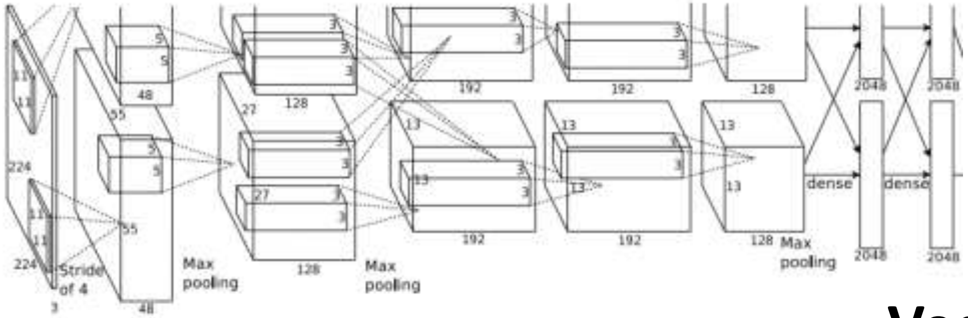
تشخیص اشیاء
Object Detection

Alireza AkhavanPour

Akhavanpour.ir
CLASS.VISION

So far: Image Classification



Vector:
4096

Fully-Connected:
4096 to 1000

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

Where is the object?

1-simple-regression-train.ipynb

Where is the object?

2-simple-regression-inference.ipynb

Where is the object?
What about class names?

3-object-classification-and-localization.ipynb

Where is the object?
What about class names?

4-object-classification-and-localization-inference.ipynb

Computer Vision Tasks

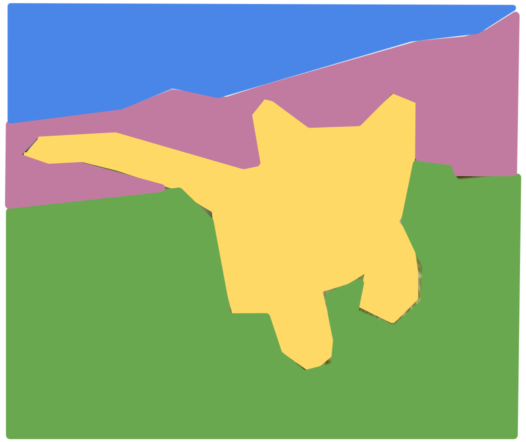
Classification	Semantic Segmentation	Object Detection	Instance Segmentation

Classification

CAT

No spatial extent

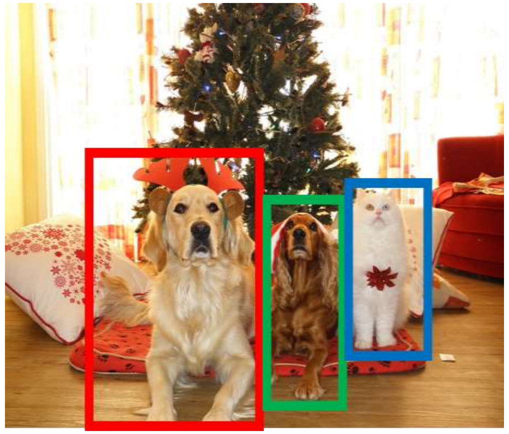
Semantic Segmentation



GRASS, CAT, TREE, SKY

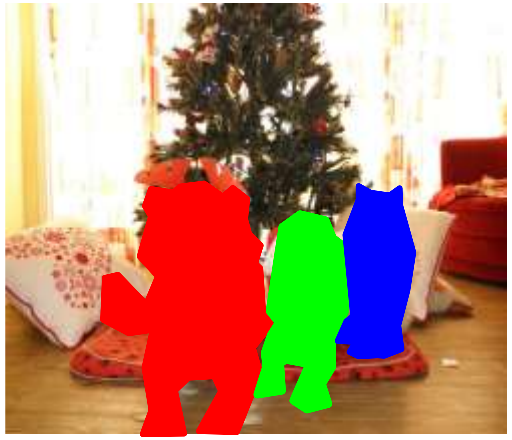
No objects, just pixels

Object Detection



DOG, DOG, CAT

Instance Segmentation



DOG, DOG, CAT

Multiple Objects

Today: Object Detection

Classification

CAT

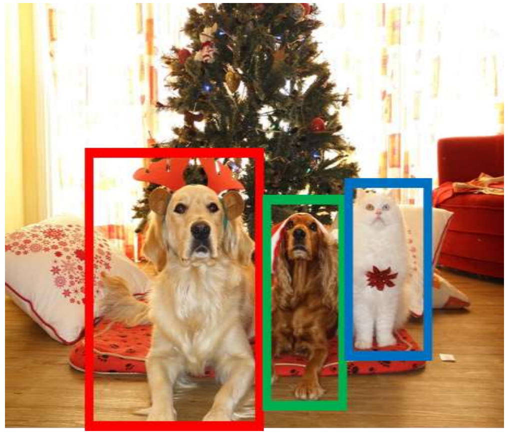
No spatial extent

Semantic Segmentation

GRASS, CAT, TREE, SKY

No objects, just pixels

Object Detection



DOG, DOG, CAT

Instance Segmentation

DOG, DOG, CAT

Multiple Objects

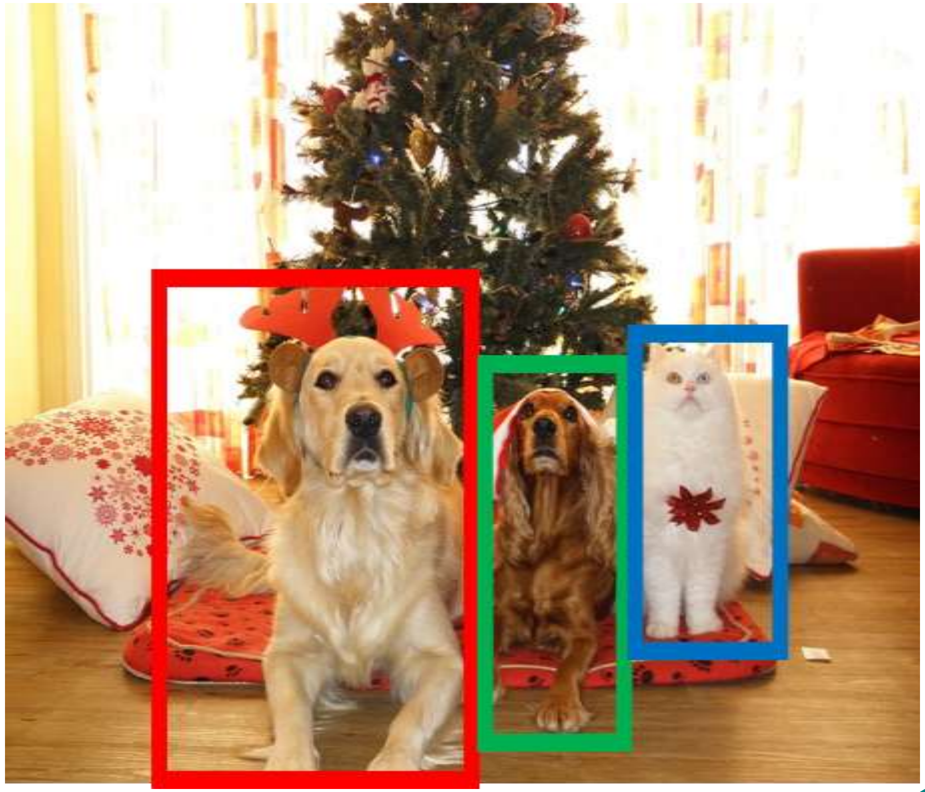
This image is CC0 public domain

Object Detection: Task Definition

Input: Single RGB Image

Output: A set of detected objects;
For each object predict:

