UNIVERSITÉ CÔTE D'AZUR

Object Detection in Deep Learning -I

Ujjwal

INRIA

19^{th} Jan of 2021

informatics mathematics

UNIVERSITÉ CÔTE D'AZUR

Outline

informatics mathematics

What is Object Detection ?

$$f : \mathcal{X} \longrightarrow \mathcal{Y}$$

- $\mathcal{X} \subset \mathbb{R}^{m \times n}$: Set of images.
- $\mathcal{Y} \subset T^M$: Target set where M is the maximum possible number of detections in $\mathcal{X}$. $T = C \times B$ where,
  - $C = [0, 1]^{N+1}$: Confidence set of detections where N is total number of object classes. $+\mathbf{1}$ is for the background class.
    - $\tilde{c}_i \in C$ denotes the confidence of i^{th} class being present.
  - $B = \mathbb{R}^4$: Bounding box set of detections.
- $C \times B$ denotes that object detection is a unification of classification and localization in machine learning.

What is Object Detection ?

Compositional View

A mapping f depicting object detection can be written as a composition :

- $f = f_2 \circ f_1 \triangleq f_2(f_1(.))$
- $f_1 : \mathcal{X} \longrightarrow C$
  - **Classification**: Which objects are present in $\mathcal{X}$?
- $f_2 : \mathcal{X} \longrightarrow B$
  - **Regression**: Where are the objects present in $\mathcal{X}$?

UNIVERSITÉ CÔTE D'AZUR

Outline

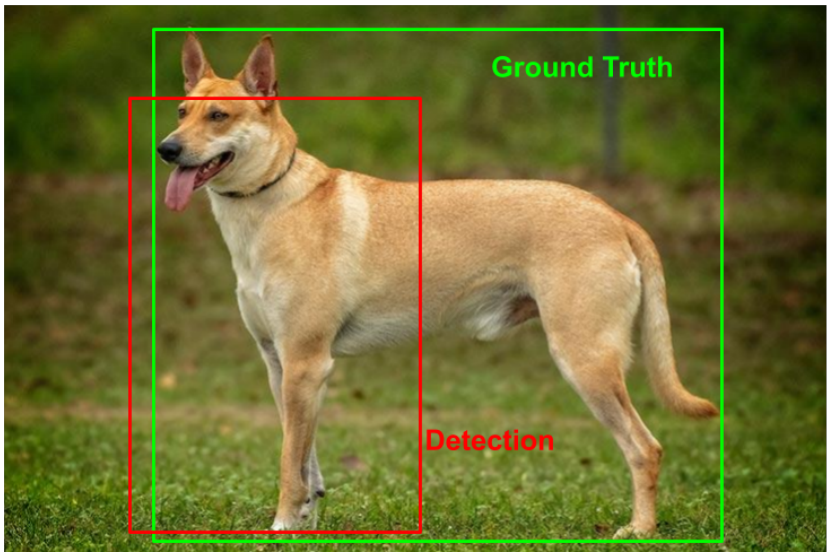
informatics mathematics

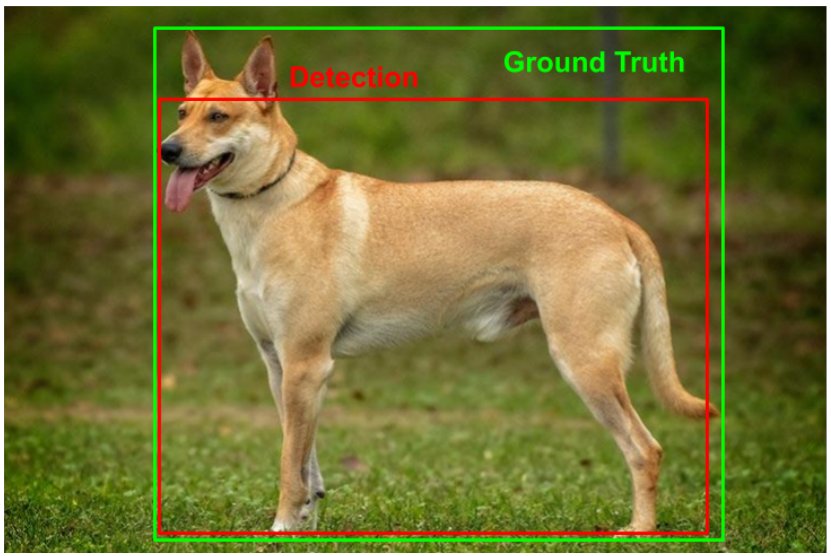
Fundamentals

Ground Truth	**Detector Output**	
	Present	**Absent**
Present	True Positive (TP)	False Negative (FN)
Absent	False Positive (FP)	True Negative (TN)

Detection result interpretation based on an object's presence in **Ground Truth** (GT) and **Detector output**.

