Computer vision and machine learning for the material scientist

Lecture 9.
Object Detection

Romain Vo

*slides adapted from CS231n

Computer Vision Tasks

Classification

DOG

$\underbrace{\hspace{4cm}}$

<u>Classify</u> the image

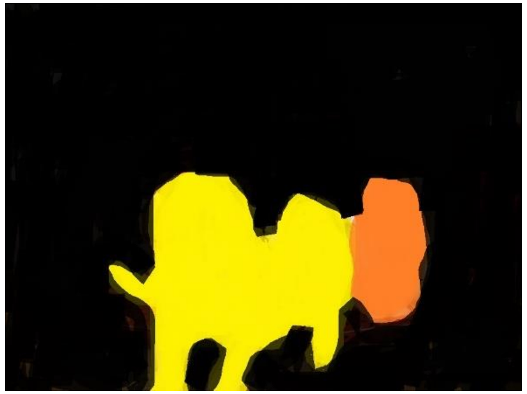
Computer Vision Tasks

Classification

DOG

Classify the image

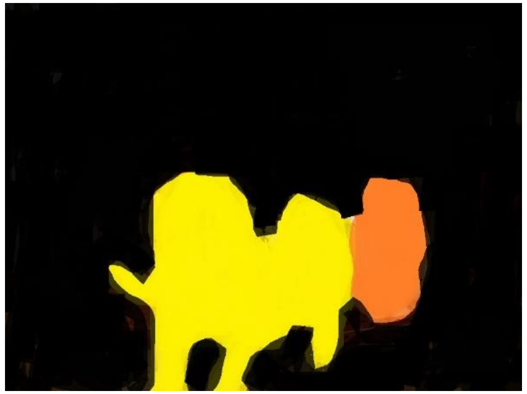
Semantic segmentation



DOG,CAT,BG

Classify each pixel

Computer Vision Tasks

Classification	Semantic segmentation	Instance Segmentation	Panoptic segmentation
			
DOG	**DOG,CAT,BG**	**SMTH, SMTH, SMTH**	**DOG,DOG,CAT**
Classify the image	Classify each pixel	Segment independent instances	Segment & Classify independent instances

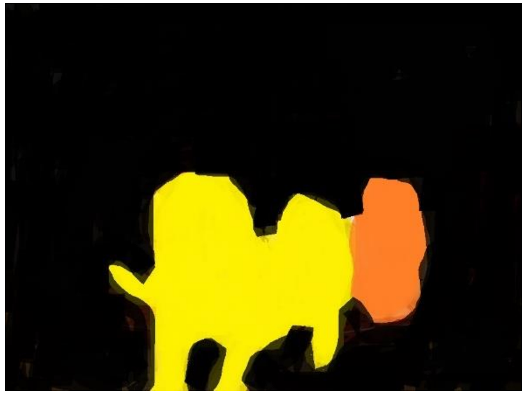
Computer Vision Tasks

Classification

DOG

<u>Classify</u> the image

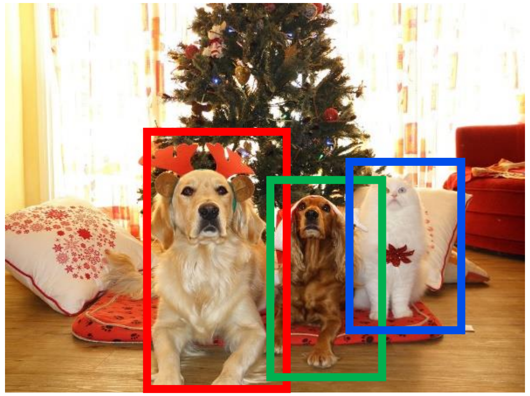
Semantic segmentation



DOG,**CAT**,**BG**

<u>Classify</u> each pixel

Instance <u>Detection</u>



SMTH, **SMTH**, **SMTH**

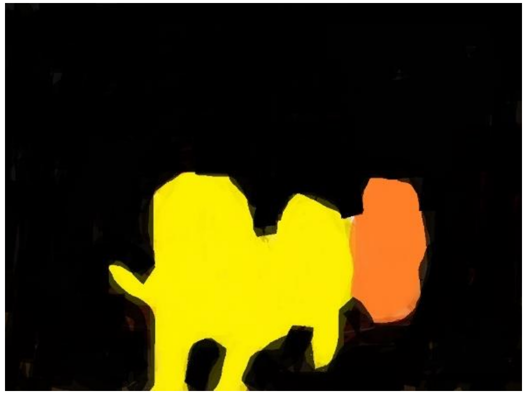
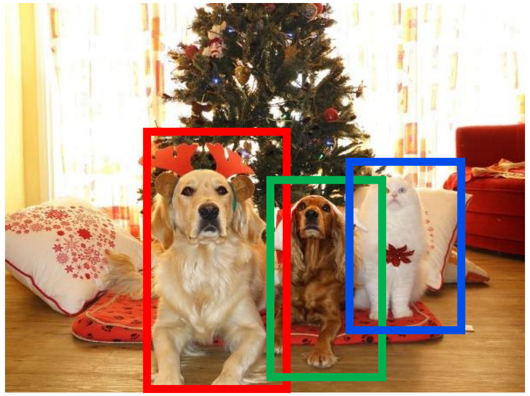
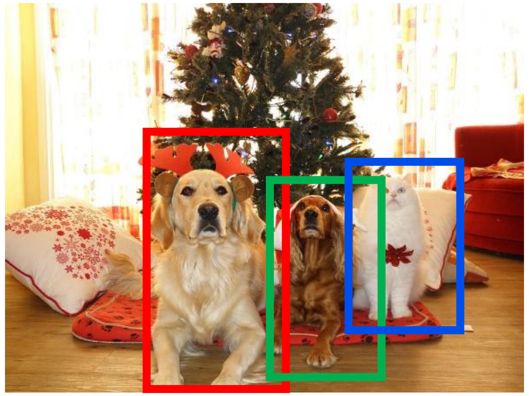
<u>Detect</u> independent instances

Panoptic segmentation

DOG,**DOG**,**CAT**

<u>Segment</u> & <u>Classify</u> independent instances

Computer Vision Tasks

Classification	Semantic segmentation	Instance <u>Detection</u>	Panoptic <u>Detection</u>
			
DOG	**DOG,CAT,BG**	**SMTH, SMTH, SMTH**	**DOG,DOG,CAT**
<u>Classify</u> the image	<u>Classify</u> each pixel	<u>Detect</u> independent instances	<u>Detect</u> & <u>Classify</u> independent instances