1. Category label (from fixed, known set of categories)
2. Bounding box (four numbers: x, y, width, height)

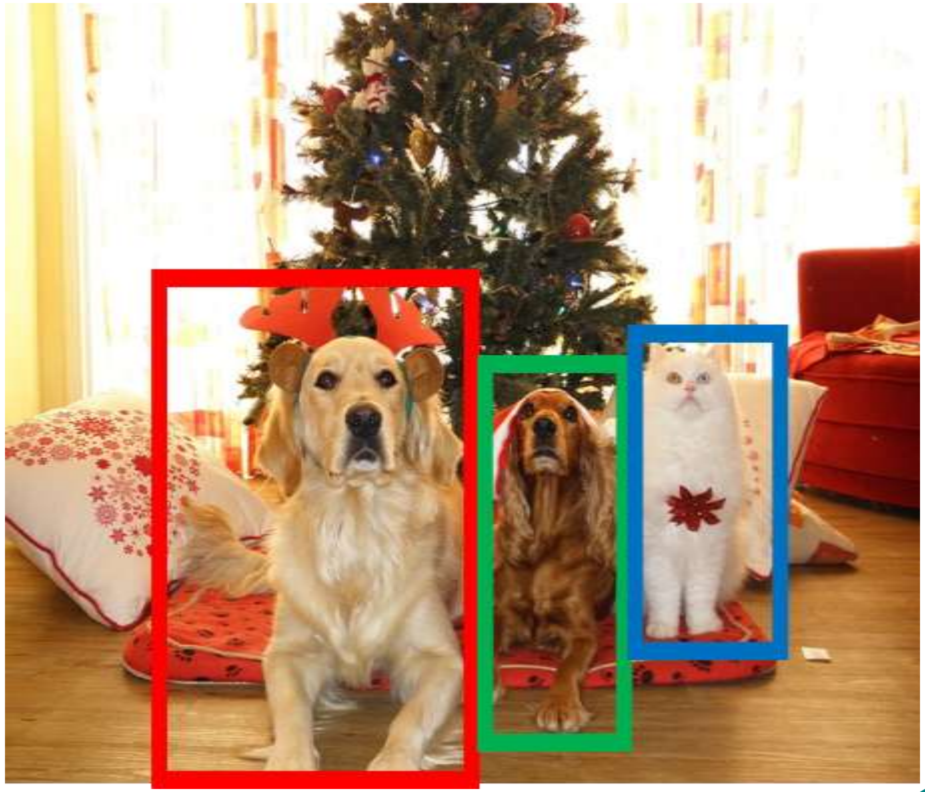


Object Detection: Challenges

- **Multiple outputs**:
  Need to output variable numbers of objects per image

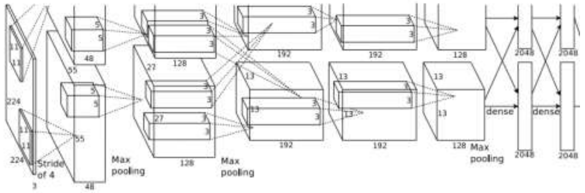
- **Multiple types of output**:
  Need to predict ”what” (category label) as well as “where” (bounding box)

- **Large images**:
  Classification works at 224x224; need higher resolution for detection, often ~800x600



Detecting a single object

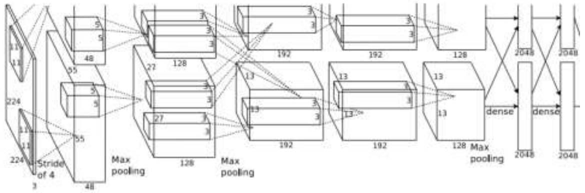
This image is CC0 public domain



Vector:
4096

Detecting a single object

This image is CC0 public domain



Vector: 4096

Fully Connected: 4096 to 1000

Class Scores

Cat: 0.9

Dog: 0.05

Car: 0.01

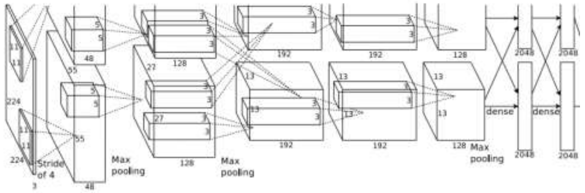
...

Correct label: Cat

Softmax Loss

Detecting a single object

This image is CC0 public domain



Vector:
4096

“What”

Fully Connected:
4096 to 1000

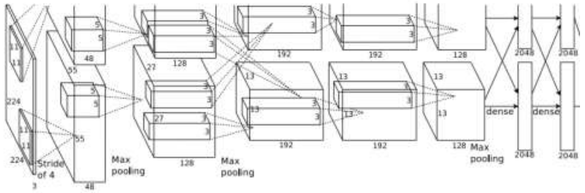
Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Correct label: Cat

Softmax Loss

Detecting a single object

This image is CC0 public domain



Vector: 4096

"What"

Fully Connected: 4096 to 1000

Class Scores

Cat: 0.9

Dog: 0.05

Car: 0.01

...

Correct label: Cat

Softmax Loss

"Where"

Fully Connected: 4096 to 4

Box Coordinates

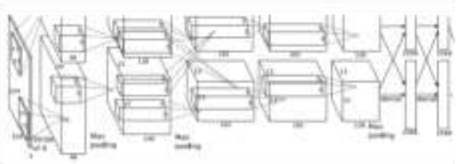
(x, y, w, h)

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

L2 Loss

Detecting a single object

This image is CC0 public domain



Vector: 4096

"What"

Fully Connected: 4096 to 1000

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

Correct label: Cat

Softmax Loss

"Where"

Fully Connected: 4096 to 4

Box Coordinates
(x, y, w, h)

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

L2 Loss

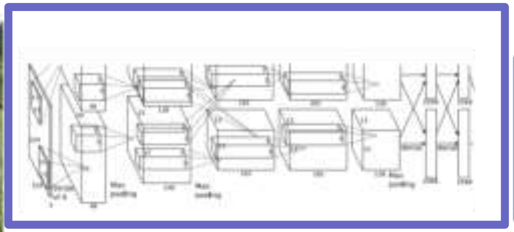
Weighted Sum → **Loss**

Multitask Loss

Detecting a single object

Often pretrained on ImageNet (Transfer learning)

This image is CC0 public domain



Vector: 4096

"What"

Fully Connected: 4096 to 1000

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

Correct label: Cat

Softmax Loss

"Where"

Fully Connected: 4096 to 4

Box Coordinates (x, y, w, h)

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

L2 Loss

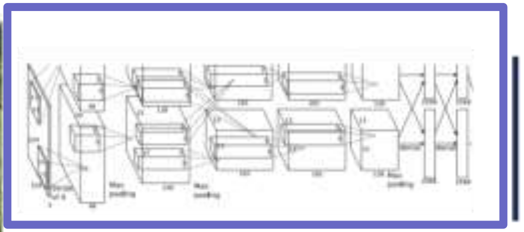
Weighted Sum → **Loss**

Multitask Loss

Detecting a single object

Often pretrained on ImageNet (Transfer learning)

This image is CC0 public domain



Vector: 4096

"What"

Fully Connected: 4096 to 1000

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

Correct label: Cat

Softmax Loss

"Where"

Fully Connected: 4096 to 4

Box Coordinates
(x, y, w, h)

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

L2 Loss

Weighted Sum → **Loss**

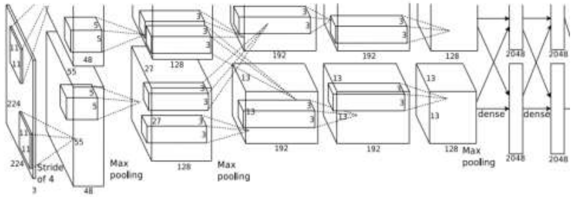
Multitask Loss

Treat localization as a regression problem!

