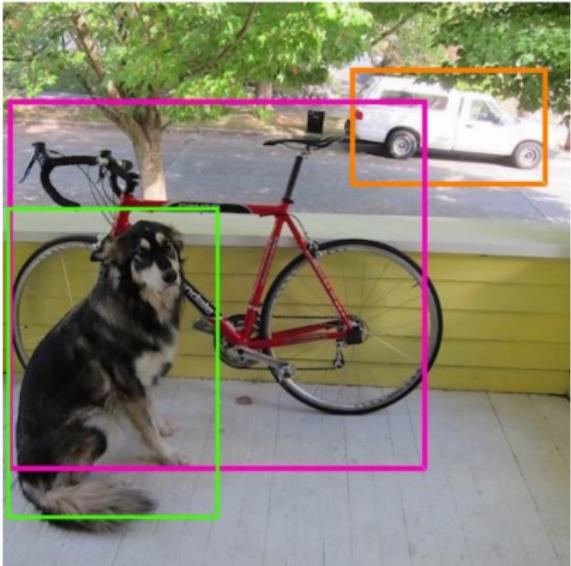
$$\begin{aligned} p_3 &= 0.9 \\ p_9 &= 0.3 \end{aligned}$$

SxSxB Bounding-Boxes ($S=7, B=2 \rightarrow 96$ Bboxes)



Each cell predicts

- ▶ $B = 2$ bounding boxes
 (x, y, w, h) + confidence score
 - ▶ $C = 20$ class probabilities
-
- ▶ Apply Non-Maximum Suppression



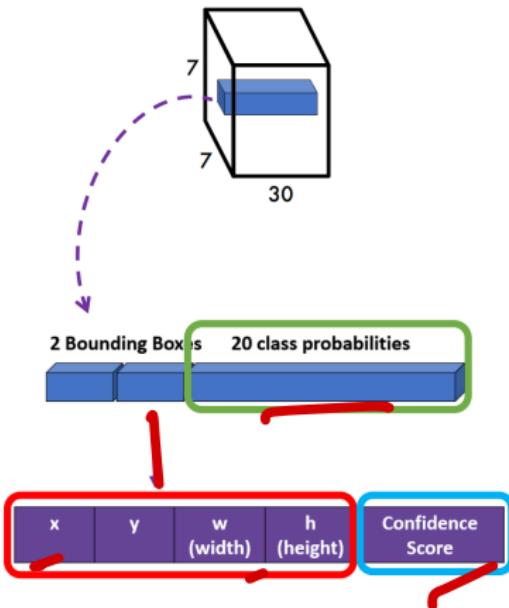
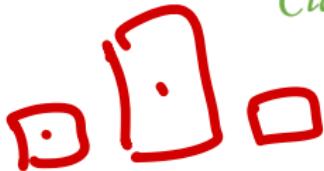
YOLO - Loss Function

YOLO – Loss function

$$\mathcal{L} = \mathcal{L}_{Localization\ Loss}$$

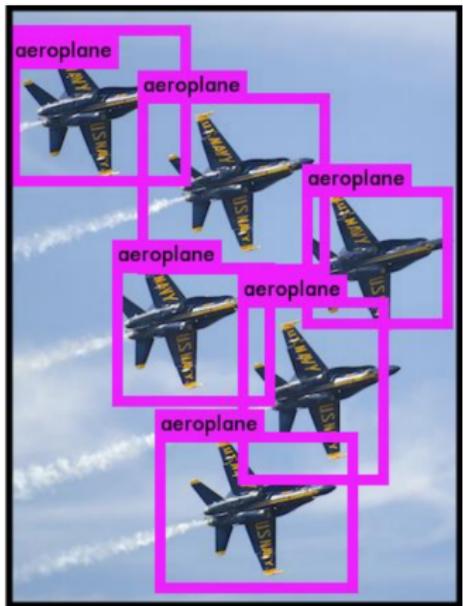
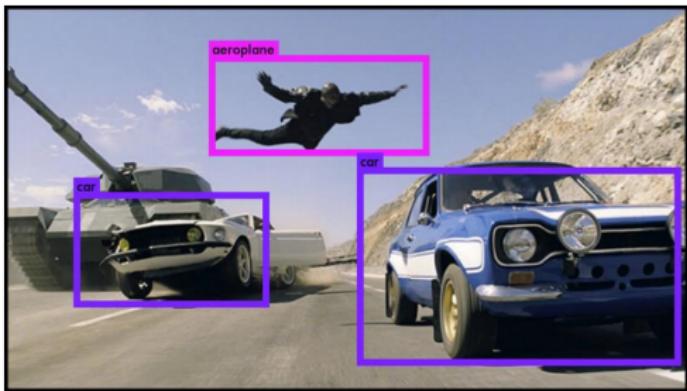
$$+ \mathcal{L}_{Confidence\ Loss}$$