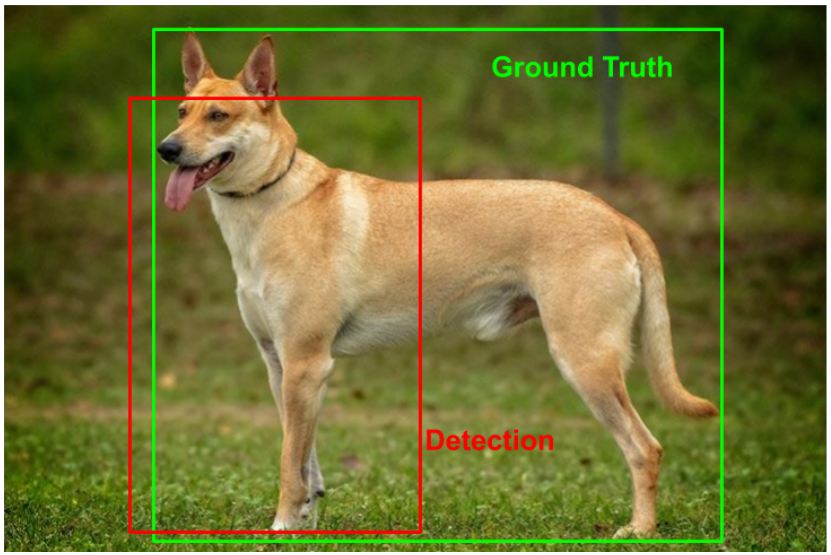
Fundamentals

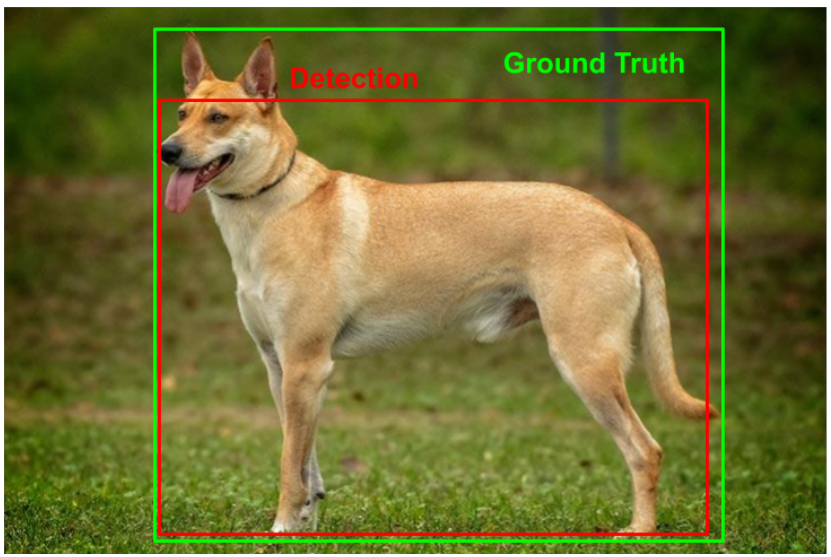




We consider two examples to demonstrate the relevance of localization in object detection.

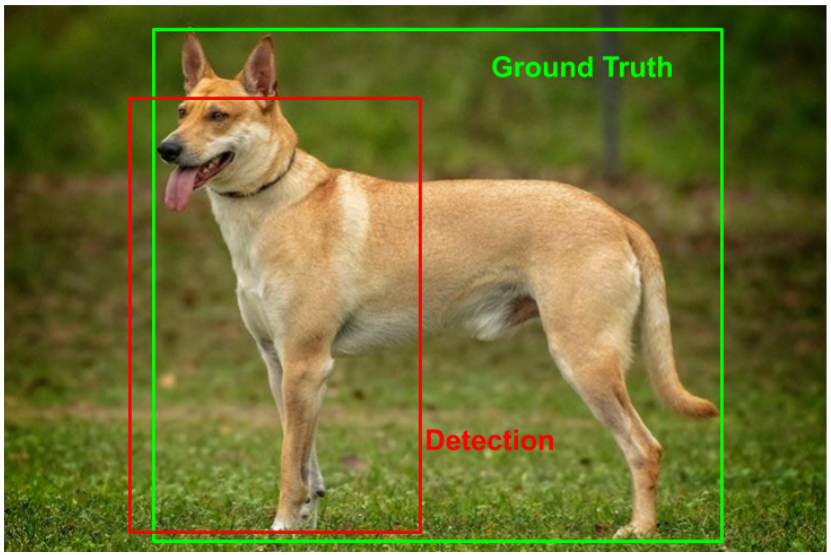
Fundamentals





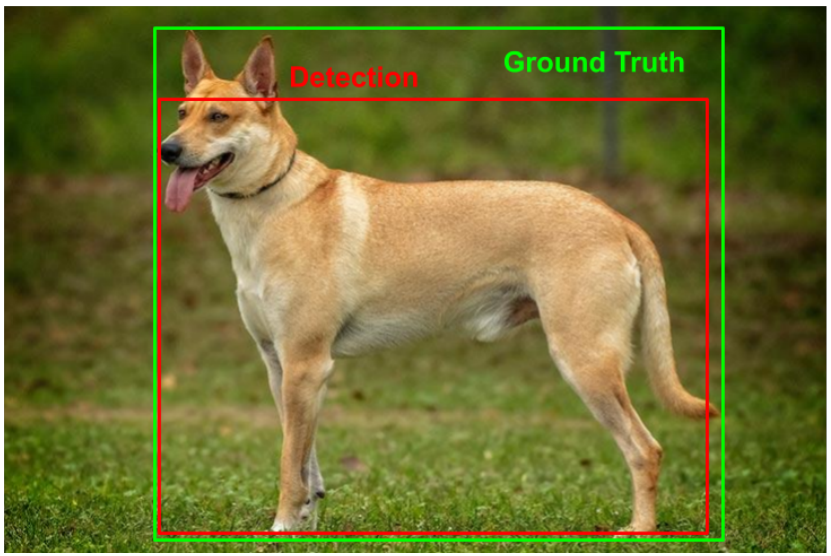
We assume that a bounding box with a label of "Dog" has been detected.