MINES PARIS | PSL

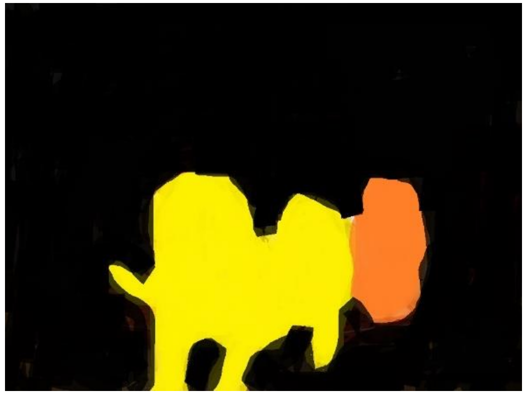
Computer Vision Tasks

Classification

DOG

Classify the image

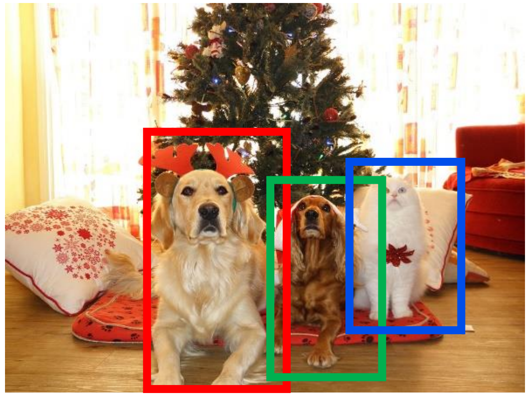
Semantic segmentation



DOG,CAT,BG

Classify each pixel

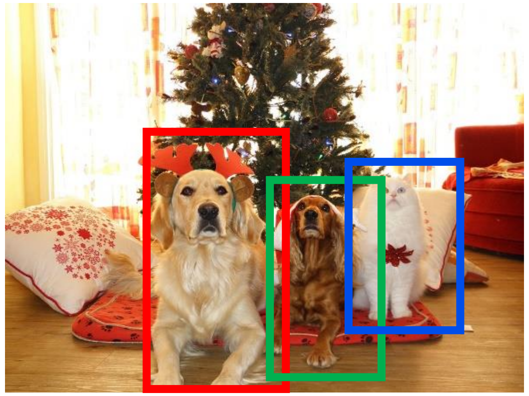
Instance Detection



SMTH, SMTH, SMTH

Detect independent instances

Object Detection



DOG,DOG,CAT

Detect & Classify independent instances

Object Detection

Classification

DOG

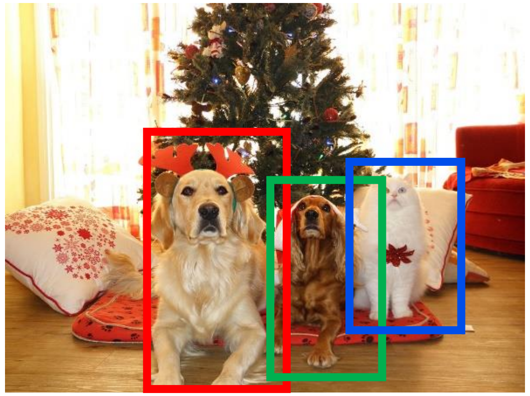
Classify the image

Semantic segmentation

DOG,CAT,BG

Classify each pixel

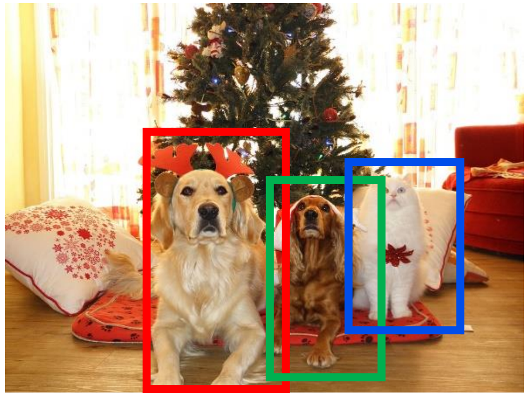
Instance Detection



SMTH, SMTH, SMTH

Detect independent instances

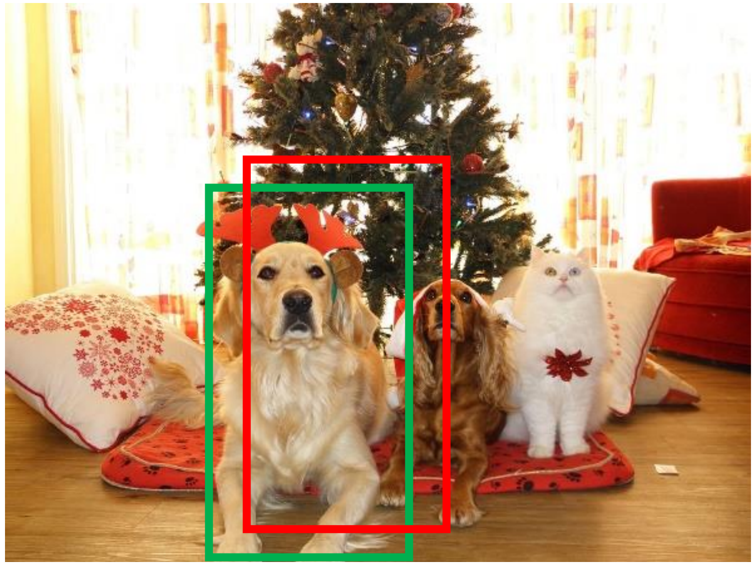
Object Detection



DOG,DOG,CAT

Detect & Classify independent instances

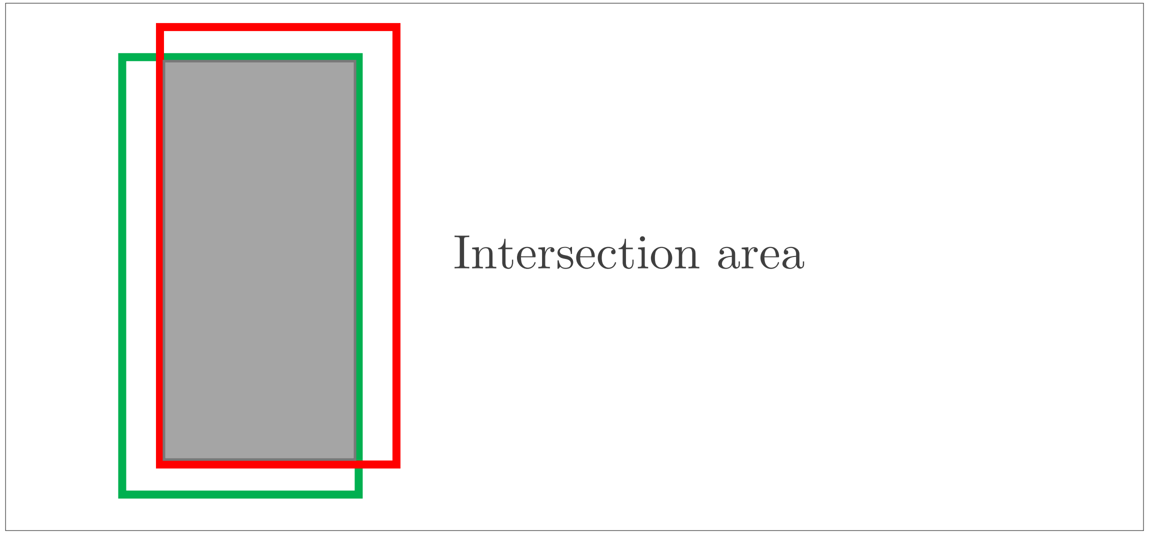
Object Detection : metrics

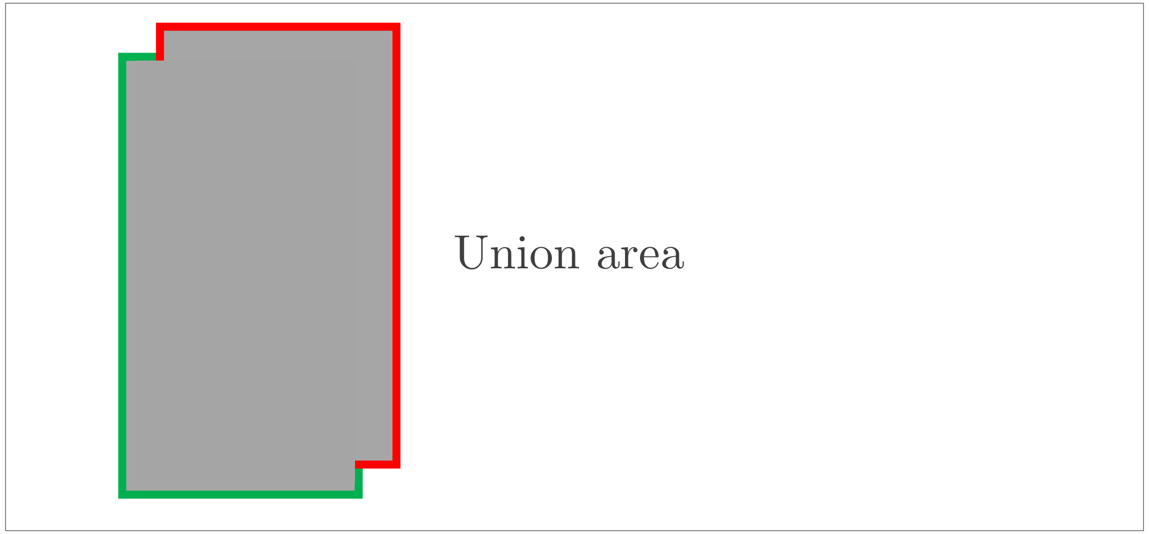


Ground-truth bounding box

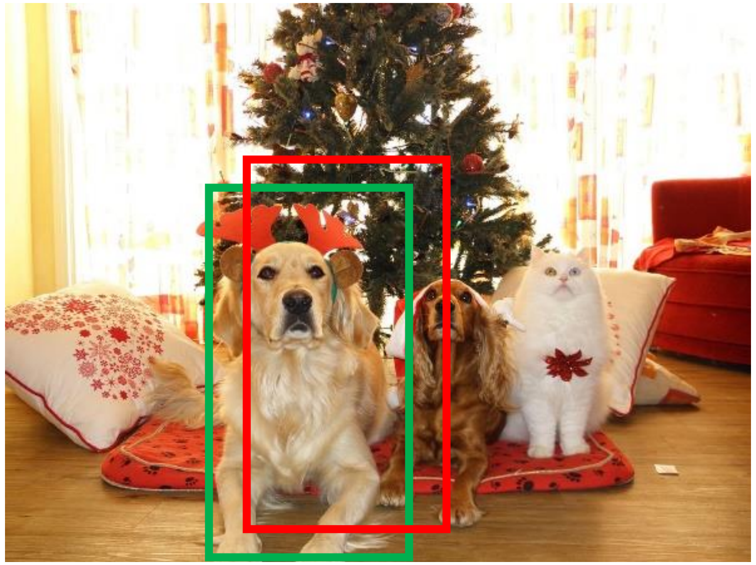
Predicted box

$$IoU = \frac{Intersection}{Union}$$





Object Detection : metrics



Ground-truth bounding box

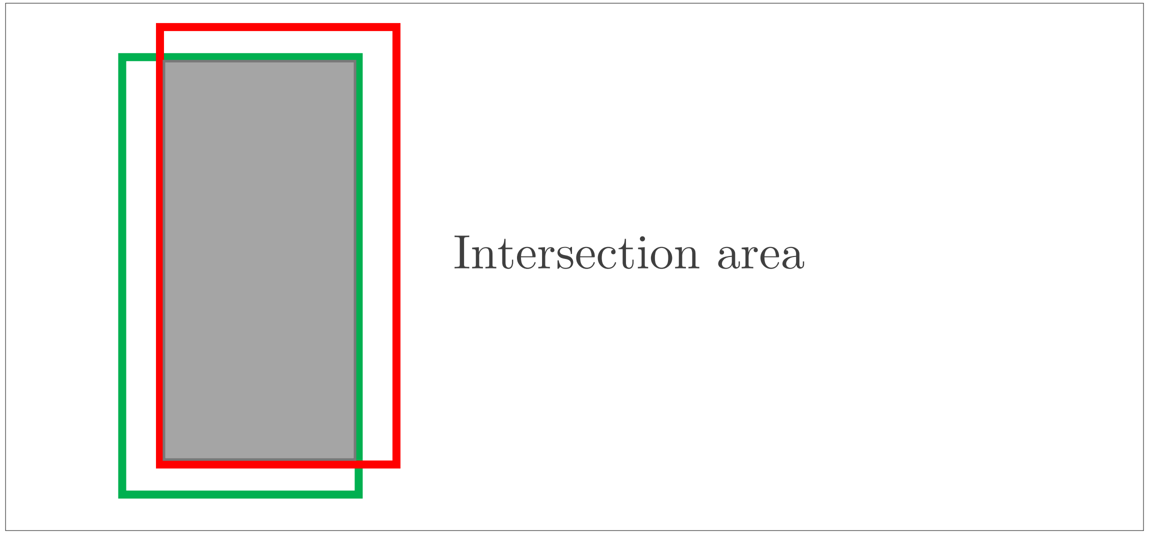
Predicted box

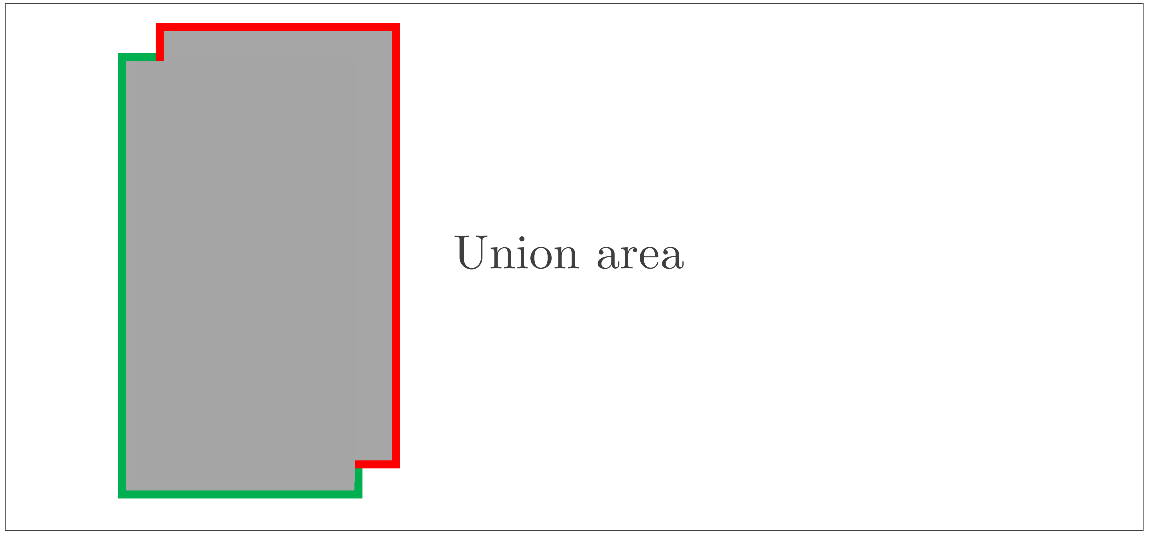
$$IoU = \frac{Intersection}{Union}$$

A box is correctly detected when $IoU > t$

t = *user-defined threshold*

(and when the label associated is correct)



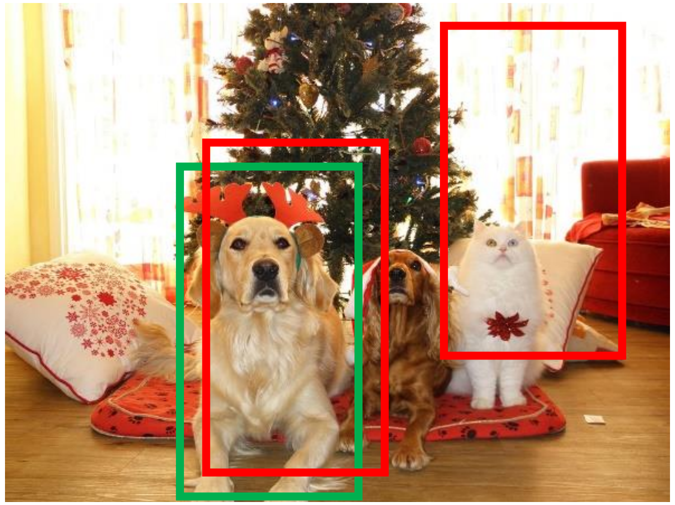


MINES PARIS | PSL

Object Detection : Precision / Recall

Ground-truth Prediction

True positive



Ground-truth Prediction

False positive

Ground-truth ~~Prediction~~

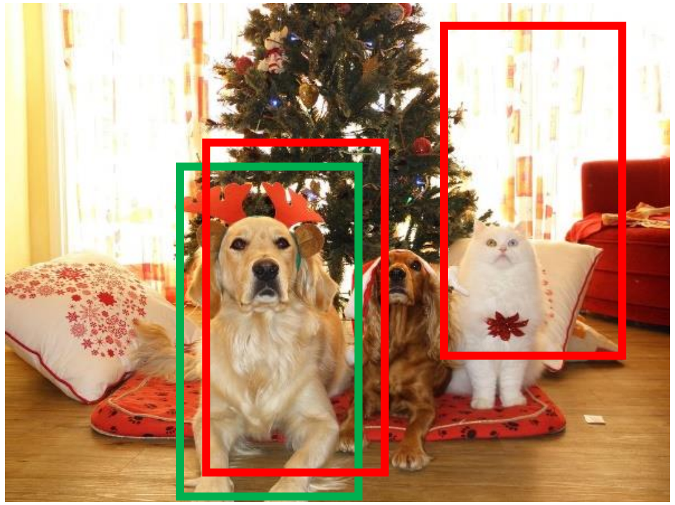
False negative

Precision: $P = \frac{TP}{TP+FP}$

Object Detection : Precision / Recall

Ground-truth Prediction

True positive



Ground-truth Prediction

False positive

Ground-truth ~~Prediction~~

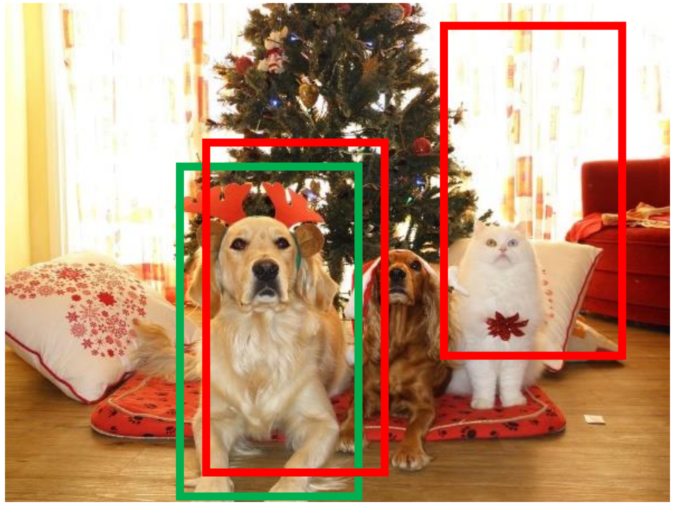
False negative

Precision: $P = \frac{TP}{TP+FP}$ $\quad$ $i.e$ rate of well-detected objects among detected objects

Object Detection : Precision / Recall

Ground-truth Prediction

True positive



Ground-truth Prediction

False positive

Ground-truth ~~Prediction~~

False negative

Precision: $P = \frac{TP}{TP+FP}$ $i.e$ rate of well-detected objects among detected objects

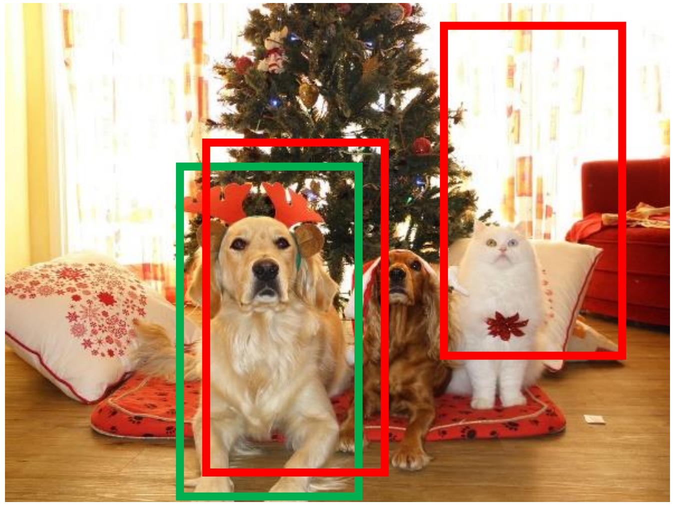
Recall: $R = \frac{TP}{TP+FN}$

MINES PARIS | PSL

Object Detection : Precision / Recall

Ground-truth Prediction

True positive



Ground-truth Prediction

False positive

Ground-truth ~~Prediction~~

False negative

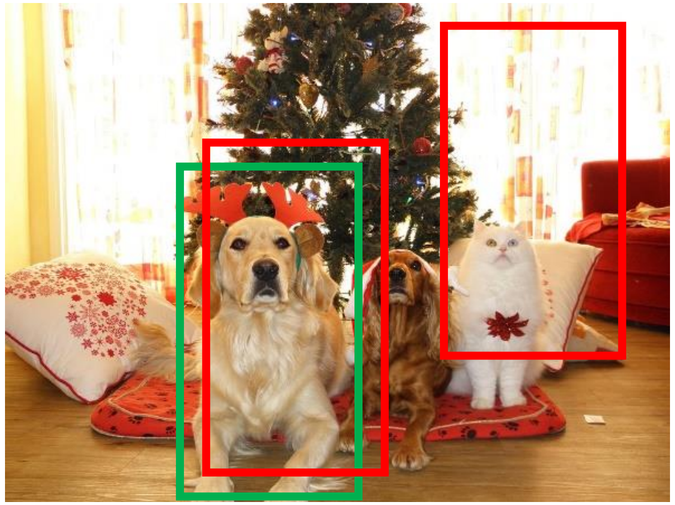
Precision: $P = \frac{TP}{TP+FP}$ *i.e* rate of well-detected objects among detected objects

Recall: $R = \frac{TP}{TP+FN}$ *i.e* rate of well-detected objects among ground-truth objects

Object Detection : Precision / Recall

Ground-truth Prediction

True positive



Ground-truth Prediction

False positive

Ground-truth ~~Prediction~~

False negative

Precision: $P = \frac{TP}{TP+FP}$ $i.e$ rate of well-detected objects among detected objects

Recall: $R = \frac{TP}{TP+FN}$ $i.e$ rate of well-detected objects among ground-truth objects

Intuition : if t is very low or close to 0, all predictions are considered True Positive, but we increase the number of False Positive (and inversely).