Problem: Images can have more than one object!

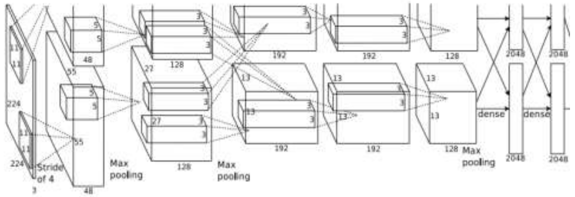
Detecting Multiple Objects

Need different numbers of outputs per image



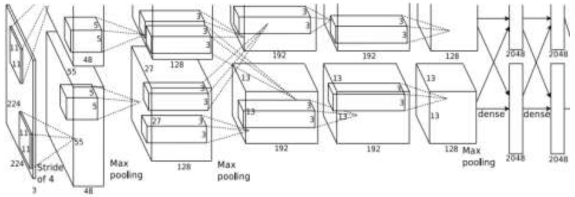
CAT: (x, y, w, h)

4 numbers



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

16 numbers

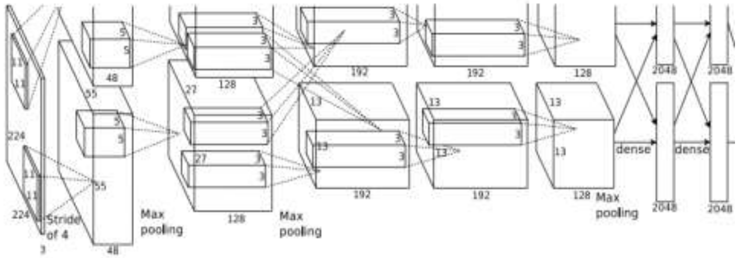


DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

Many numbers!

Detecting Multiple Objects: **Sliding Window**

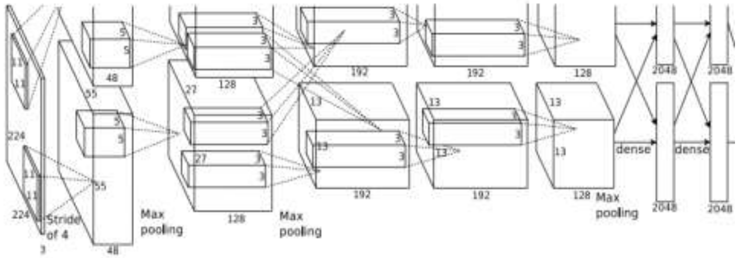
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? NO
Background? YES

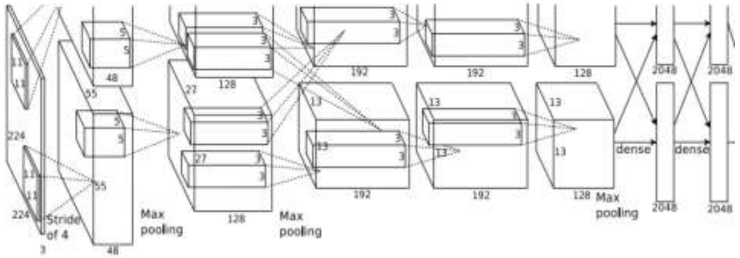
Detecting Multiple Objects: **Sliding Window**

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



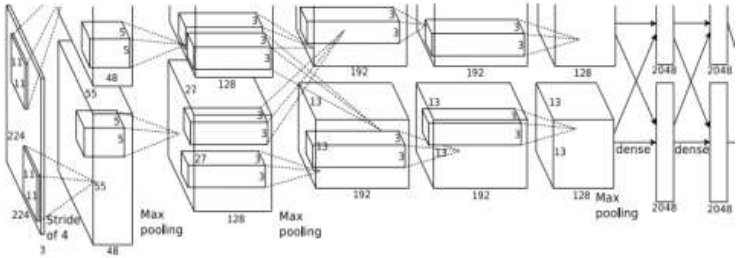
Dog? YES
Cat? NO
Background? NO

Detecting Multiple Objects: **Sliding Window**



Dog? YES
Cat? NO
Background? NO

Detecting Multiple Objects: **Sliding Window**



Dog? NO
Cat? YES
Background? NO

Detecting Multiple Objects: **Sliding Window**

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

Detecting Multiple Objects: **Sliding Window**

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

Consider a box of size h x w:
Possible x positions: W – w + 1
Possible y positions: H – h + 1
Possible positions: (W – w + 1) * (H – h + 1)

Detecting Multiple Objects: **Sliding Window**

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

Consider a box of size h x 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**

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

Total possible boxes:

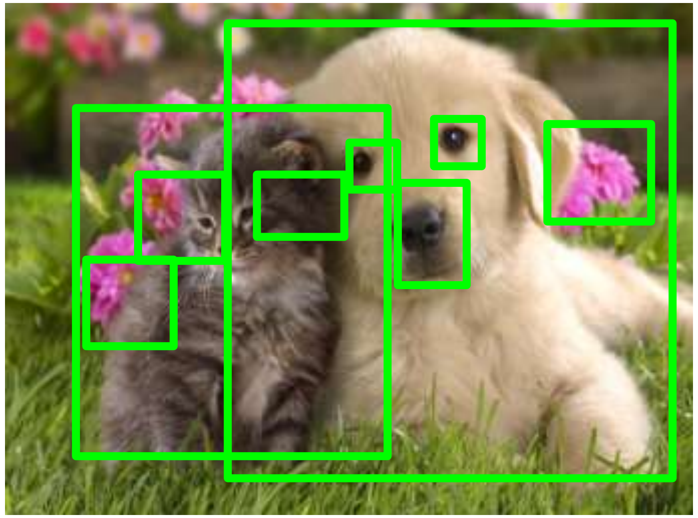
$$\sum_{h=1}^{H}\sum_{w=1}^{W}(W-w+1)(H-h+1)$$

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

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

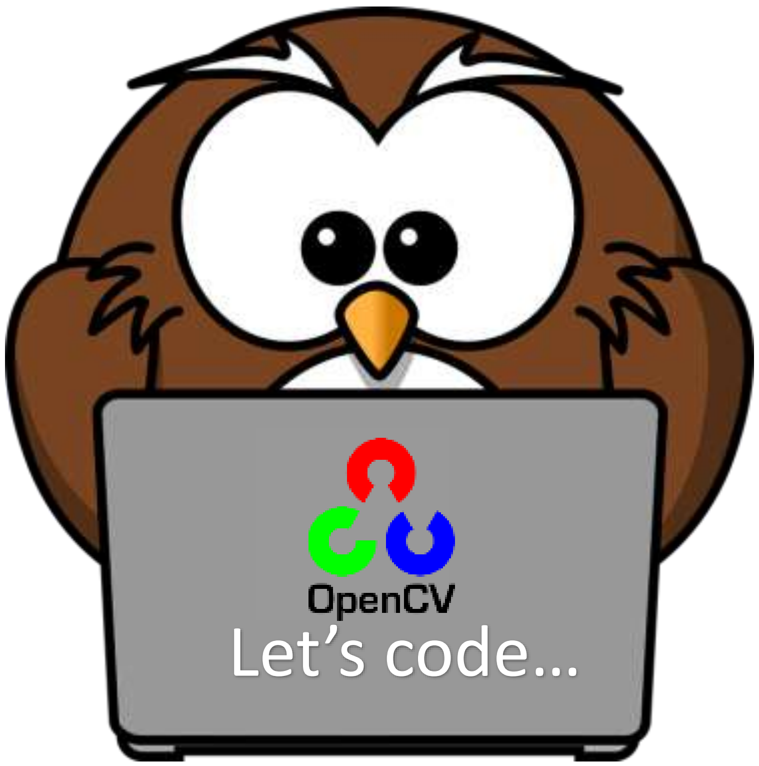
Region Proposals

- 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



Alexe et al, “Measuring the objectness of image windows”, TPAMI 2012
Uijlings et al, “Selective Search for Object Recognition”, IJCV 2013
Cheng et al, “BING: Binarized normed gradients for objectness estimation at 300fps”, CVPR 2014
Zitnick and Dollar, “Edge boxes: Locating object proposals from edges”, ECCV 2014