$$+ \mathcal{L}_{Classification\ Loss}$$

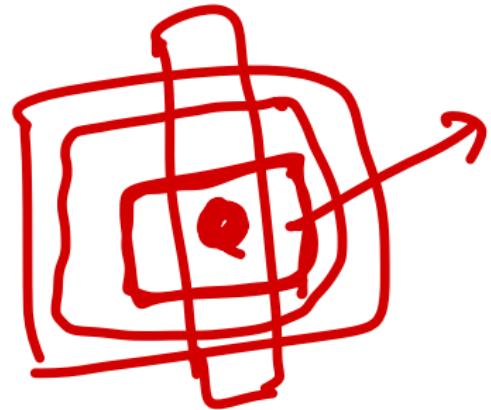


YOLO - Benefits

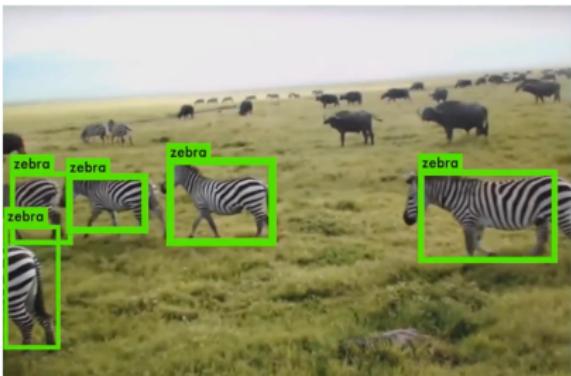
- ▶ Fast. Good for real-time processing
- ▶ End-to-end training



YOLO - Limitations

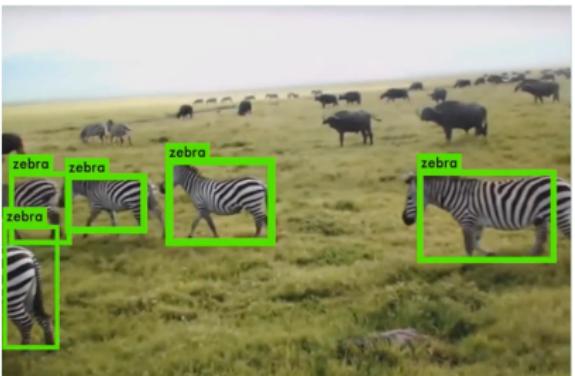


selected car.



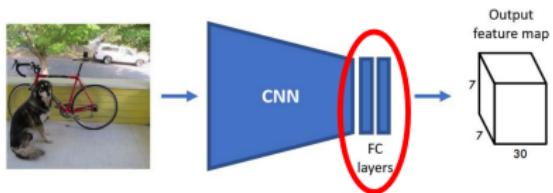
YOLO - Limitations

- ▶ Difficult to detect small objects
- ▶ Coarse predictions



YOLO - Limitations

- ▶ Difficult to detect small objects
- ▶ Coarse predictions
- ▶ Fixed input size



YOLO - Limitations

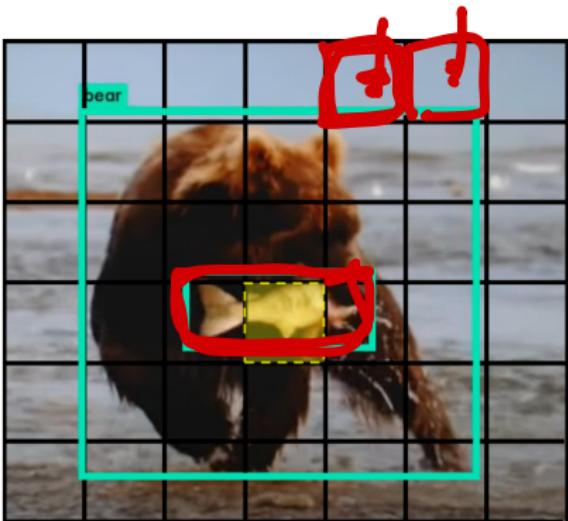
- ▶ Difficult to detect small objects
- ▶ Coarse predictions
- ▶ Fixed input size
- ▶ A grid cell can predict only one class