Precision/Recall trade-off

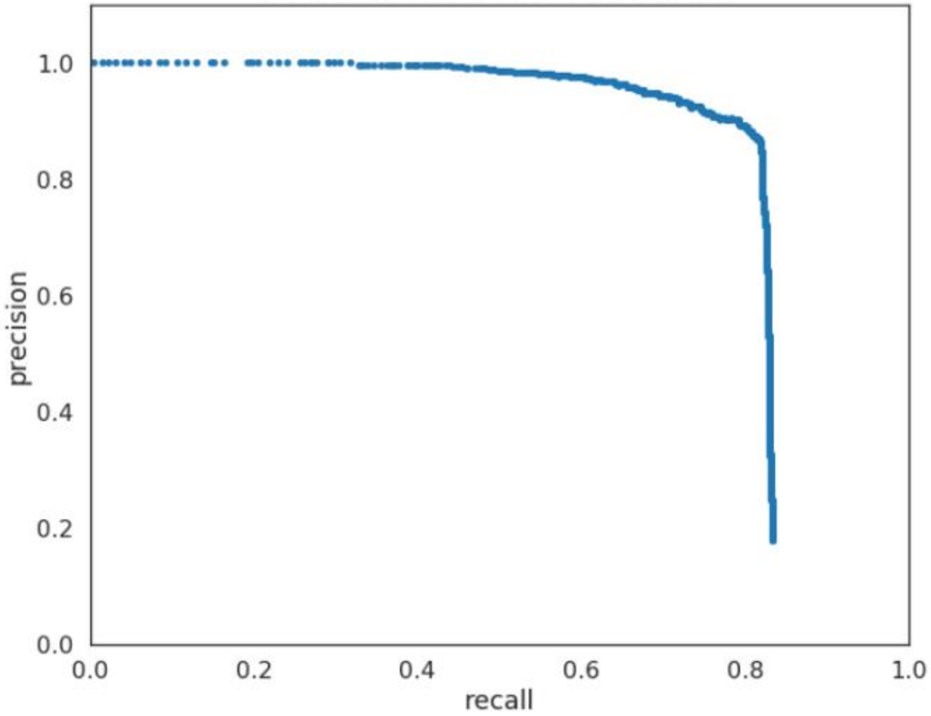
MINES PARIS | PSL

Object Detection : Precision / Recall curve and mAP

Precision: $P = \frac{TP}{TP+FP}$ **Recall:** $R = \frac{TP}{TP+FN}$

True Positive if $IoU > t$

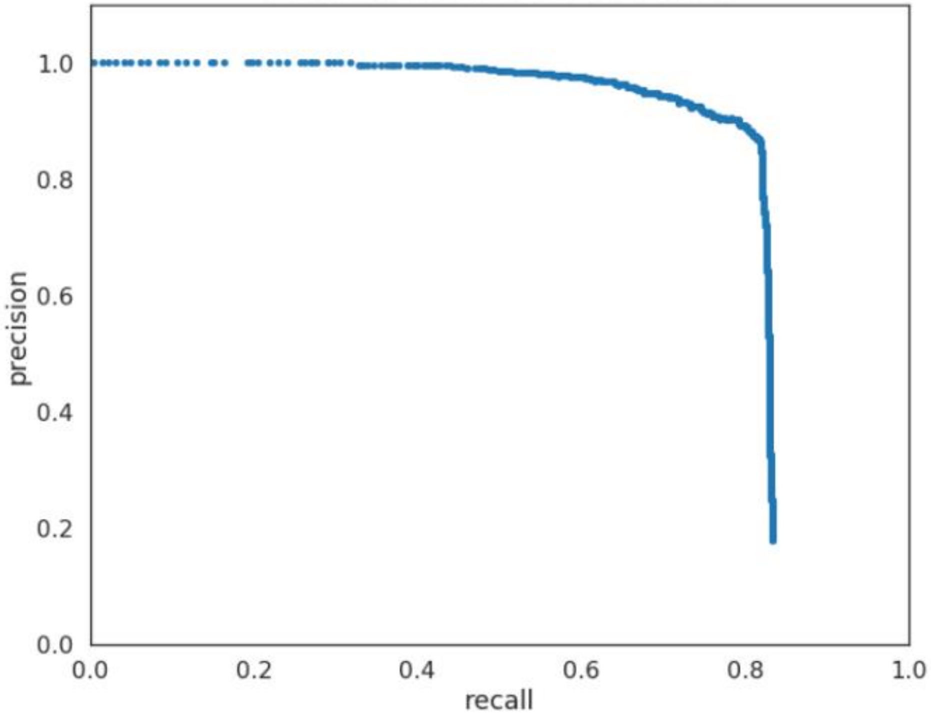
- By varying the confidence threshold t for the detection, one can obtain so-called precision-recall curve



Object Detection : Precision / Recall curve and mAP

Precision: $P = \frac{TP}{TP+FP}$ **Recall:** $R = \frac{TP}{TP+FN}$ True Positive if $IoU > t$

- By varying the confidence threshold t for the detection, one can obtain so-called precision-recall curve
- The average precision (AP) is defined as the area under the curve



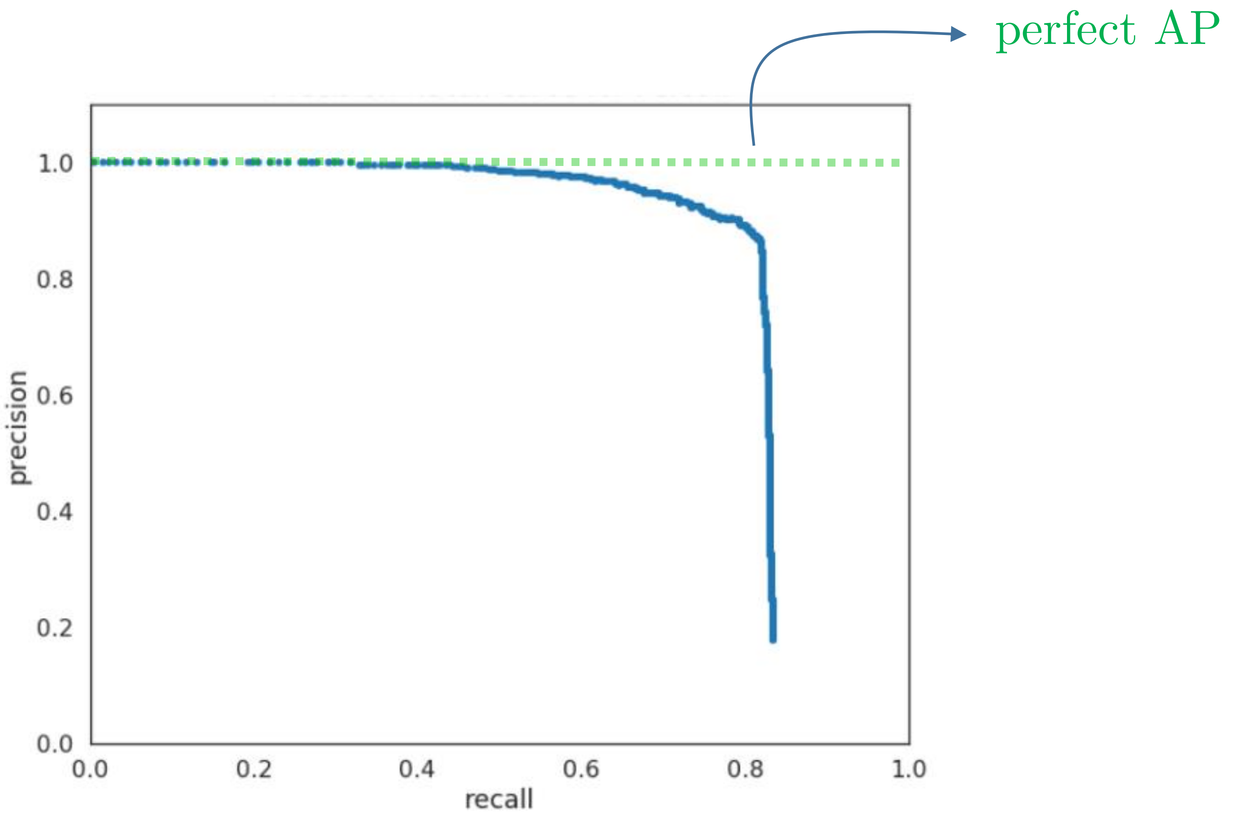
MINES PARIS | PSL

Object Detection : Precision / Recall curve and mAP

Precision: $P = \frac{TP}{TP+FP}$ **Recall:** $R = \frac{TP}{TP+FN}$

True Positive if $IoU > t$

- By varying the confidence threshold t for the detection, one can obtain so-called precision-recall curve
- The average precision (AP) is defined as the area under the curve



MINES PARIS | PSL

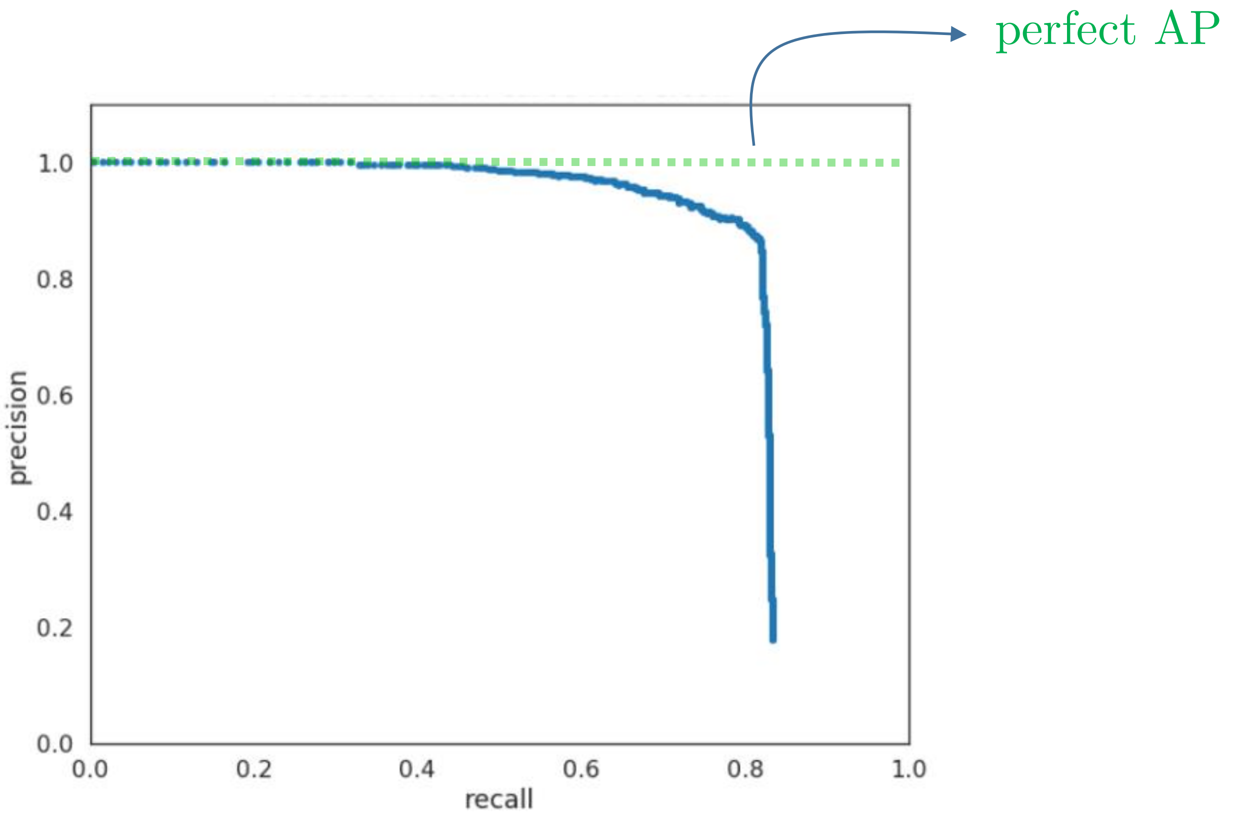
Object Detection : Precision / Recall curve and mAP

Precision: $P = \frac{TP}{TP+FP}$ **Recall:** $R = \frac{TP}{TP+FN}$

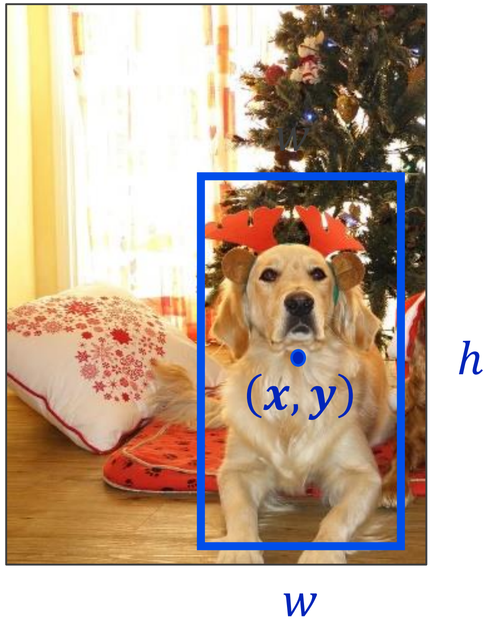
True Positive if $IoU > t$

- By varying the confidence threshold t for the detection, one can obtain so-called precision-recall curve
- The average precision (AP) is defined as the area under the curve
- We say mean average precision (mAP) because the AP is computed for all object classes

$$\text{mAP} = mean(AP_{dog}, AP_{cat}, AP_{human})$$



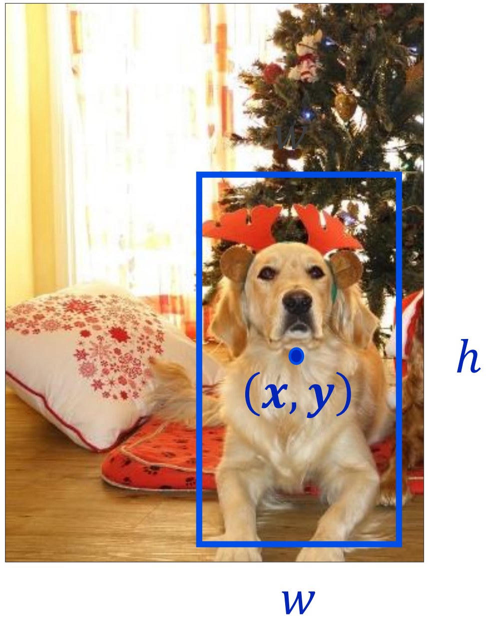
Object Detection : Box Regression



a box is defined by 4 parameters:

- center (x, y)
- height h
- width w

Object Detection : Single object

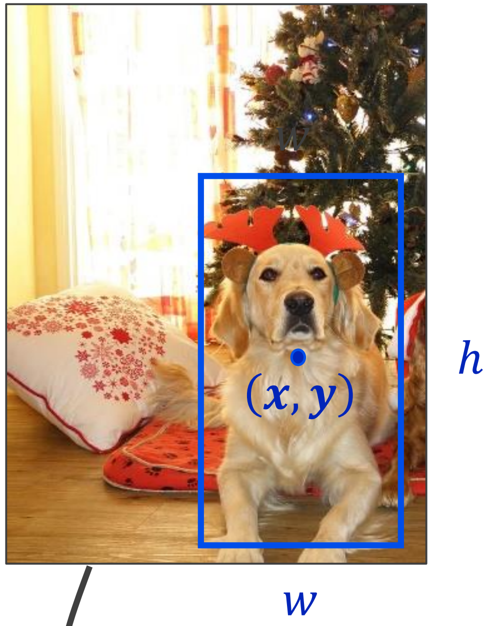


a box is defined by 4 parameters:

- center (x, y)
- height h
- width w

and, we assign a class label c to this box

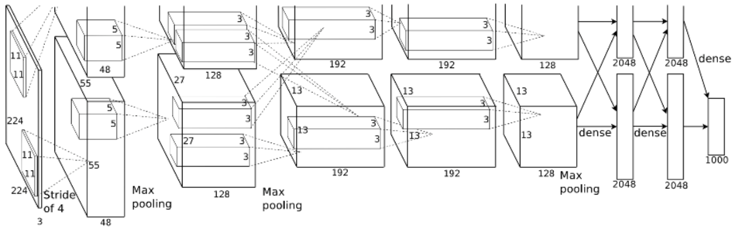
Object Detection : Single object



a box is defined by 4 parameters:

- center (x, y)
- height h
- width w

and, we assign a class label $\underline{c}$ to this box

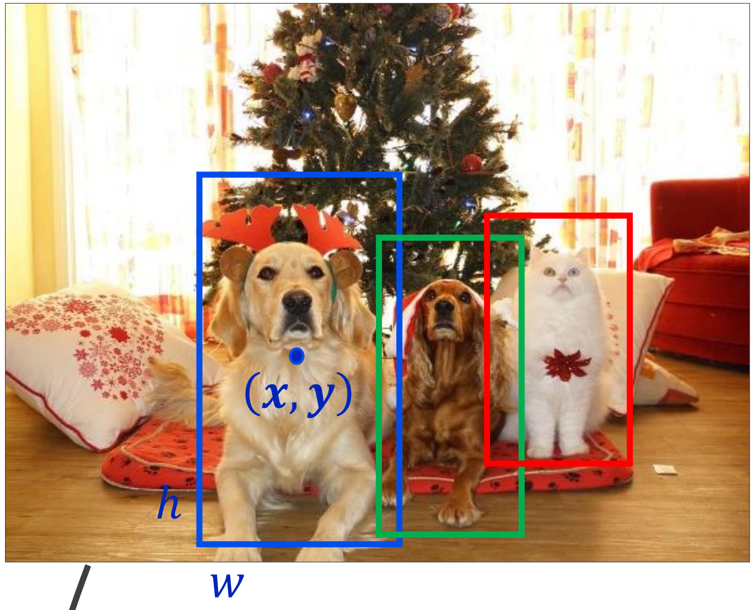


class vector $z \in \mathbb{R}^C$ → cross-entropy loss

box prediction (x', y', h', w') → ℓ_2 regression loss

$\boxed{+}$ → Total loss

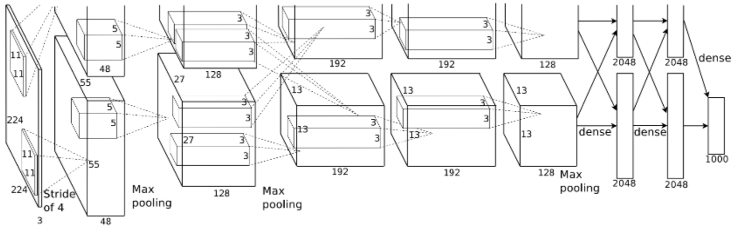
Object Detection : Single object



a box is defined by 4 parameters:

- center (x, y)
- height h
- width w

and, we assign a class label c to this box

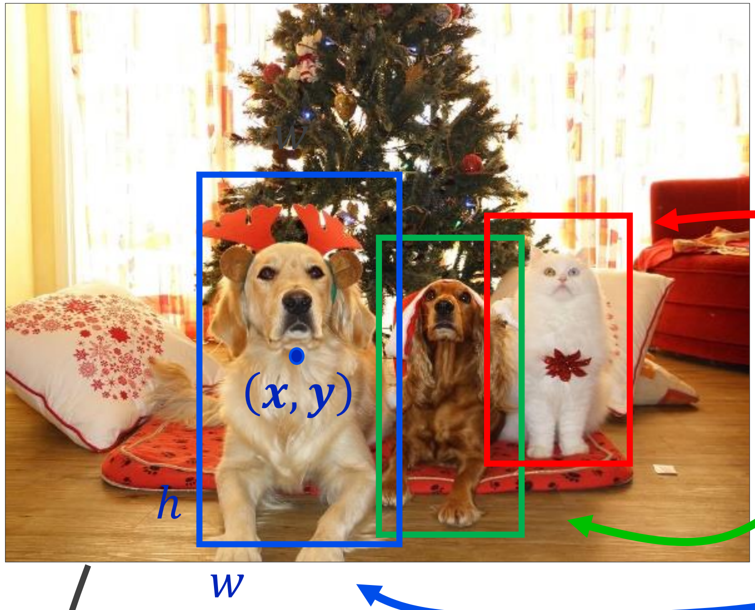


class vector $z \in \mathbb{R}^C$ → cross-entropy loss

box prediction (x', y', h', w') → ℓ_2 regression loss

cross-entropy loss + ℓ_2 regression loss → Total loss

Object Detection : Multi objects

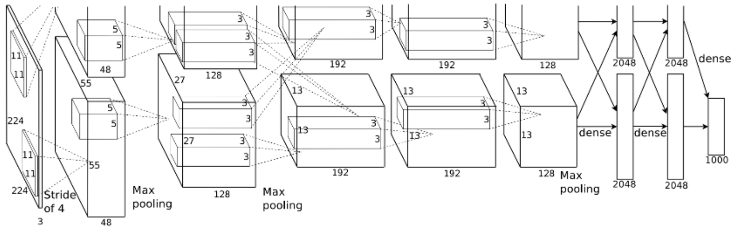


a box is defined by 4 parameters:

- center (x, y)
- height h
- width w

and, we assign a class label c to this box

Problem: how to treat a varying number of Objects ??



class vector $z \in \mathbb{R}^C$ → cross-entropy loss

box prediction (x', y', h', w') → ℓ_2 regression loss

$\boxed{+}$ → Total loss

Object Detection : Multi objects

a box is defined by 4 parameters:

- center (x, y)
- height h
- width w

and, we assign a class label c to this box

Naive approach : sample all possible crops on the image and evaluate the classification/regression network

MINES PARIS | PSL

Object Detection : Multi objects

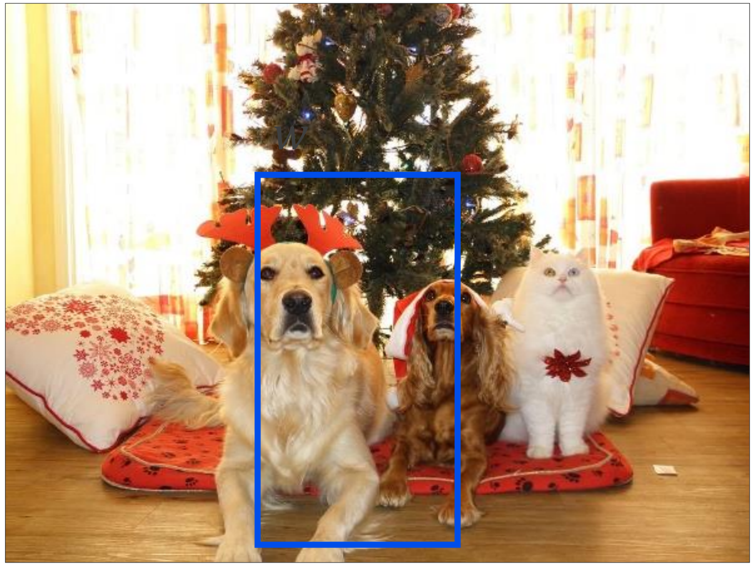
a box is defined by 4 parameters:

- center (x, y)
- height h
- width w

and, we assign a class label c to this box

Naive approach : sample all possible crops on the image and evaluate the classification/regression network

Object Detection : Multi objects



a box is defined by 4 parameters:

- center (x, y)
- height h
- width w

and, we assign a class label c to this box

Naive approach : sample all possible crops on the image and evaluate the classification/regression network

Object Detection : Multi objects

a box is defined by 4 parameters:

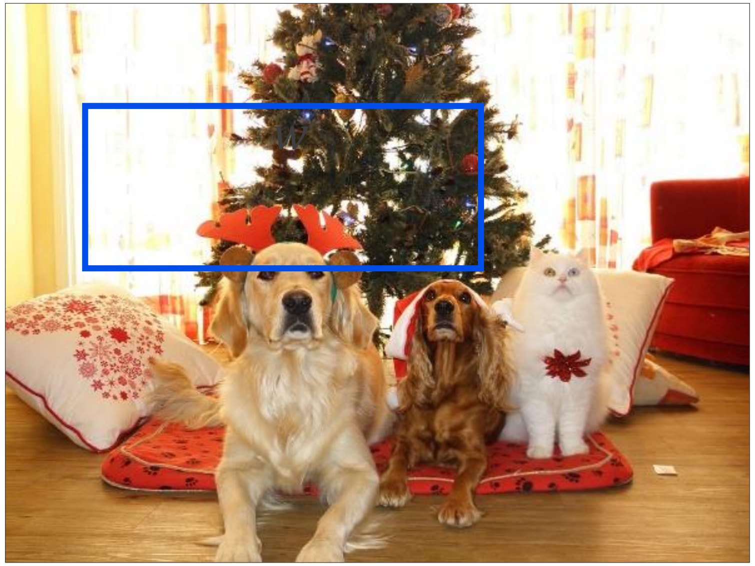
- center (x, y)
- height h
- width w

and, we assign a class label c to this box

Naive approach : sample all possible crops on the image and evaluate the classification/regression network

MINES PARIS | PSL

Object Detection : Multi objects



a box is defined by 4 parameters:

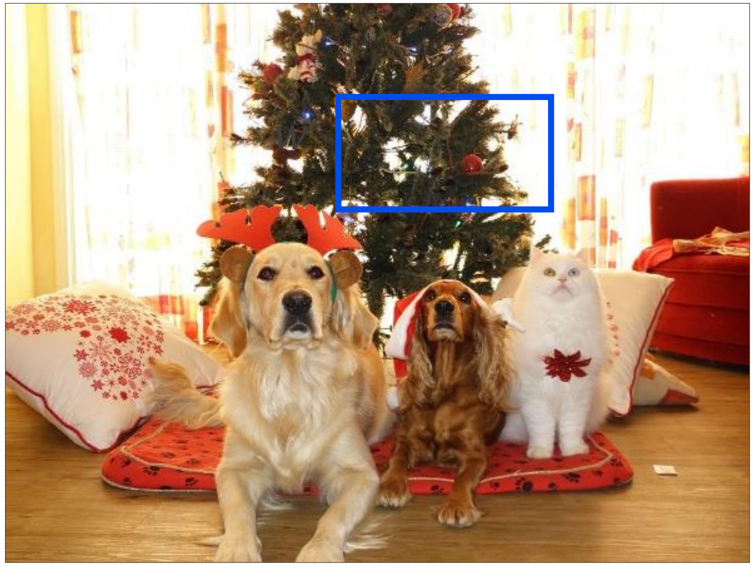
- center (x, y)
- height h
- width w

and, we assign a class label c to this box

Naive approach : sample all possible crops on the image and evaluate the classification/regression network

MINES PARIS | PSL

Object Detection : Multi objects



a box is defined by 4 parameters:

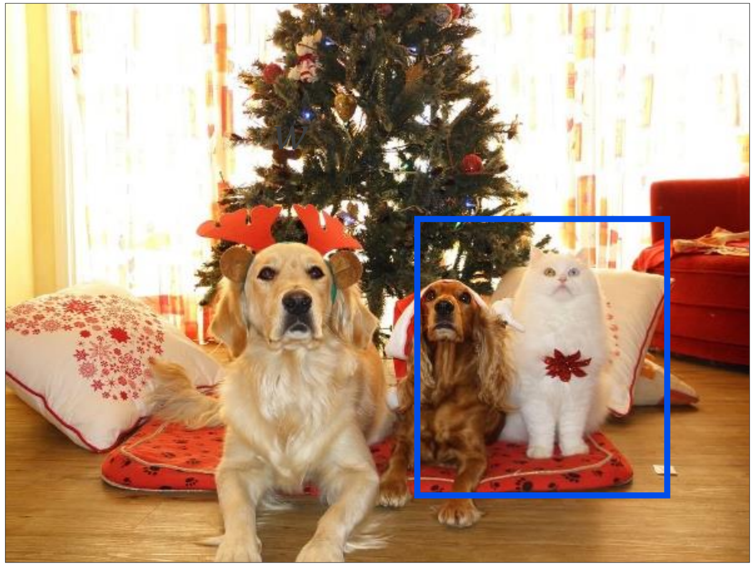
- center (x, y)
- height h
- width w

and, we assign a class label c to this box

Naive approach : sample all possible crops on the image and evaluate the classification/regression network