Region Proposals



5-selective-search.ipynb

R-CNN: Region-Based CNN

Input image

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

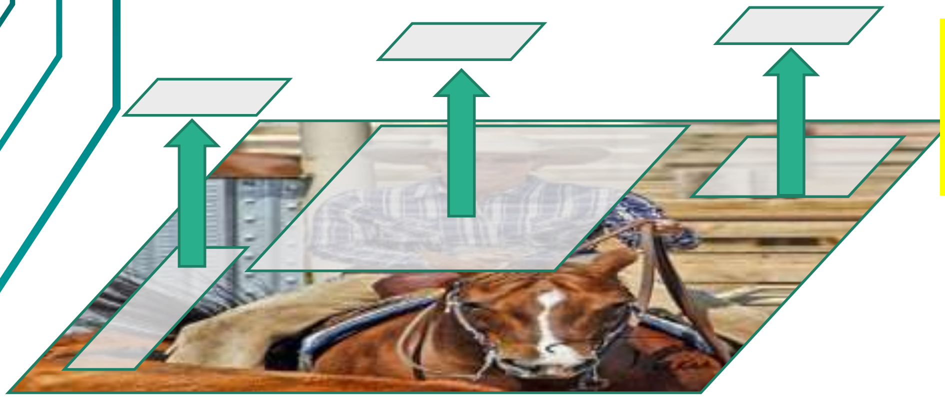
R-CNN: Region-Based CNN

Regions of Interest (RoI) from a proposal method (~2k)

Input image

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: Region-Based CNN

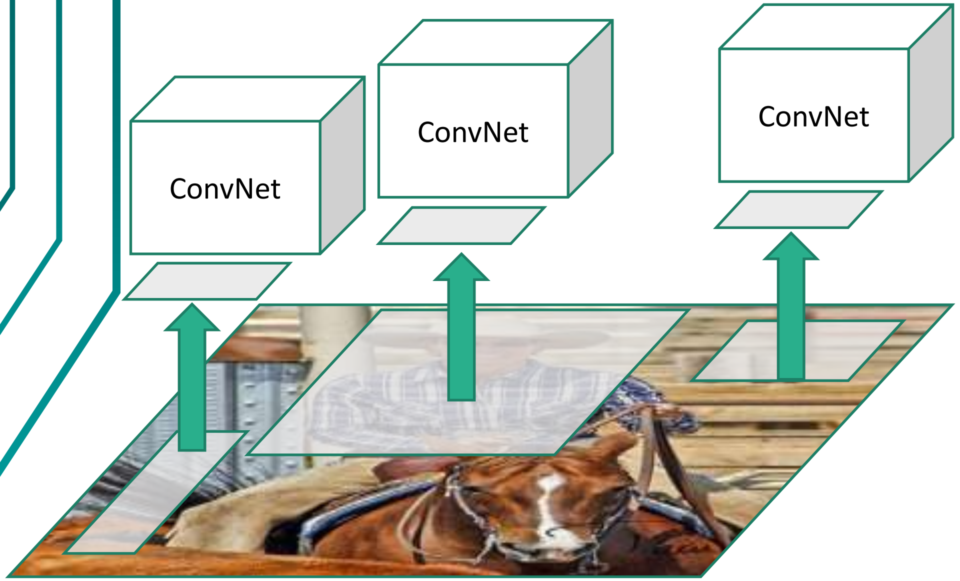


Warped image regions (224x224)

Input image

Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.