Fundamentals



Bad Detection

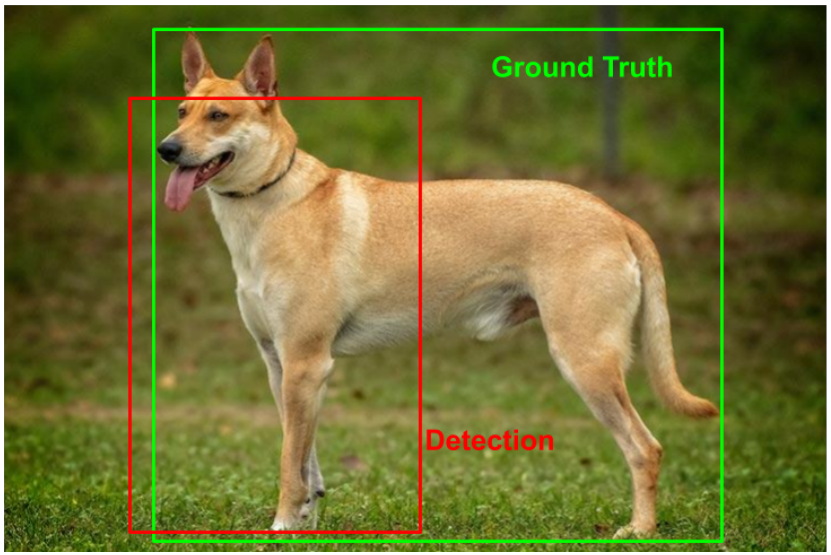
Detection overlap with GT is low.



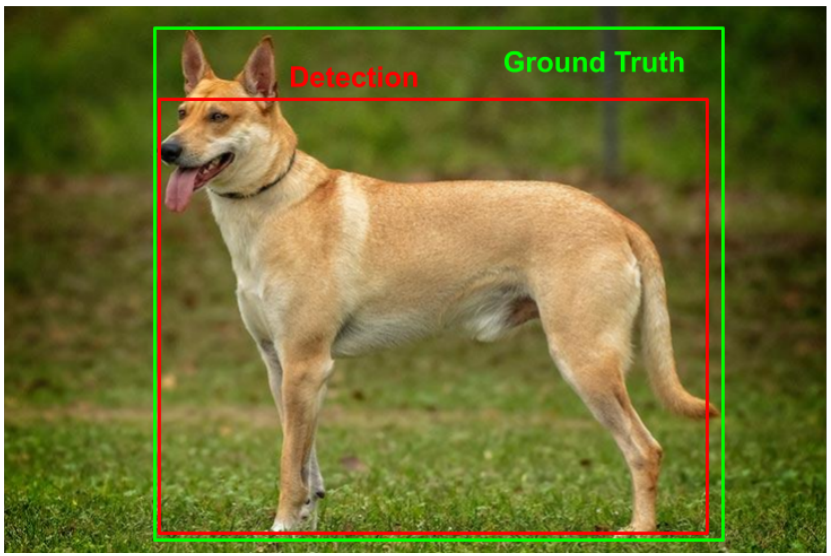
Good Detection

Detection overlap with GT is high.

Fundamentals



Dog is not detected. (FP)



Dog is detected. (TP)

UNIVERSITÉ CÔTE D'AZUR

Intersection-over-Union (IoU)

Given $B_1, B_2 \subset \mathbb{R}^4$,

- Let, $I = Area(B_1 \bigcap B_2)$
- Let, $U = Area(B_1 \bigcup B_2)$
- $IoU \triangleq \frac{I}{U}$
- $IoU \in [0, 1]$

IoU $\propto$ Agreement with the GT.

informatics mathematics

Intersection-over-Union (IoU)

Given $B_1, B_2 \subset \mathbb{R}^4$,

- Let, $I = Area(B_1 \bigcap B_2)$
- Let, $U = Area(B_1 \bigcup B_2)$
- $IoU \triangleq \frac{I}{U}$
- $IoU \in [0, 1]$

IoU $\propto$ Agreement with the GT.

Precision and Recall

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Precision and Recall

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Of all detected objects how many are actually matching with GT

Precision and Recall

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Of all detected objects how many are actually matching with GT

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

Precision and Recall

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Of all detected objects how many are actually matching with GT

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

Of all GT objects how many are actually detected.

Precision and Recall

High Precision and **high recall** are desirable traits of an object detector.

Interpreting Object Detector Output

Given an image $x \in \mathcal{X}$, $f(x) = \{(c_i, b_i)\}_{i=1}^{M}$, $c_i \in C$ and $b_i \in B$.

Interpreting Object Detector Output

Given an image $x \in \mathcal{X}$, $f(x) = \{(c_i, b_i)\}_{i=1}^{M}$, $c_i \in C$ and $b_i \in B$.

$c_i \in [0, 1]^{N+1}$ denotes the probability of different object classes enclosed by b_i.

Interpreting Object Detector Output

Given an image $x \in \mathcal{X}$, $f(x) = \{(c_i, b_i)\}_{i=1}^{M}$, $c_i \in C$ and $b_i \in B$.

$c_i \in [0,1]^{N+1}$ denotes the probability of different object classes enclosed by b_i.

The object class enclosed by b_i is determined by
$\mathcal{C} = \text{argmax}_j\{o_j \in c_i\}$

Interpreting Object Detector Output

Given an image $x \in \mathcal{X}$, $f(x) = \{(c_i, b_i)\}_{i=1}^{M}$, $c_i \in C$ and $b_i \in B$.