MINES PARIS | PSL

Object Detection : Multi objects



a box is defined by 4 parameters:

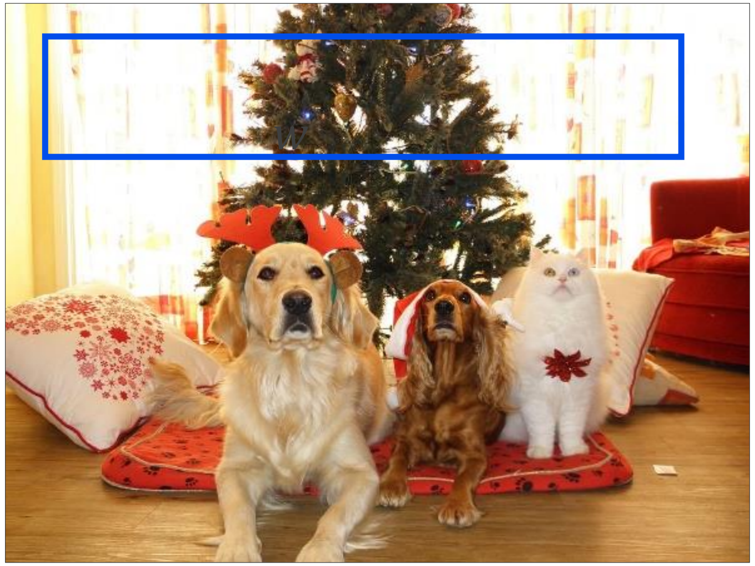
- center (x, y)
- height h
- width w

and, we assign a class label c to this box

Naive approach : sample all possible crops on the image and evaluate the classification/regression network

MINES PARIS | PSL

Object Detection : Multi objects



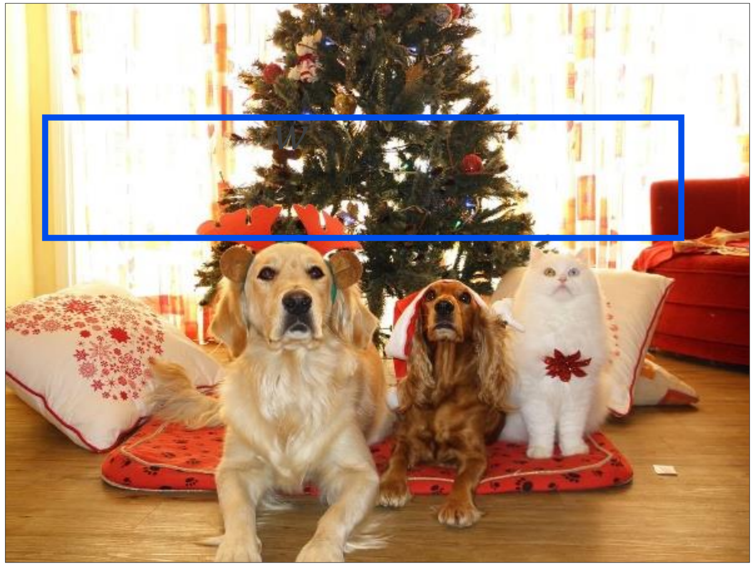
a box is defined by 4 parameters:

- center (x, y)
- height h
- width w

and, we assign a class label c to this box

Naive approach : sample all possible crops on the image and evaluate the classification/regression network

Object Detection : Multi objects



a box is defined by 4 parameters:

- center (x, y)
- height h
- width w

and, we assign a class label c to this box

Naive approach : sample all possible crops on the image and evaluate the classification/regression network

Object Detection : Multi objects

a box is defined by 4 parameters:

- center (x, y)
- height h
- width w

and, we assign a class label c to this box

Naive approach : sample all possible crops on the image and evaluate the classification/regression network

Object Detection : Region proposals

a box is defined by 4 parameters:

- center (x, y)
- height h
- width w

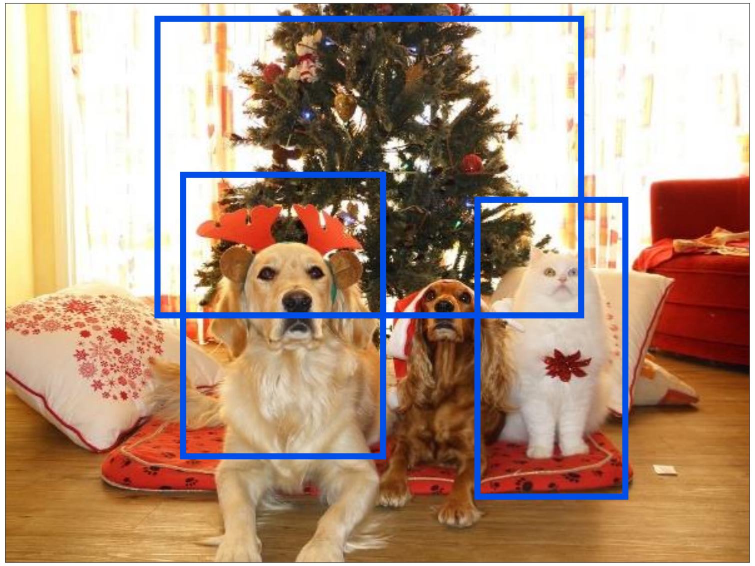
and, we assign a class label c to this box

Region of Interest (RoI) proposal approach : use SOTA algorithms (in 2015) to evaluate objectness of the image and propose a reduced set of patches

2012, Alexe et al, "Measuring the objectness of image windows"
2013, Uijlings et al, "Selective Search for Object Recognition"
2014, Zitnick and Dollar, "Edge boxes: Locating object proposals from edges"

MINES PARIS | PSL

Object Detection : Region of Interest proposals



a box is defined by 4 parameters:

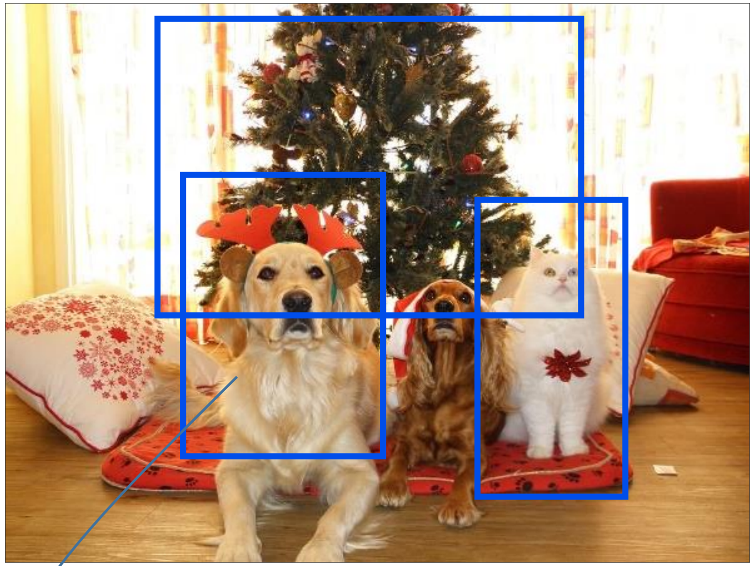
- center (x, y)
- height h
- width w

and, we assign a class label c to this box

Region of Interest (RoI) proposal approach : use SOTA algorithms (in 2015) to evaluate objectness of the image and propose a reduced set of patches

2012, Alexe et al, "Measuring the objectness of image windows"
2013, Uijlings et al, "Selective Search for Object Recognition"
2014, Zitnick and Dollar, "Edge boxes: Locating object proposals from edges"

Object Detection : Region of Interest proposals

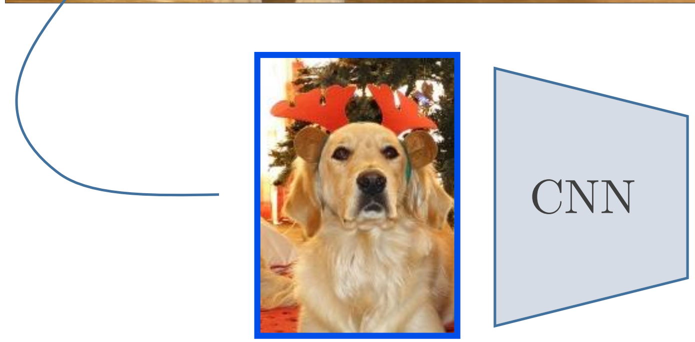


a box is defined by 4 parameters:

- center (x, y)
- height h
- width w

and, we assign a class label c to this box

Region of Interest (RoI) proposal approach : use SOTA algorithms (in 2015) to evaluate objectness of the image and propose a reduced set of patches

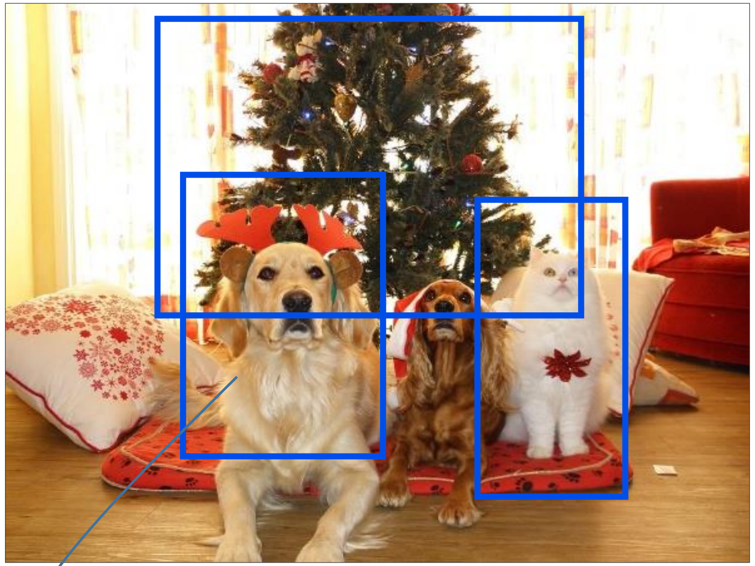


class vector $z \in \mathbb{R}^C$

box correction (x', y', h', w')

2012, Alexe et al, "Measuring the objectness of image windows"
2013, Uijlings et al, "Selective Search for Object Recognition"
2014, Zitnick and Dollar, "Edge boxes: Locating object proposals from edges"

Object Detection : Region of Interest proposals



a box is defined by 4 parameters:

- center (x, y)
- height h
- width w

and, we assign a class label c to this box

Region of Interest (RoI) proposal approach : use SOTA algorithms (in 2015) to evaluate objectness of the image and propose a reduced set of patches

class vector $z \in \mathbb{R}^C$

box correction (x', y', h', w')

Problem: still super slow !

Need to run a CNN a few 1000 times per image

2012, Alexe et al, "Measuring the objectness of image windows"
2013, Uijlings et al, "Selective Search for Object Recognition"
2014, Zitnick and Dollar, "Edge boxes: Locating object proposals from edges"

Object Detection : From R-CNN to RetinaNet

1. R-CNN to Fast R-CNN
2. Fast R-CNN to Faster R-CNN
3. Faster R-CNN to RetinaNet

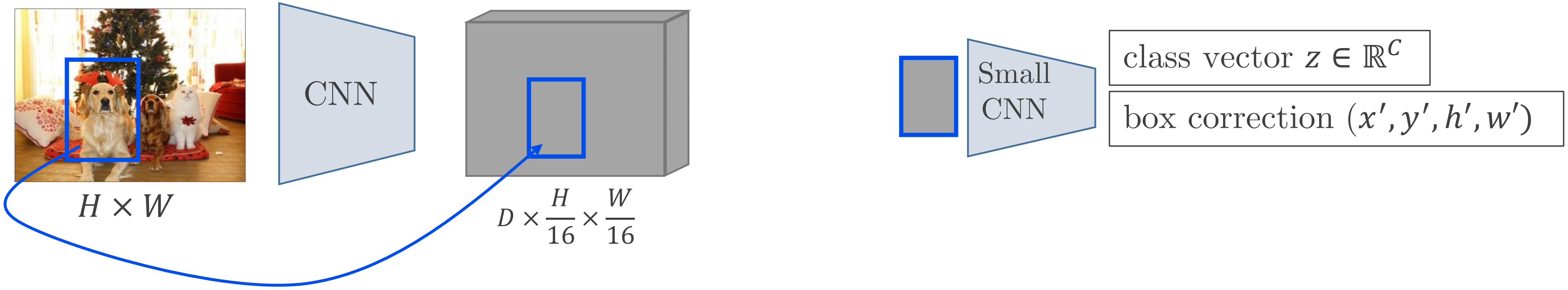
2015, Girshick, Fast- R-CNN

Object Detection : RoI pooling

1. R-CNN to Fast R-CNN : run a CNN one time on the image, and extract the patches from the feature map not the image. Then small CNN for classification and box regression

2015, Girshick, Fast- R-CNN

1. R-CNN to Fast R-CNN : run a CNN one time on the image, and extract the patches from the feature map not the image. Then small CNN for classification and box regression



Project RoI proposal onto feature map : RoI pooling

2015, Girshick, Fast- R-CNN

Object Detection : Region Proposal Network

1. R-CNN to Fast R-CNN : run a CNN one time on the image, and extract the patches from the feature map not the image. Then small CNN for classification and box regression
2. From Fast R-CNN to Faster R-CNN : Use a Region Proposal Network (instead of handcrafted algorithm) to propose RoI