R-CNN: Region-Based CNN



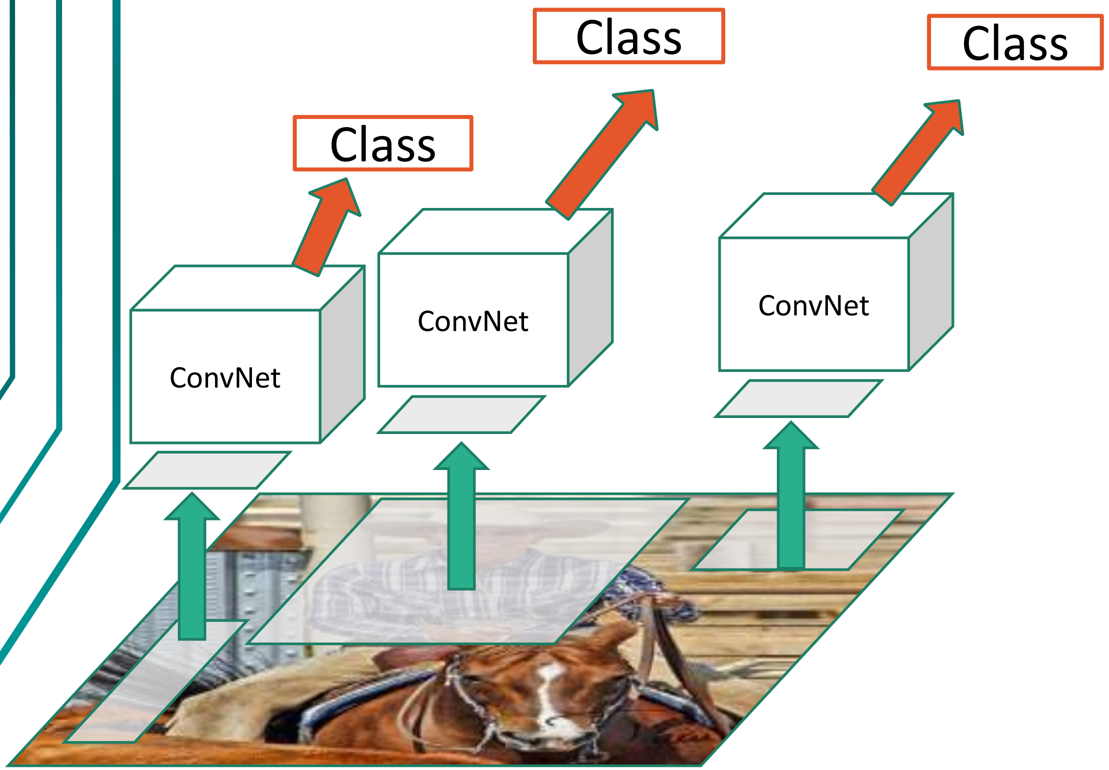
Input image

Forward each region through Convolutional network

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: Region-Based CNN

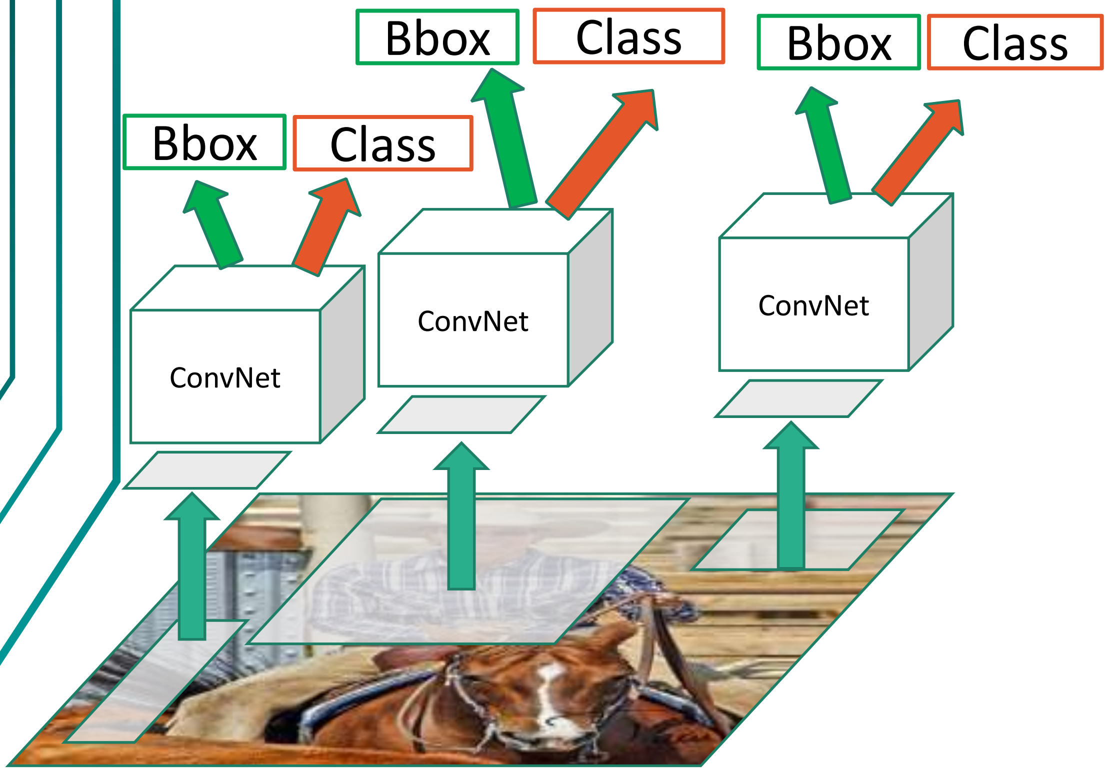
Classify each region



Input image

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: Region-Based CNN



Input image

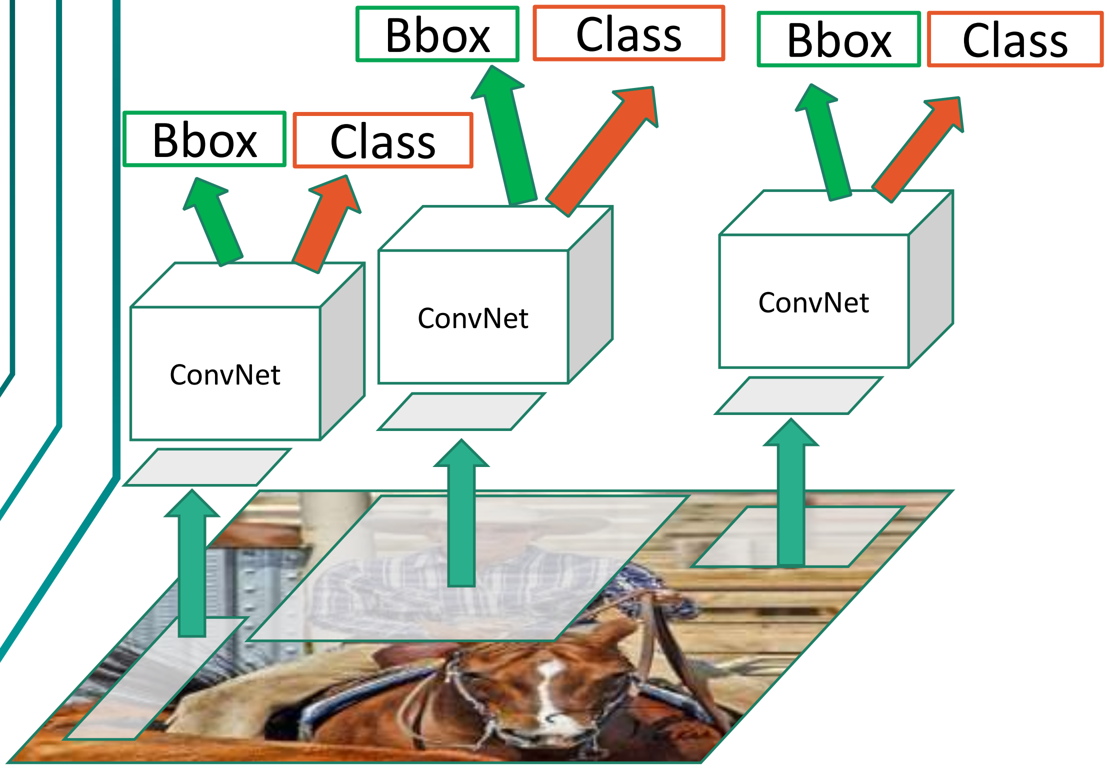
Classify each region

Bounding box regression:

Predict “transform” to correct the RoI: 4 numbers (t_x, t_y, t_h, t_w)

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: Test-time



Input image

Input: Single RGB Image

1. Run region proposal method to compute ~2000 region proposals
2. Resize each region to 224x224 and run independently through CNN to predict class scores and bbox transform
3. Use scores to select a subset of region proposals to output
   (Many choices here: threshold on background, or per-category? Or take top K proposals per image?)
4. Compare with ground-truth boxes

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

Comparing Boxes: Intersection over Union (IoU)

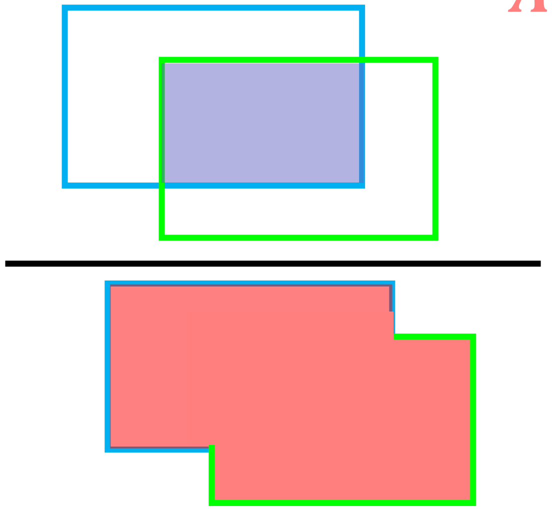
How can we compare our prediction to the ground-truth box?