$c_i \in [0,1]^{N+1}$ denotes the probability of different object classes enclosed by b_i.

The object class enclosed by b_i is determined by
$\mathcal{C} = \text{argmax}_j\{o_j \in c_i\}$

The object class enclosed by a bounding box is the class whose probability of presence is highest.

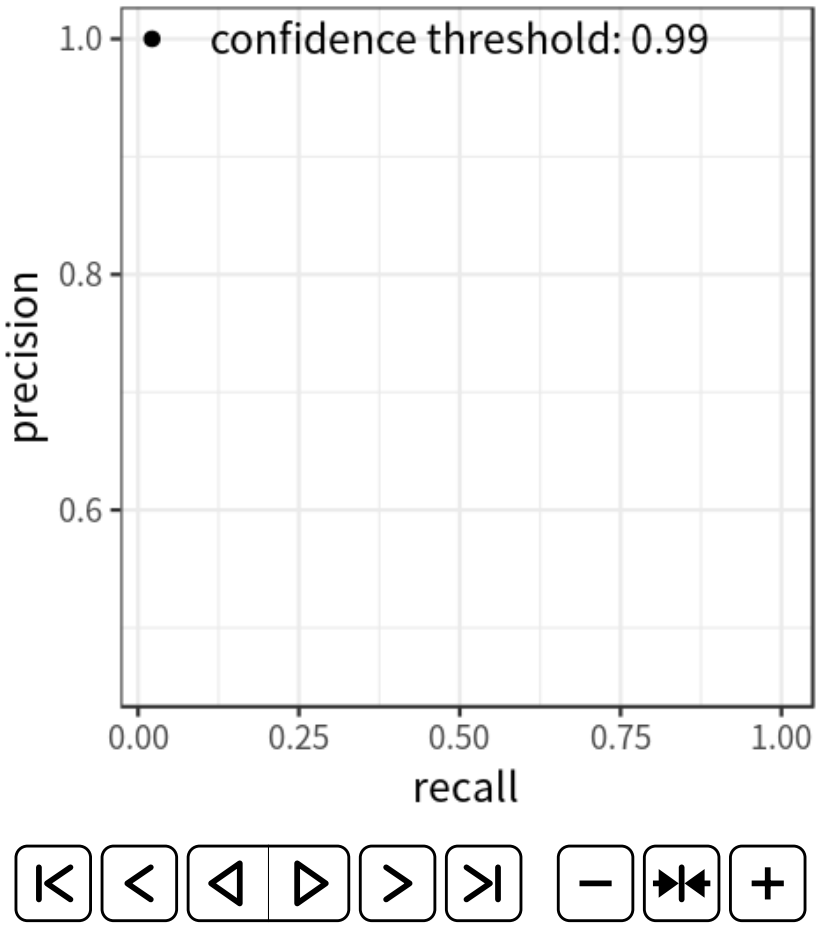
Interpreting Object Detector Output

Given an image $x \in \mathcal{X}$, the detector output $f(x)$ can be interpreted as follows

$$f(x) = \{(\phi_x(i), p_{\phi(i)}, b_i)\}_{i=1}^{M}$$

- M objects have been detected.
- $\phi_x : 1 \leq i \in \mathbb{N} \leq M \longrightarrow 1 \leq j \in \mathbb{N} \leq N$.
  - $\phi_x(i)$ is a mapping which assigns to i^{th} element of the output set for image x, a class label.
  - ϕ_x in general is a many-to-one mapping.
- $p_{\phi(i)}$ is the probability of the i^{th} object to have the class $\phi(i)$.
- b_i is the bounding box representing the i^{th} object.

Precision Recall Curve



Computing the Precision-Recall Curve

Algorithm 1: Determining a detection as TP, FP or FN.

Result: out

Given,

1. An element $d_i \in f(x) = \{(\phi_x(i), p_{\phi_x(i)}, b_i^{det})\}_{i=1}^{M}$ for an image x;
2. GT set $G = \{\phi_g(i), 1, b_i\}_{i=1}^{\sigma(x)}\}$, where $\sigma(x)$ is number of GT objects in image x.
3. Pre-defined IoU threshold T.

Compute $\boldsymbol{I_o} = \{I : I = IOU(d_i, g_i)\ , g_i \in G\}$ and $\boldsymbol{j} = \text{argmax}_j\ \boldsymbol{I_o}$.

if $\boldsymbol{I_o}(j) < T$ **then**
    out=None
else
    if $class(d_i) = class(g_j)$ **then**
        if $confidence(d_i) < 0.5$ **then**
            out = FN
        else
            out = TP
        end
    else
        out = FP
    end
end

Computing the Precision-Recall Curve

Algorithm 2: How to write algorithms

Result: Write here the result

Given,

1. Detection set $\mathcal{D} = f(x) = \{(\phi(i), p_{\phi(i)}, b_i^{det})\}_{i=1}^{M}$ for an image x;
2. GT set.
3. Pre-defined IoU threshold T.
4. $\tau = 0.0$

while $\tau \leq 1$ **do**

  select $\tilde{\mathcal{D}} \subseteq \mathcal{D}$ $s.t.$ $p_{\phi(i)} \geq \tau \; \forall \; 1 \leq i \leq |\tilde{\mathcal{D}}|$;

  $\forall \; \tilde{d} \in \tilde{\mathcal{D}}$; Find $G \subseteq GT$, such that $IoU(\tilde{d}, g \in G) \geq T$;

end

UNIVERSITÉ CÔTE D'AZUR

Outline

informatics mathematics

Brief Overview

1. **Pre-processor**
2. **Detection Head**
3. **Post-Processor**
4. **Loss Function** (During training phase)

Pre-processor

Transforms an image into a feature map.