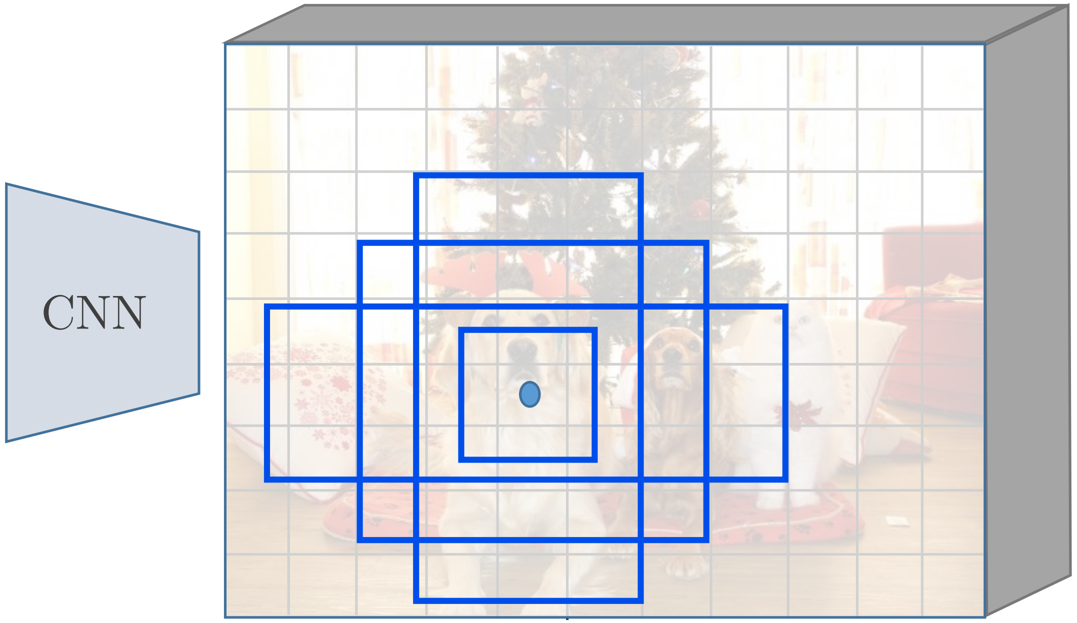
MINES PARIS | PSL

Object Detection : Anchor-based

1. R-CNN to Fast R-CNN : run a CNN one time on the image, and extract the patches from the feature map not the image. Then small CNN for classification and box regression
2. From Fast R-CNN to Faster R-CNN : Use a Region Proposal Network (instead of handcrafted algorithm) to propose RoI

Associate K anchor boxes of different size/scale at each point

$H \times W$

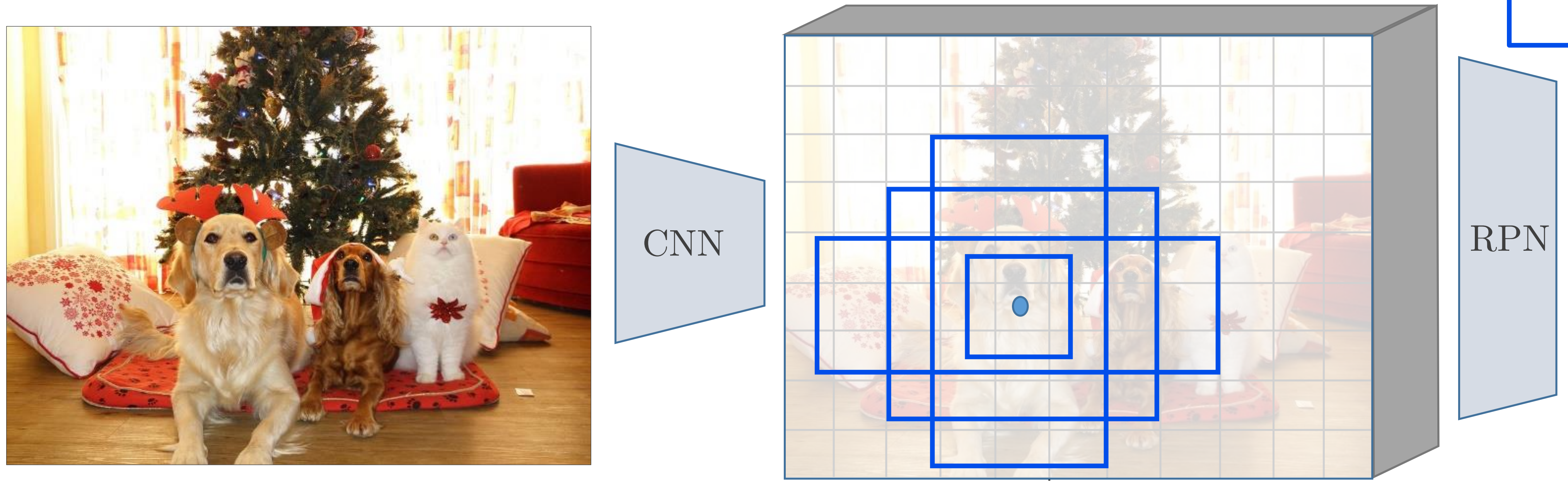


$D \times \frac{H}{16} \times \frac{W}{16}$

Object Detection : Anchor-based

1. R-CNN to Fast R-CNN : run a CNN one time on the image, and extract the patches from the feature map not the image. Then small CNN for classification and box regression
2. From Fast R-CNN to Faster R-CNN : Use a Region Proposal Network (instead of handcrafted algorithm) to propose RoI

Associate K anchor boxes of different size/scale at each point



$H \times W$

$D \times \frac{H}{16} \times \frac{W}{16}$

Objectness ?

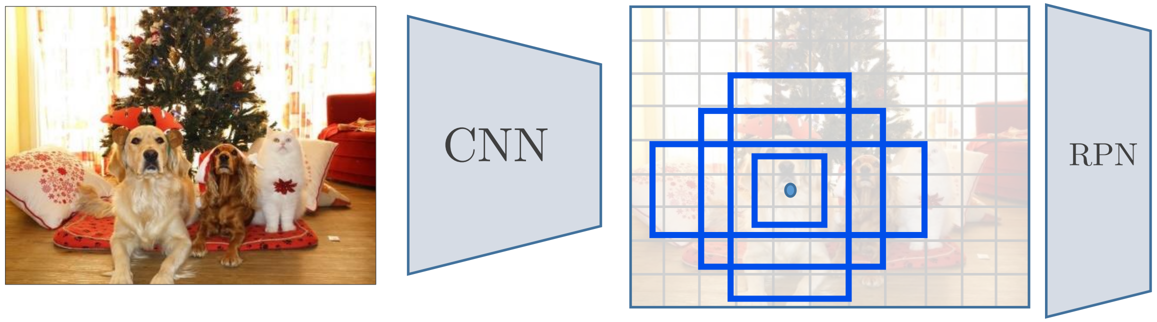
$K \times \frac{H}{16} \times \frac{W}{16}$

Box correction

$4K \times \frac{H}{16} \times \frac{W}{16}$

Object Detection : Anchor-based RPN

1. R-CNN to Fast R-CNN : run a CNN one time on the image, and extract the patches from the feature map not the image. Then small CNN for classification and box regression
2. From Fast R-CNN to Faster R-CNN : Use a Region Proposal Network (instead of handcrafted algorithm) to propose RoI



$H \times W$

$D \times \frac{H}{16} \times \frac{W}{16}$

Objectness ?

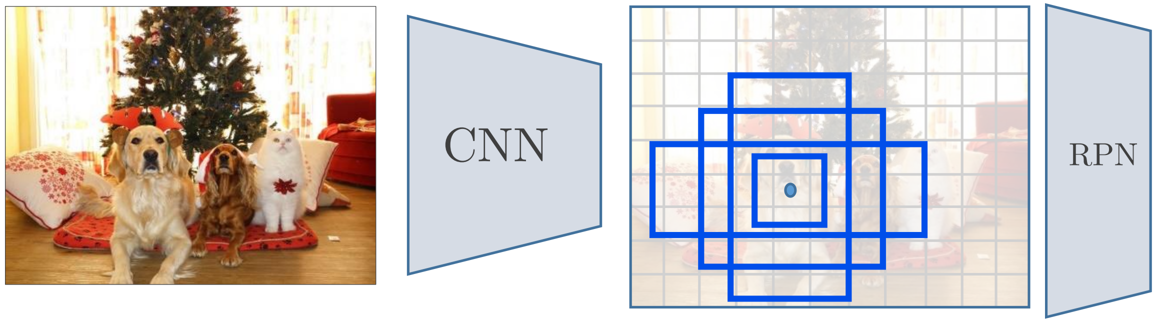
$K \times \frac{H}{16} \times \frac{W}{16}$

Box correction

$4K \times \frac{H}{16} \times \frac{W}{16}$

2016, Ren et al, Faster R-CNN

1. R-CNN to Fast R-CNN : run a CNN one time on the image, and extract the patches from the feature map not the image. Then small CNN for classification and box regression
2. From Fast R-CNN to Faster R-CNN : Use a Region Proposal Network (instead of handcrafted algorithm) to propose RoI



$H \times W$

$D \times \frac{H}{16} \times \frac{W}{16}$

Objectness ?

$K \times \frac{H}{16} \times \frac{W}{16}$

Box correction

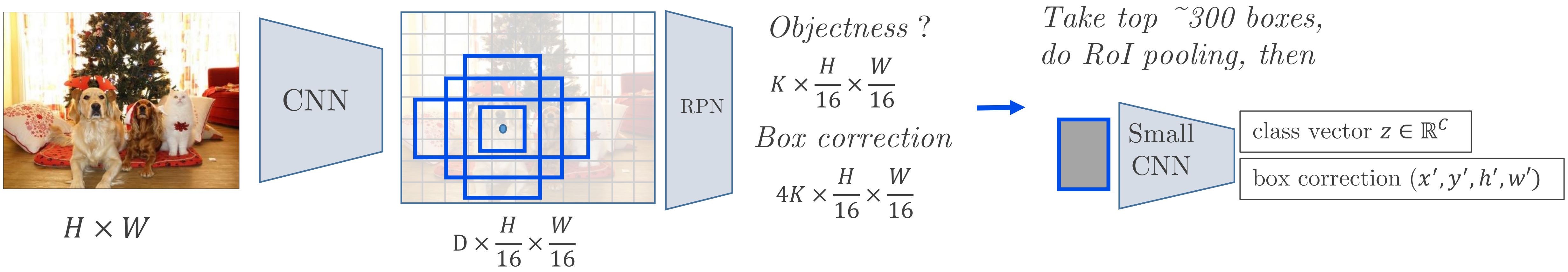
$4K \times \frac{H}{16} \times \frac{W}{16}$

→ *Take top ~300 boxes, do RoI pooling, then*

2016, Ren et al, Faster R-CNN

Object Detection : Anchor-based RPN

1. R-CNN to Fast R-CNN : run a CNN one time on the image, and extract the patches from the feature map not the image. Then small CNN for classification and box regression
2. From Fast R-CNN to Faster R-CNN : Use a Region Proposal Network (instead of handcrafted algorithm) to propose RoI



2016, Ren et al, Faster R-CNN

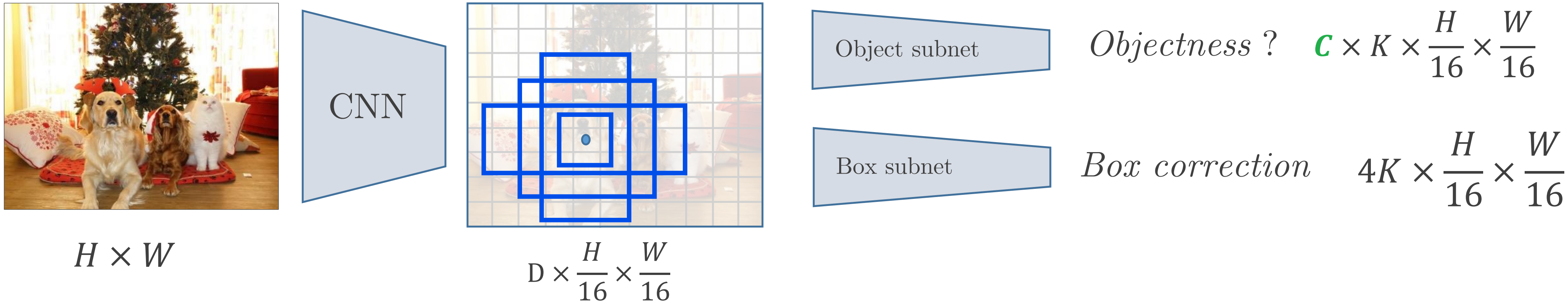
Object Detection : RetinaNat – One stage detector

1. R-CNN to Fast R-CNN : run a CNN one time on the image, and extract the patches from the feature map not the image. Then small CNN for classification and box regression
2. From Fast R-CNN to Faster R-CNN : Use a Region Proposal Network (instead of handcrafted algorithm) to propose RoI
3. From Faster R-CNN to RetinaNet : Remove the RoI pooling and do a 1-stage approach

2018, Lin et al, Focal Loss for Dense Object Detection

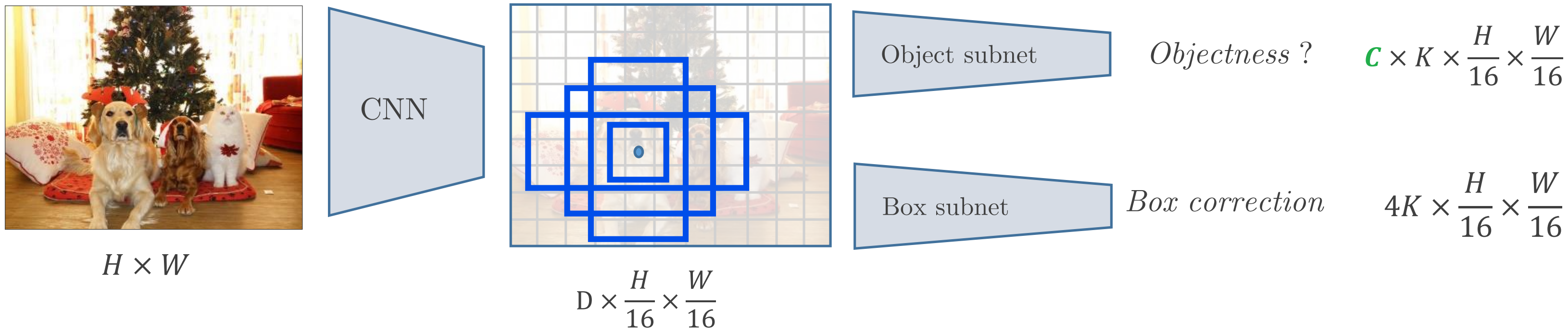
Object Detection : RetinaNat – One stage detector