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"):

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

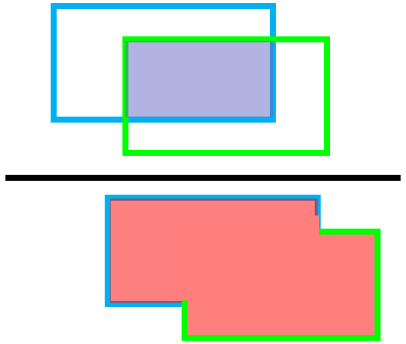


Comparing Boxes: Intersection over Union (IoU)

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

Intersection over Union (IoU)
(Also called “Jaccard similarity” or “Jaccard index”):

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



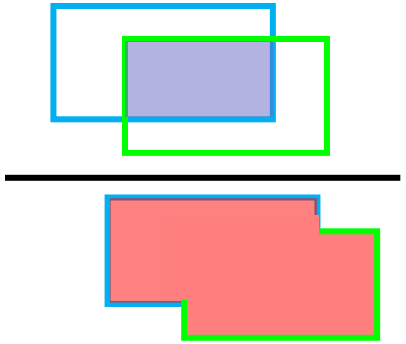
IOU > 0.5 is “decent”

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"):

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



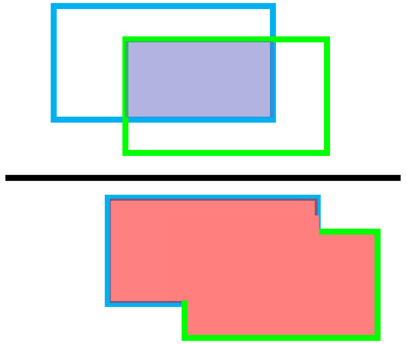
IOU > 0.5 is "decent"
IOU > 0.7 is "pretty good"

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”):

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



IOU > 0.5 is “decent”
IOU > 0.7 is “pretty good”
IOU > 0.9 is “almost perfect”

What is bBox?

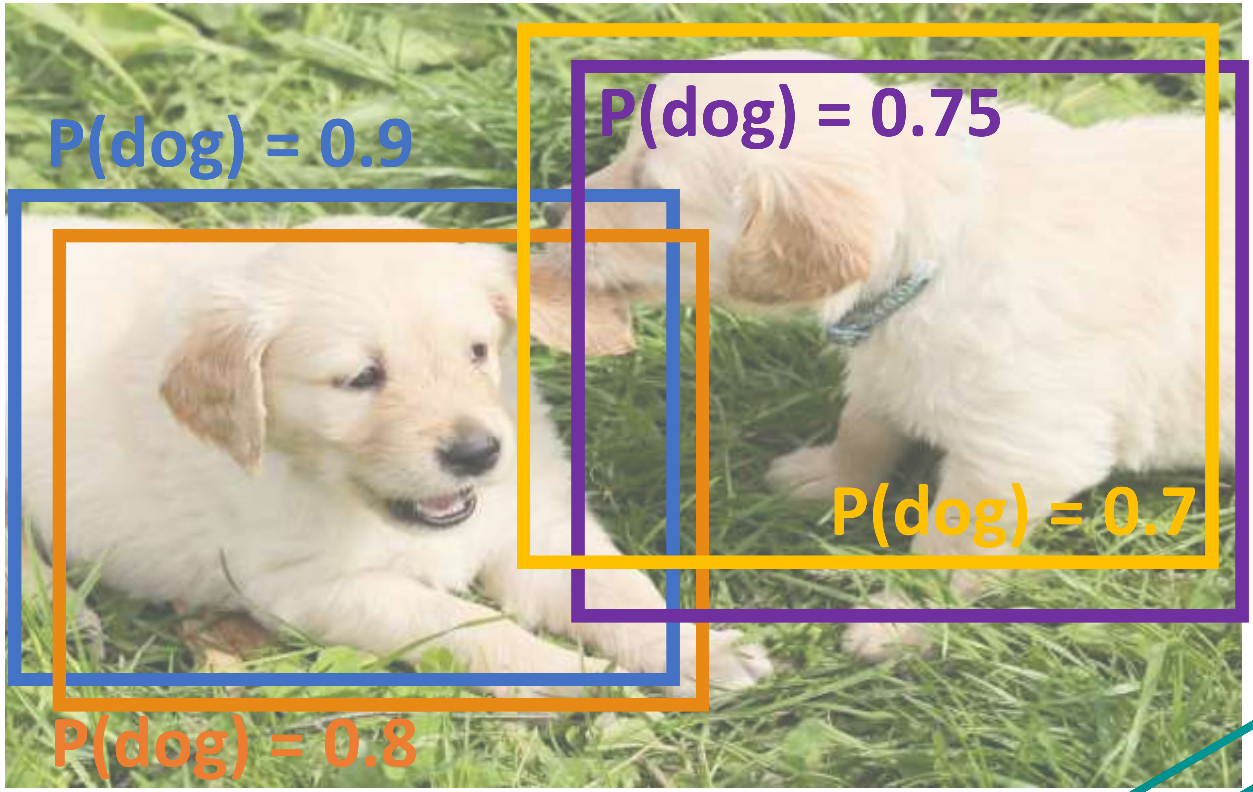
6-object-detection-and-bounding-boxes.ipynb

Intersection over Union (IoU)

7-Intersection-over-Union(IoU).ipynb

Overlapping Boxes

Problem: Object detectors often output many overlapping detections:



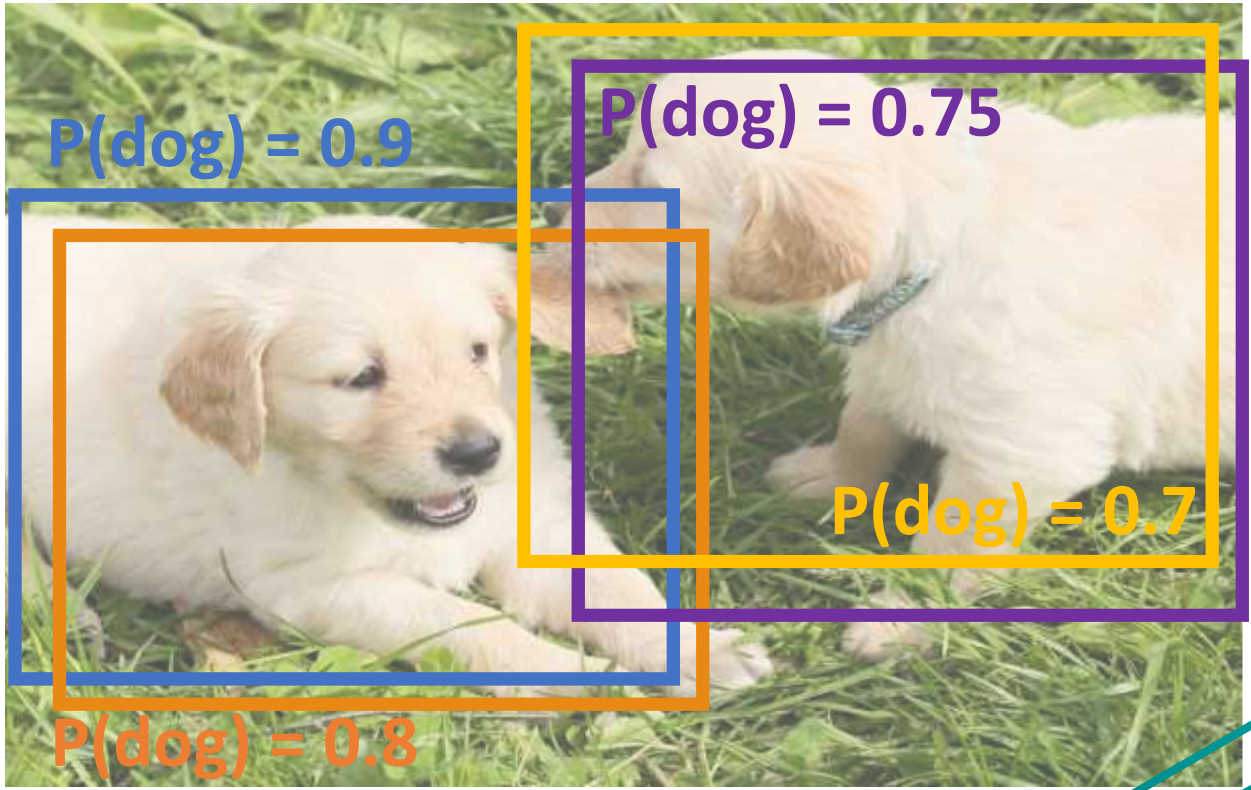
Puppy image is CC0 Public Domain

Overlapping Boxes: **Non-Max Suppression (NMS)**

Problem: Object detectors often output many overlapping detections:

Solution: Post-process raw detections using **Non-Max Suppression (NMS)**

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If any boxes remain, GOTO 1



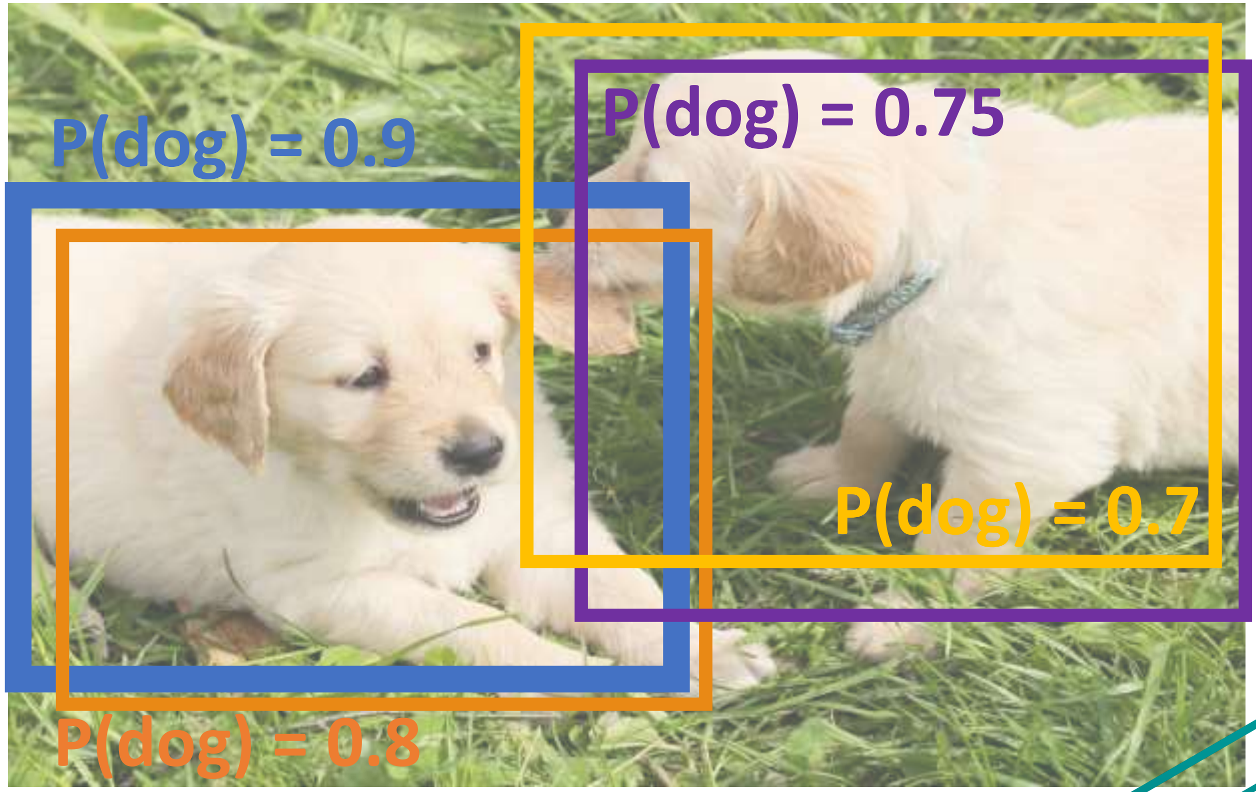
Overlapping Boxes: **Non-Max Suppression (NMS)**

Problem: Object detectors often output many overlapping detections:

Solution: Post-process raw detections using **Non-Max Suppression (NMS)**

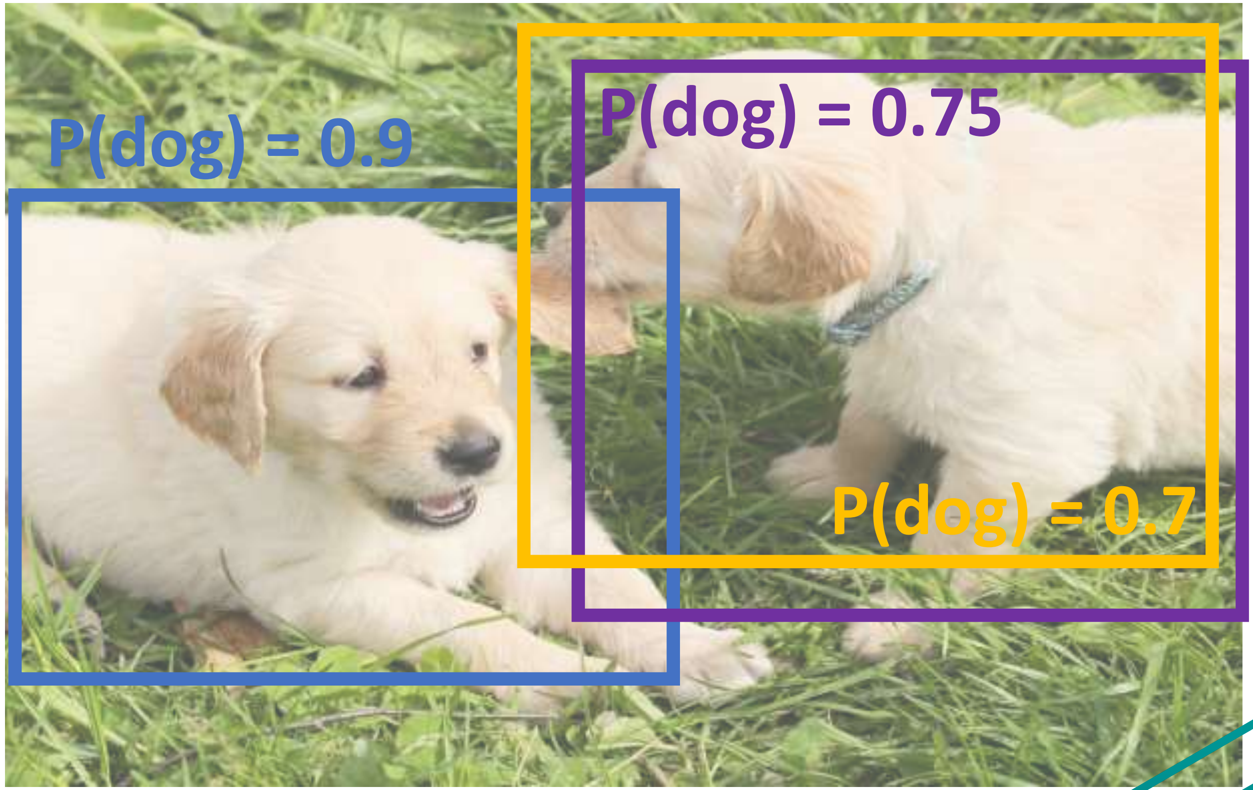
1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If any boxes remain, GOTO 1

IoU(■, ■) = **0.78**
IoU(■, ■) = 0.05
IoU(■, ■) = 0.07



Overlapping Boxes: **Non-Max Suppression (NMS)**

Problem: Object detectors often output many overlapping detections:



Solution: Post-process raw detections using **Non-Max Suppression (NMS)**

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If any boxes remain, GOTO 1

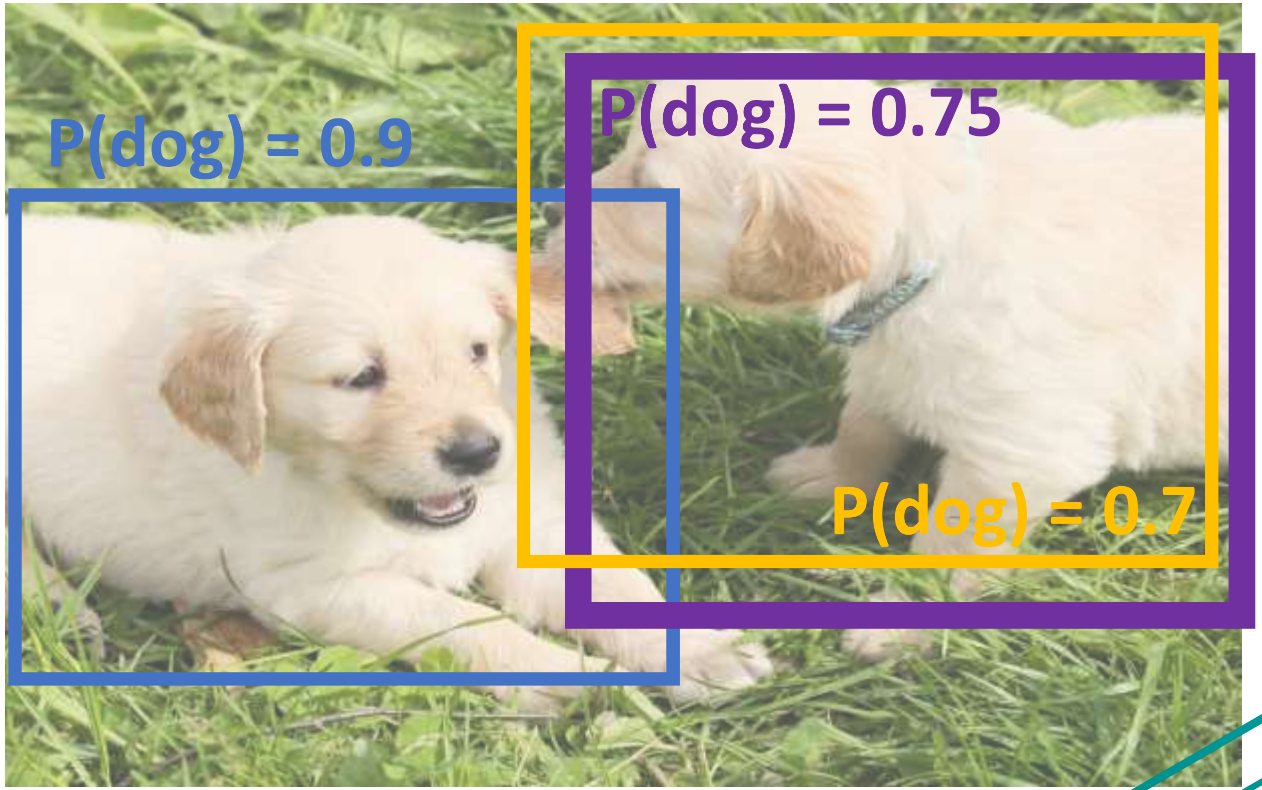
Overlapping Boxes: **Non-Max Suppression (NMS)**

Problem: Object detectors often output many overlapping detections:

Solution: Post-process raw detections using **Non-Max Suppression (NMS)**

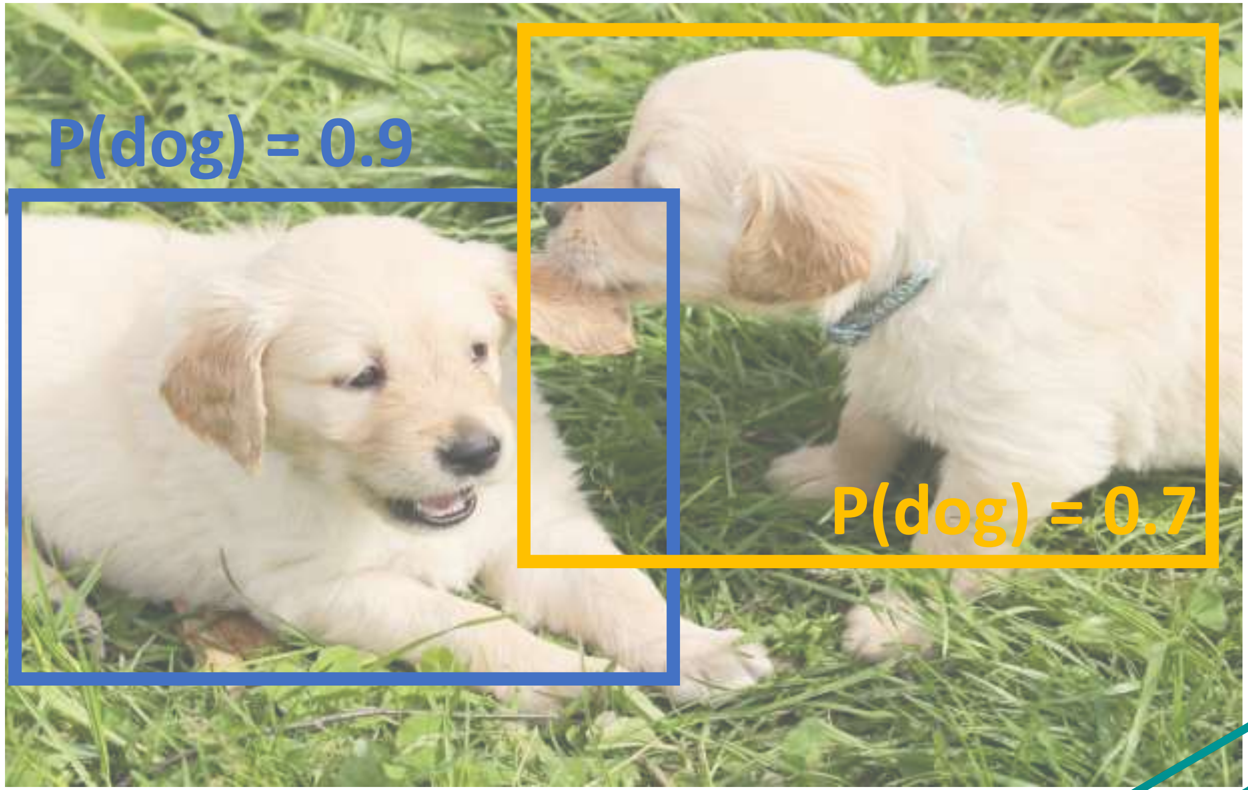
1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If any boxes remain, GOTO 1

IoU(■, ■) = **0.74**



Overlapping Boxes: **Non-Max Suppression (NMS)**

Problem: Object detectors often output many overlapping detections:



Solution: Post-process raw detections using **Non-Max Suppression (NMS)**

1. Select next highest-scoring box
2. Eliminate lower-scoring boxes with IoU > threshold (e.g. 0.7)
3. If any boxes remain, GOTO 1

Overlapping Boxes: **Non-Max Suppression (NMS)**

Problem:

NMS may eliminate ”good” boxes when objects are highly overlapping!

Evaluating Object Detectors: Mean Average Precision (mAP)