Detection Head

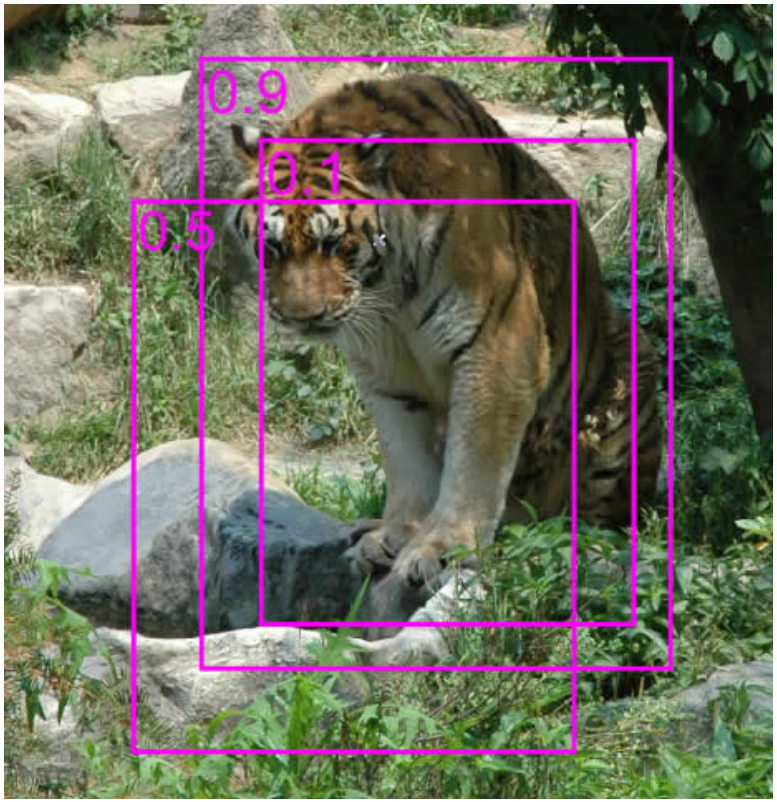
Detection head is made of two components:

1. **Classification Head**: Performs classification of a region in the feature map.
2. **Regression Head**: Performs bounding box regression over a region in the feature map.

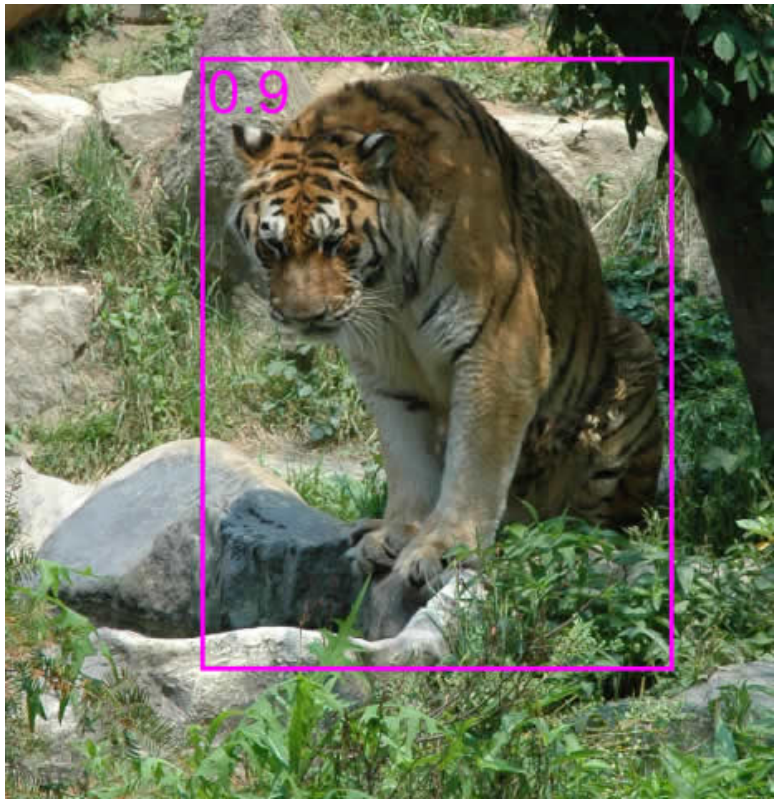
Post-Processor

Object detectors often give multiple detections for each object. Post-Processing aims at extracting the most relevant detection for each object.

Post-Processor



Before post-processing.



After post-processing.

Loss Function

Generally a detector loss function is of the following form:

$$f_{det} \triangleq f_{cls} + f_{reg} + f_{regularization} \tag{3}$$

Classification Losses

Some classification losses are:

1. Cross Entropy Loss.
2. Focal Loss.
3. Generalized Focal Loss.

Regression Losses

Some regression losses are:

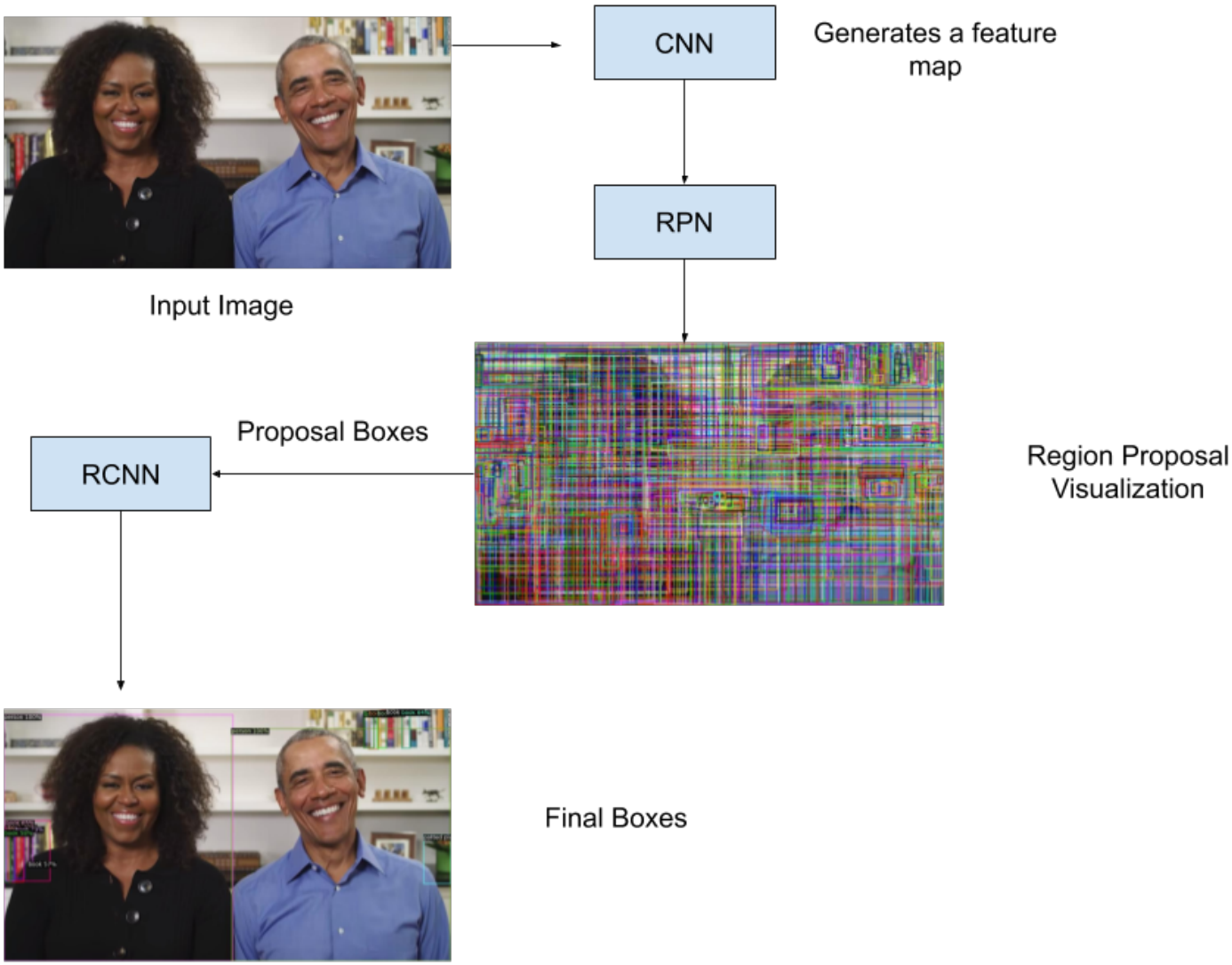
1. Smooth L1-Loss.
2. Repulsion Loss.

UNIVERSITÉ CÔTE D'AZUR

Outline

informatics mathematics

Faster-RCNN:Overview



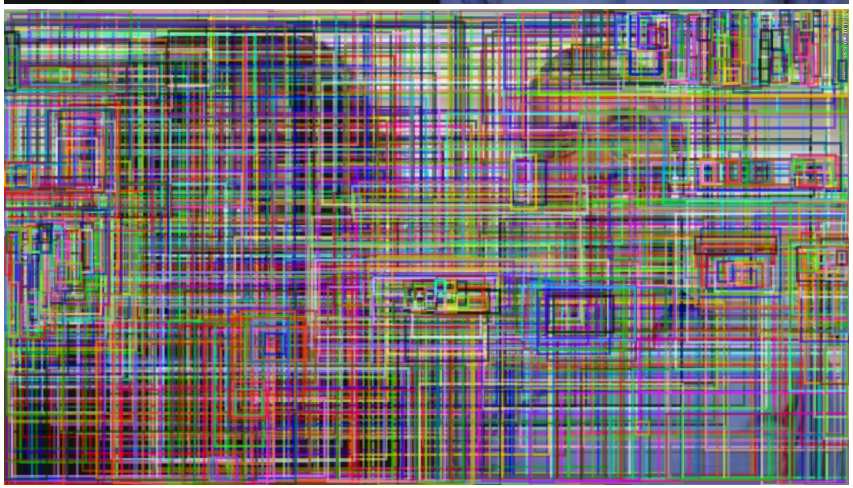
First Steps

An input image is passed to a CNN to generate a feature map.

- Usually a CNN architecture trained for image classification is used.
  - This promotes faster convergence during training.
- Let $m \times n$ be the size of an input image. If the CNN has an output stride of s, the spatial dimensions of the output feature map is $\frac{m}{s} \times \frac{n}{s}$.

RPN : Region Proposal Network

Original Image



RPN Output

RPN: Rationale

- Let us consider an image of size $m \times n$.
- Let us be provided with an information that all objects in the image have following constraints:
  - All objects' heights must be in the range $[H_{min}, H_{max}]$.
  - The aspect ratio r (i.e *height/width*) of every object must be in the range $[r_{min}, r_{max}]$.
- With these constraints, there is a very large number of possible regions in the image containing an object.