1. R-CNN to Fast R-CNN : run a CNN one time on the image, and extract the patches from the feature map not the image. Then small CNN for classification and box regression
2. From Fast R-CNN to Faster R-CNN : Use a Region Proposal Network (instead of handcrafted algorithm) to propose RoI
3. From Faster R-CNN to RetinaNet : Remove the RoI pooling and do a 1-stage approach



2018, Lin et al, Focal Loss for Dense Object Detection

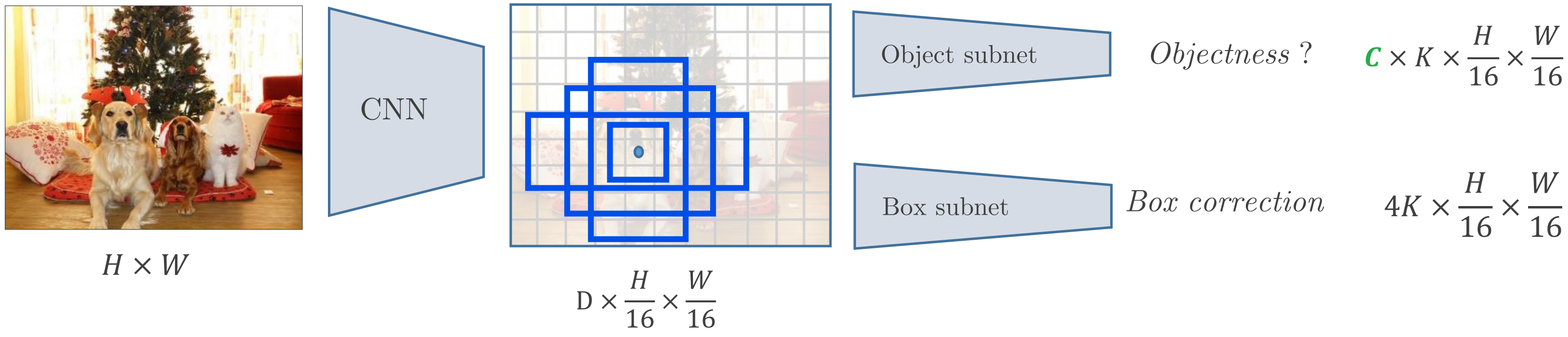
Object Detection : Non-Maximum Suppresion



Skipped a lot of modelisation details :

- NMS or *non-maximum suppression*

Object Detection : Non-Maximum Suppresion

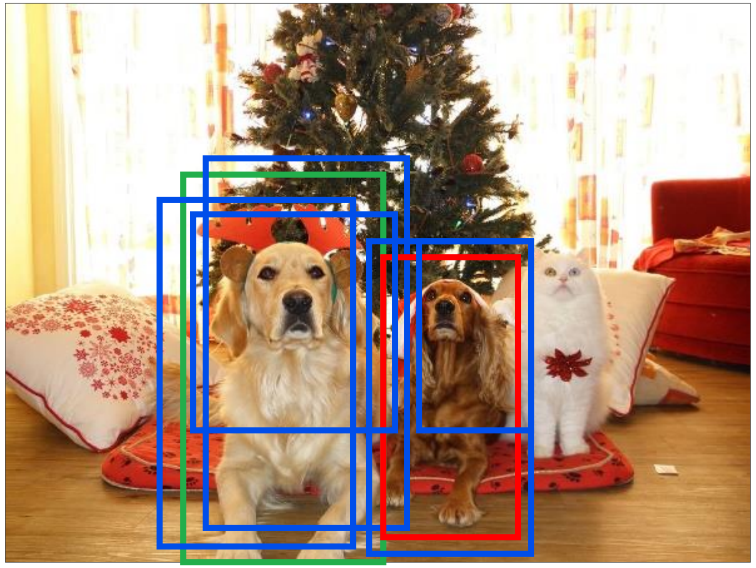


Skipped a lot of modelisation details :

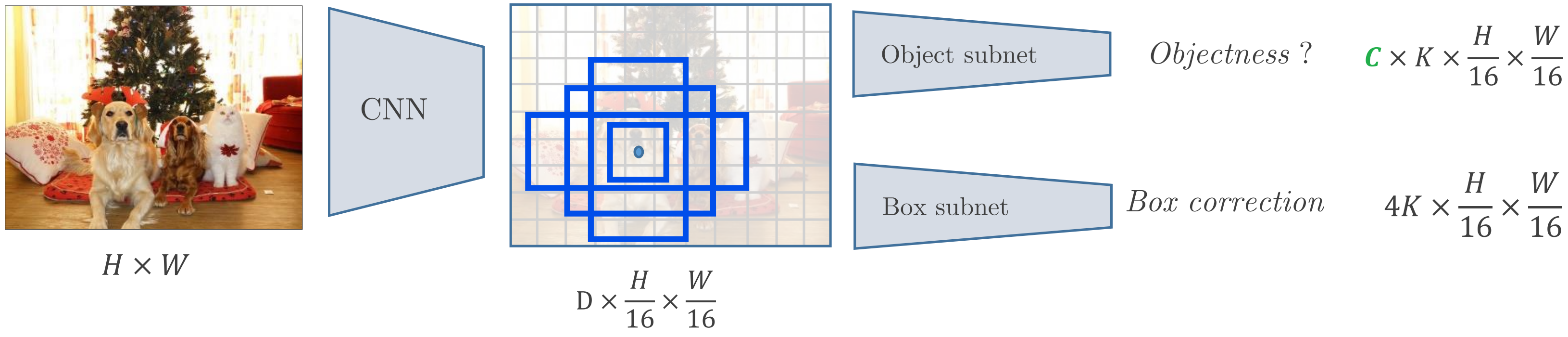
- NMS or *non-maximum suppression*

Algorithm 1 Non-Max Suppression

```
procedure NMS(B,c)
    B_nms ← ∅
    for b_i ∈ B do
        discard ← False
        for b_j ∈ B do
            if IoU(b_i, b_j) > λ_nms then
                if score(c, b_j) > score(c, b_i) then
                    discard ← True
        if not discard then
            B_nms ← B_nms ∪ b_i
    return B_nms
```

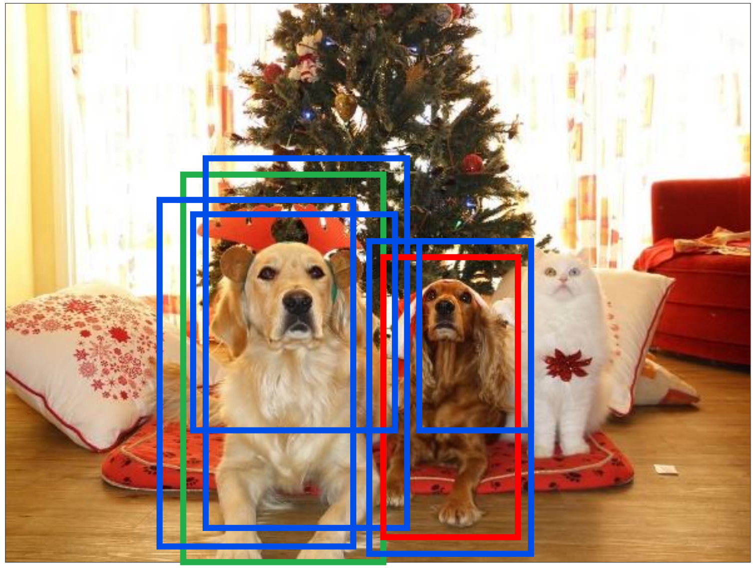


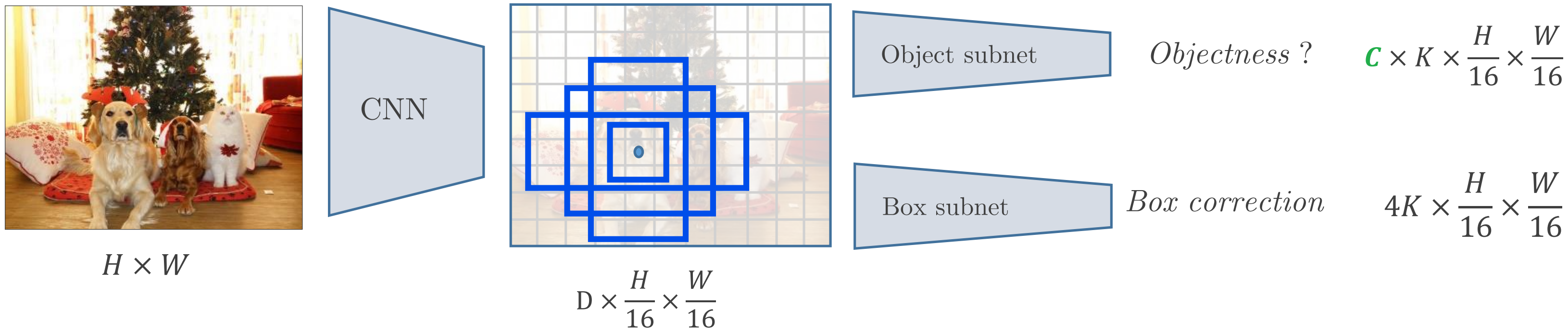
Object Detection : Non-Maximum Suppresion



Skipped a lot of modelisation details :

- NMS or *non-maximum suppression*



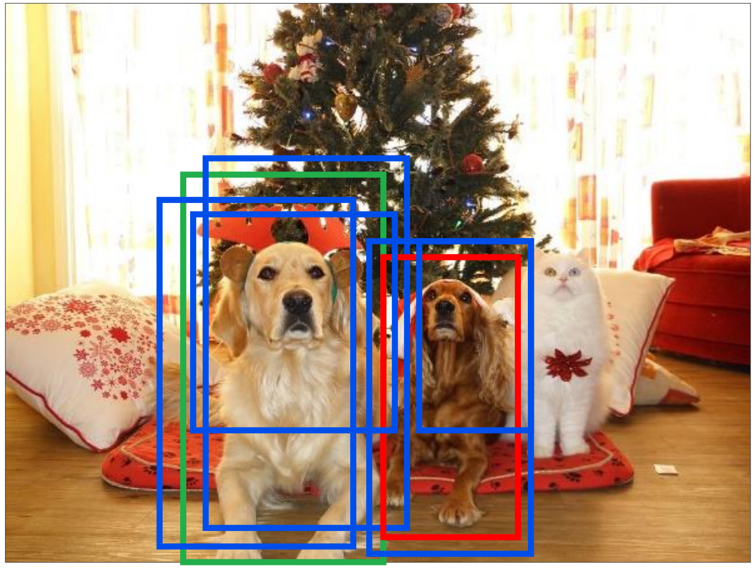


Skipped a lot of modelisation details :

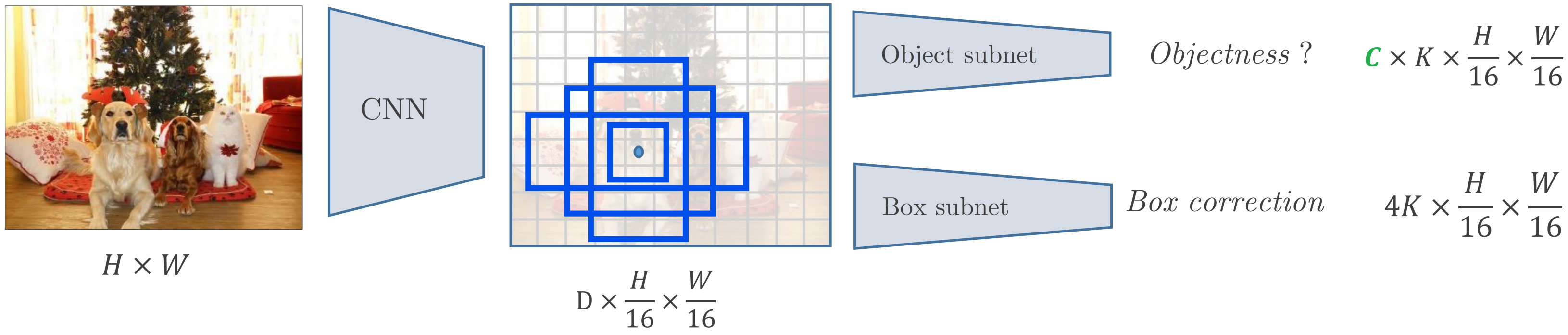
- NMS or *non-maximum suppression*

Algorithm 1 Non-Max Suppression

procedure NMS(B,c)
  $B_{nms} \leftarrow \emptyset$
  for $b_i \in B$ **do**
    $discard \leftarrow$ False
    for $b_j \in B$ **do**
      if IoU $(b_i, b_j) > \lambda_{\text{nms}}$ **then**
        if score$(c, b_j) >$ score(c, b_i) **then**
          $discard \leftarrow$ True
    if not $discard$ **then**
      $B_{nms} \leftarrow B_{nms} \cup b_i$
  return B_{nms}



Object Detection : Non-Maximum Suppresion

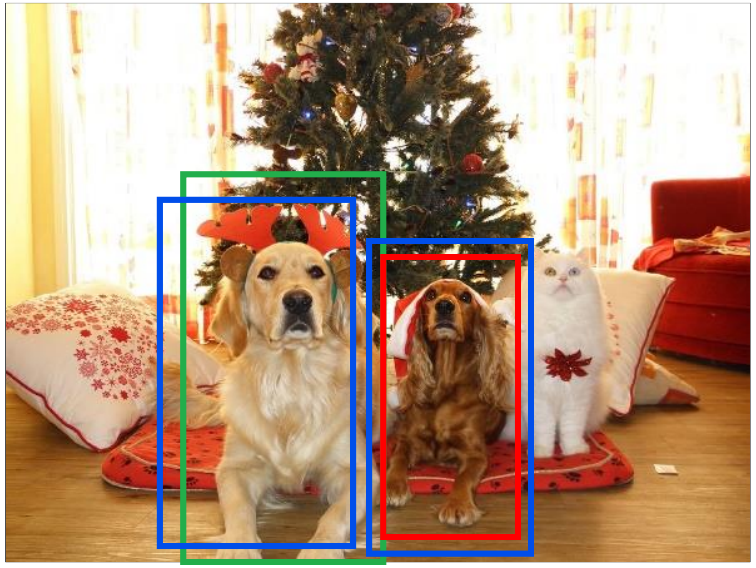


Skipped a lot of modelisation details :