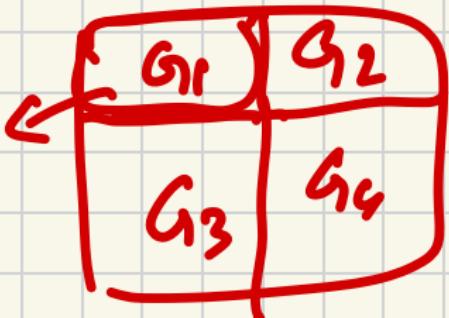


YOLO - Limitations

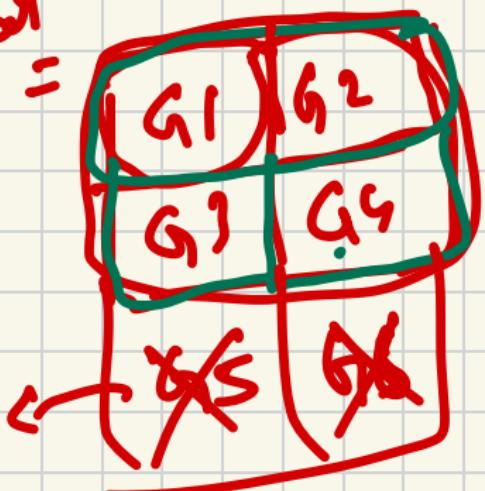


- ▶ Difficult to detect small objects
 - ▶ Coarse predictions
 - ▶ Fixed input size
 - ▶ A grid cell can predict only one class
 - ▶ Solutions:
 - Remove fc layers!
 - Predict class per bbox (not per cell)





$$P_1 G_5^{\text{out}} =$$



dog¹

$$P_1 G_1 = 0.9$$

$$\rightarrow P_1 G_2 = 0.8$$

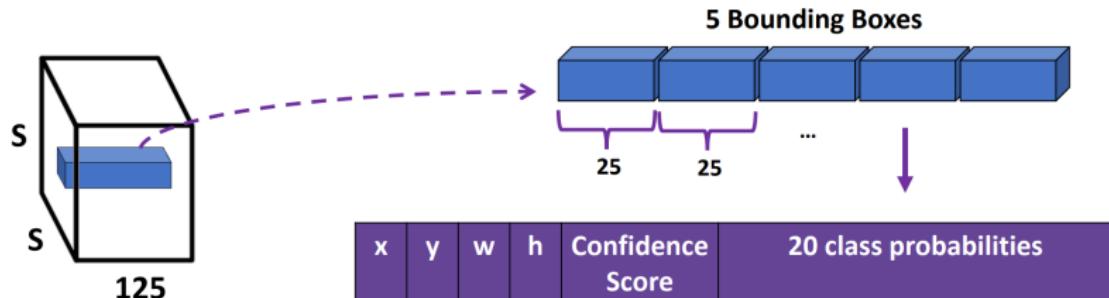
$$\rightarrow \text{dog} \rightarrow P_1 G_3 = 0.8$$

$$P_1 G_4 = \underline{0.7}$$

$$\rightarrow P_1 G_5 = \underline{0.2}$$

$$P_1 G_6 = \underline{0.3}$$

- ▶ Removed fully connected layers
- ▶ A grid cell predicts class probabilities for each box



► YOLOv3

- J. Redmon, A. Farhadi. Yolov3: An incremental improvement, 2018

► YOLOv4

- A. Bochkovskiy, C. Wang, H. Liao. Yolov4: Optimal speed and accuracy of object detection (Feb. 2020)

► YOLOv5

- YOLOv5 by ultralytics (June 2020)

► PP-YOLO

- X. Long, K. Deng, G. Wang, Y. Zhang, Q. Dang, Y. Gao, H. Shen, J. Ren, S. Han, E. Ding, S. Wen. Pp-yolo: An effective and efficient implementation of object detector (June 2020)

► PP-YOLOv2 (2021)

- J. X. Huang, X. Wang, W. Lv, X. Bai, X. Long, K. Deng, Q. Dang, S. Han, Q. Liu, X. Hu, D. Yu, Y. Ma, O. Yoshie. PP-YOLOv2: A Practical Object Detector (2021)

Object Detection: Summary

- ▶ Object detection is a complex task that involves identifying and localizing objects within images.
- ▶ Key challenges include:
 - Variability in object appearance, scale, and orientation.
 - Occlusion and clutter in images.
 - Real-time processing requirements for applications like autonomous driving.
- ▶ Techniques range from traditional methods like sliding windows to advanced deep learning approaches like YOLO and Faster R-CNN.
- ▶ Intersection over Union (IoU) is a critical metric for evaluating the accuracy of object detection models.
- ▶ YOLO (You Only Look Once) is a popular real-time object detection algorithm that processes images in a single pass, making it efficient for applications requiring speed.

These slides have been adapted from

- ▶ Fei-Fei Li, Yunzhu Li & Ruohan Gao, Stanford CS231n: Deep Learning for Computer Vision
- ▶ Assaf Shocher, Shai Bagon, Meirav Galun & Tali Dekel, WAIC DL4CV Deep Learning for Computer Vision: Fundamentals and Applications
- ▶ Justin Johnson, UMich EECS 498.008/598.008: Deep Learning for Computer Vision

Credits

Dr. Prashant Aparajeya

Computer Vision Scientist — Director(AISimply Ltd)

p.aparajeya@aisimply.uk

This project benefited from external collaboration, and we acknowledge their contribution with gratitude.