RPN: Numerical example of rationale

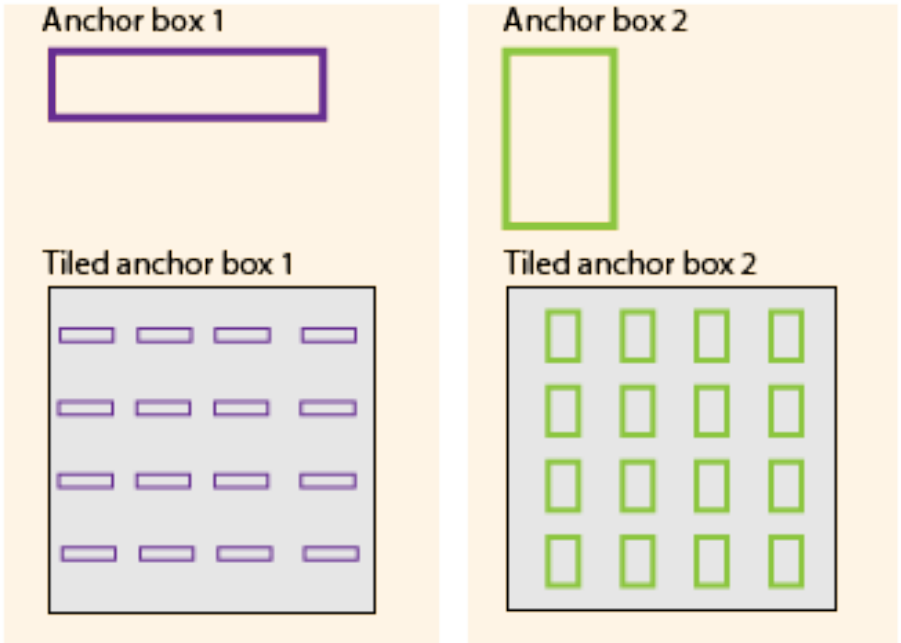
- Let an image be of size 512×512.
- For the height range let us consider an array $H = [20 : 10 : 200]$ i.e minimum height is 20, maximum height is 200 with a step size of 10.
- For aspect ratio let us consider an array $AR = [0.2 : 0.1 : 5]$.
- Therefore for each location in the image, we have to search over $len(H) \times len(AR) = 19 \times 49 = 931$ regions centered at it.
- Therefore total number of search regions is $512 \times 512 \times 931 = 244056064$.
- A RPN reduces the number of search regions.

RPN: Back to Rationale

- Given an input image, a RPN provides a set of R bounding boxes known as proposals.
- Each bounding box denotes a region which the neural network thinks has a possibility of containing an object.
- This reduces the number of possible regions to be searched for objects.