- NMS or *non-maximum suppression*

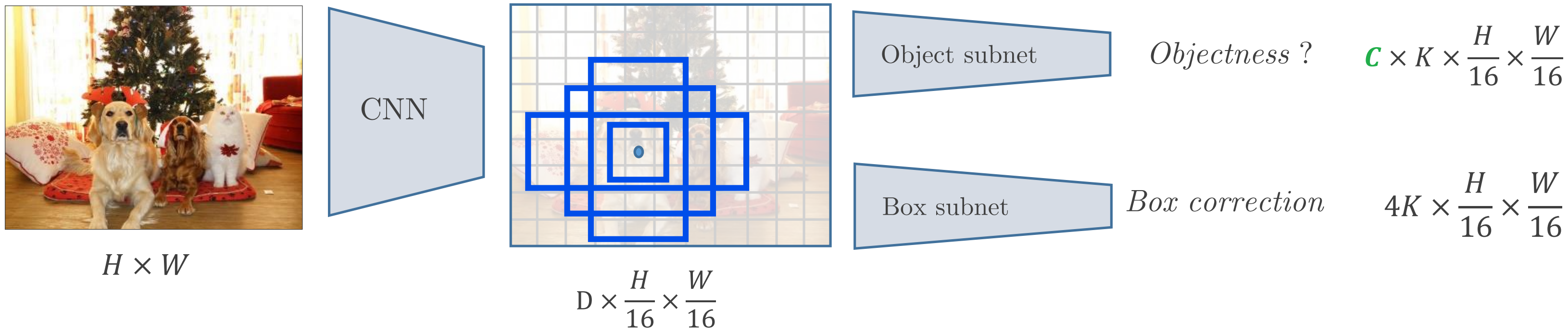
Algorithm 1 Non-Max Suppression

procedure NMS(B,c)
  $B_{nms} \leftarrow \emptyset$
  for $b_i \in B$ **do**
    $discard \leftarrow$ False
    for $b_j \in B$ **do**
      if **IoU** $(b_i, b_j) > \lambda_{\text{nms}}$ **then**
        if score$(c, b_j) >$ score(c, b_i) **then**
          $discard \leftarrow$ True
    if not $discard$ **then**
      $B_{nms} \leftarrow B_{nms} \cup b_i$
  return B_{nms}



MINES PARIS | PSL

Object Detection : Non-Maximum Suppresion

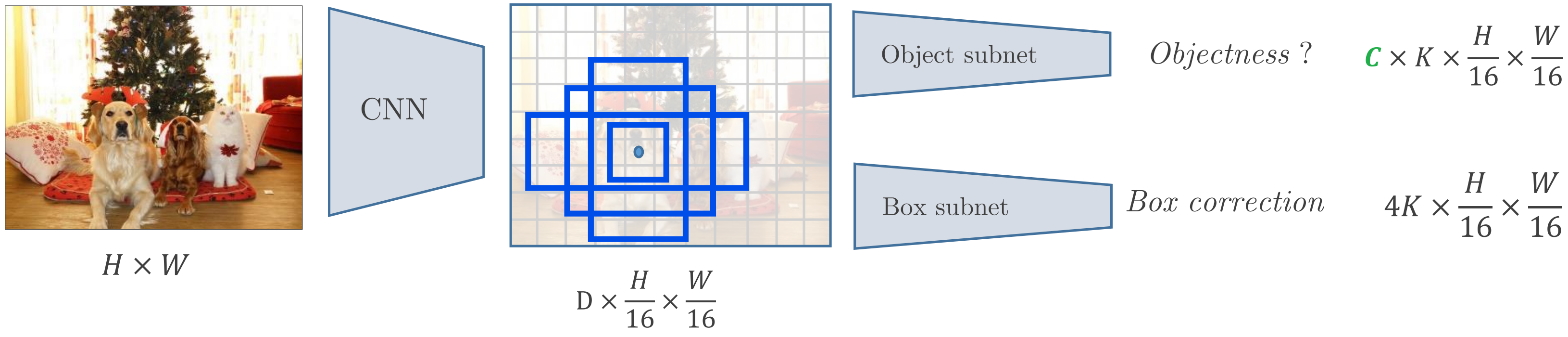


Skipped a lot of modelisation details :

- NMS or *non-maximum suppression*

2016, Shrivastava et al, Training region based object detectors with online hard example mining
2018, Lin et al, Focal Loss for Dense Object Detection

Object Detection : Non-Maximum Suppresion

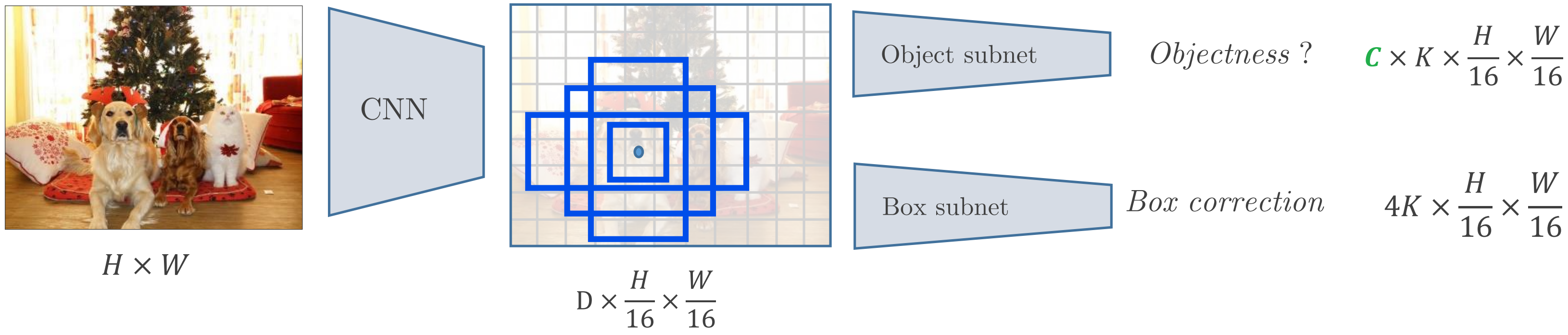


Skipped a lot of modelisation details :

- NMS or *non-maximum suppression*
- Solve *imbalance* between negative and positive anchors, *i.e* among K~50000 anchors most of them do not bind an object

2016, Shrivastava et al, Training region based object detectors with online hard example mining
2018, Lin et al, Focal Loss for Dense Object Detection

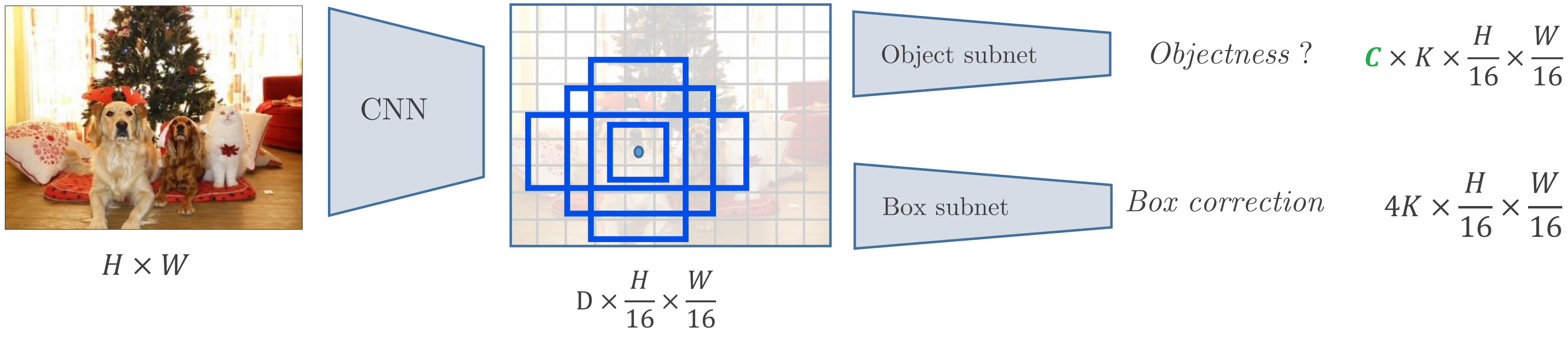
Object Detection : Foreground-Background imbalance



Skipped a lot of modelisation details :

- NMS or *non-maximum suppression*
- Solve *imbalance* between negative and positive anchors, *i.e* among K~50000 anchors most of them do not bind an object
  - Sampling heuristics: fixed foreground-background ratio of anchors, online hard example mining

2016, Shrivastava et al, Training region based object detectors with online hard example mining
2018, Lin et al, Focal Loss for Dense Object Detection

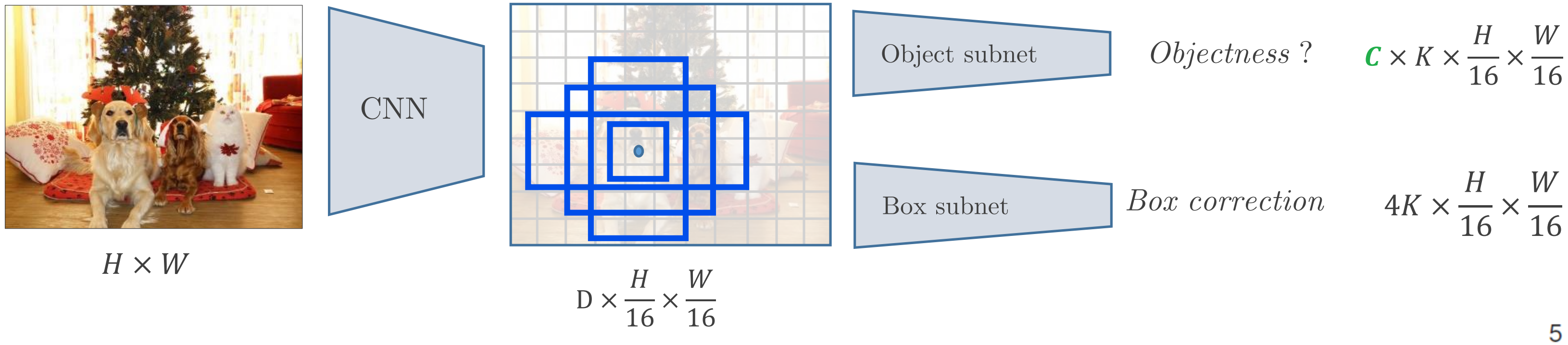


Skipped a lot of modelisation details :

- NMS or *non-maximum suppression*
- Solve *imbalance* between negative and positive anchors, *i.e* among K~50000 anchors most of them do not bind an object
  - Sampling heuristics: fixed foreground-background ratio of anchors, online hard example mining
  - Focal loss for 1-stage detectors is the turning point

2016, Shrivastava et al, Training region based object detectors with online hard example mining
2018, Lin et al, Focal Loss for Dense Object Detection

Object Detection : Focal loss



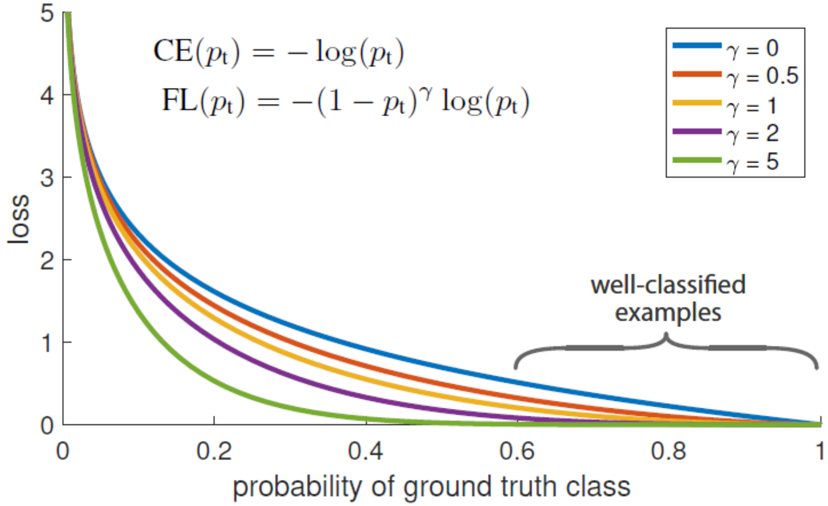


Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1 - p_t)^\gamma$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclassified examples. As our experiments will demonstrate, the proposed focal loss enables training highly accurate dense object detectors in the presence of vast numbers of easy background examples.

Skipped a lot of modelisation details :

- NMS or *non-maximum suppression*
- Solve *imbalance* between negative and positive anchors, *i.e* among K~50000 anchors most of them do not bind an object
  - Sampling heuristics: fixed foreground-background ratio of anchors, online hard example mining
  - Focal loss for 1-stage detectors is the turning point

2016, Shrivastava et al, Training region based object detectors with online hard example mining
2018, Lin et al, Focal Loss for Dense Object Detection

Object Detection : Other architectures

Subsequent anchor-based methods:

- 2017, He et al, Mask R-CNN → perform instance and semantic segmentation
- 2020, Bochkovskiy et al, YOLOv4: Optimal Speed and Accuracy of Object Detection
- 2020, Tan et al, EfficientDet: Scalable and Efficient Object Detection

Object Detection : Other architectures

Subsequent anchor-based methods:

- 2017, He et al, Mask R-CNN → perform instance and semantic segmentation
- 2020, Bochkovskiy et al, YOLOv4: Optimal Speed and Accuracy of Object Detection
- 2020, Tan et al, EfficientDet: Scalable and Efficient Object Detection

---

Anchor-Free detectors : avoid cumbersome modelisation, NMS, etc..

CNN-based architecture :

- 2019, Tian et al, FCOS: Fully Convolutional One-Stage Object Detection
- 2019, Zhou et al, Objects as points.

Object Detection : Other architectures

Subsequent anchor-based methods:

- 2017, He et al, Mask R-CNN → perform instance and semantic segmentation
- 2020, Bochkovskiy et al, YOLOv4: Optimal Speed and Accuracy of Object Detection
- 2020, Tan et al, EfficientDet: Scalable and Efficient Object Detection

---

Anchor-Free detectors : avoid cumbersome modelisation, NMS, etc..

CNN-based architecture :

- 2019, Tian et al, FCOS: Fully Convolutional One-Stage Object Detection
- 2019, Zhou et al, Objects as points.

Transformers-based:

- 2020, Carion et al, DETR: End-to-End Object Detection with Transformers
- 2023, Zong et al, DETRs with Collaborative Hybrid Assignments Training

MINES PARIS | PSL

Thank you for your attention

*slides adapted from CS231n