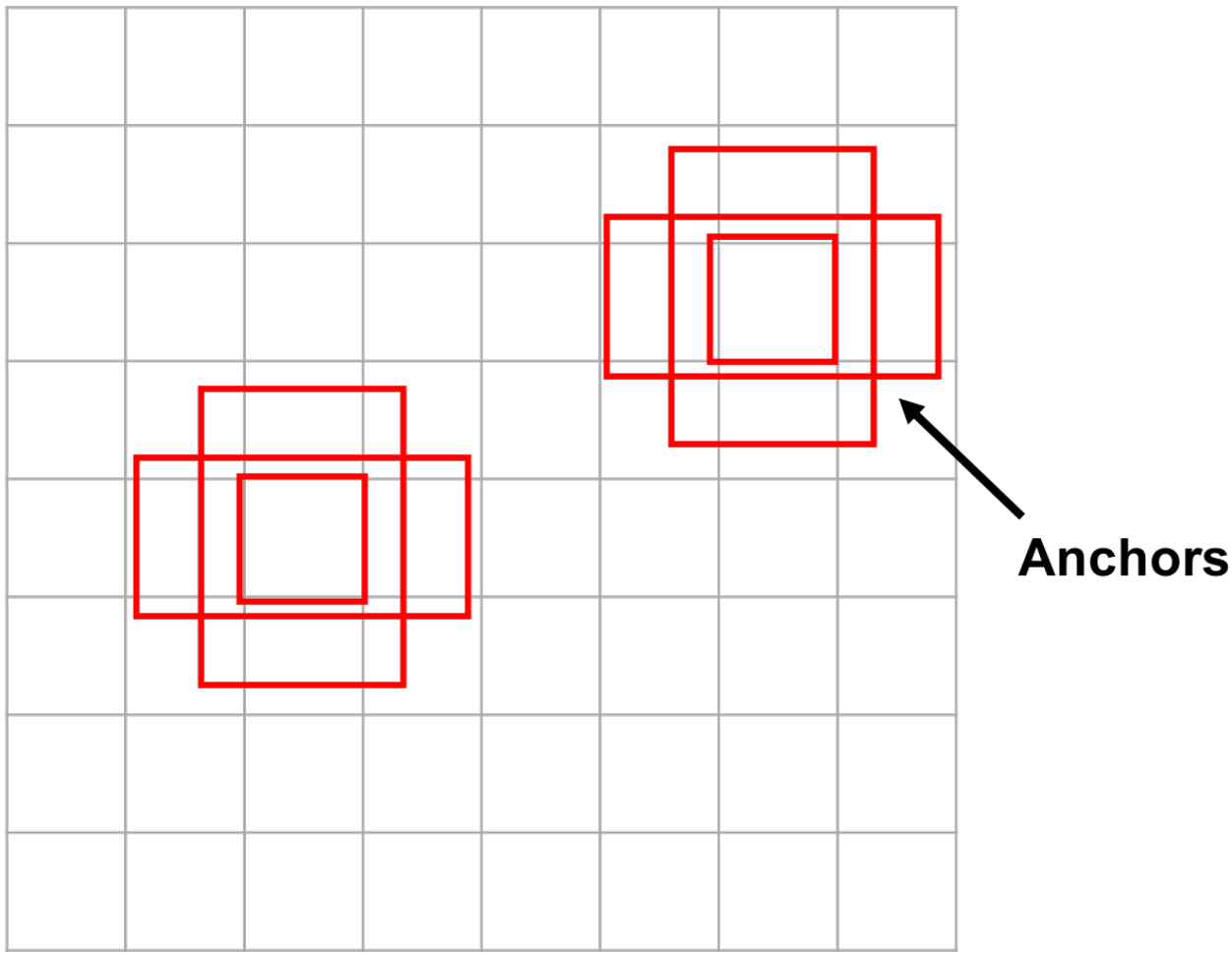
RPN: Anchors

- Anchors are hypothetical bounding boxes assumed to overlay a feature map.

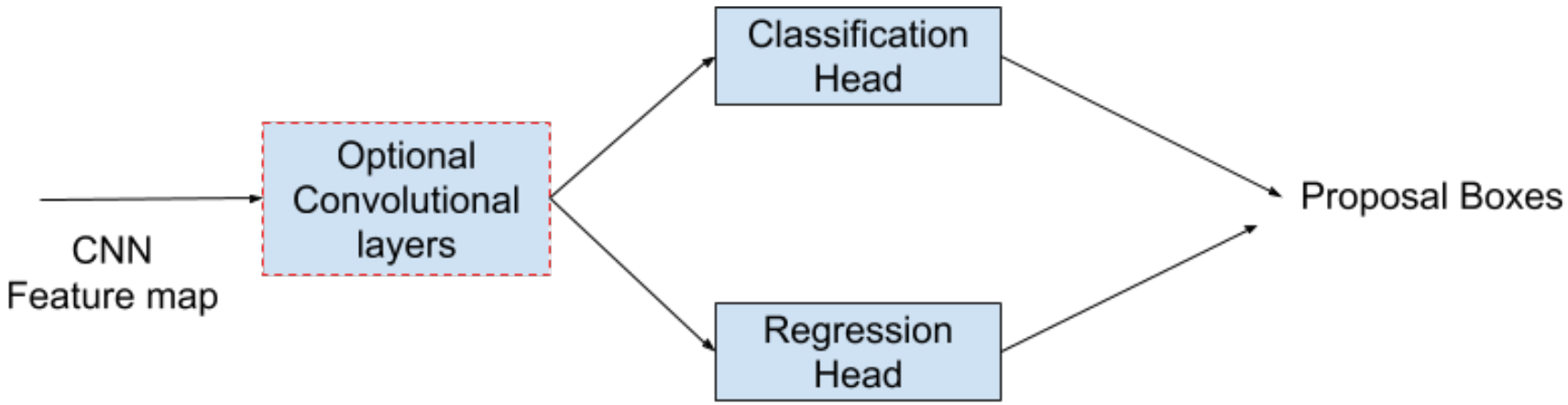


Anchors

Since à priori knowledge of object sizes is not available, a variety of anchors are assumed to overlay a feature map.



RPN Architecture



RPN: Classification Head

- Let us consider that there are R confocal anchors at each location.
- A classification head is a convolutional layer with $2R$ filters.
  - The size of filters is an experimental hyperparameter. Usually 3×3 or 1×1 is used.
- $2R$ filters denote that for each anchor, the prediction has to be made across 2 object classes
  1. Object.
  2. Not an object.

RPN: Regression Head

- A regression head is a convolutional layer with $4R$ filters.
  - The size of filters is an experimental hyperparameter. Usually 3×3 or 1×1 is used.
- $4R$ filters denote that for each anchor, the prediction should be for 4 quantities to predict the coordinates of the bounding box.

RPN during training

- During training, the groundtruth bounding boxes are known.
- the IoU is computed for each groundtruth boundinx box with each anchor.
- The anchors for which $IoU > \tau_u$ are meant to be positive anchors.
  - For positive anchors, they must be predicted as the "object" class and their regression target must be the coordinates of the groundtruth box.
- The anchors for which $IoU < \tau_l$ are meant to be negative anchors.
  - For negative anchors, they must be predicted as the "not an object" class and their regression targets are not relevant. This will be come clearer when we discuss the loss function for RPN.
- τ_u and τ_l are hyperparameters. In the original paper, $\tau_u = 0.7$ and $\tau_l = 0.3$.
- For IoU in the range $[0.3, 0.7]$, the outputs are not considered during training.

RPN Loss Function

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*).$$

- starred $(*)$ terms are groundtruth values. Non-starred are predicted values.
- p_i is the classification head output for i^{th} anchor.
- t_i is the regression head output for i^{th} anchor.
- N_{cls} and N_{reg} are scalar quantities used for normalizing the values. We discuss this in the next slide.
- λ is a regularization parameter to balance the importance of classification and regression. Usually both are equally important and so $\lambda = 1$.

RPN during training

- Total number of anchors is usually much larger than number of objects.
- As a result most anchors based on IoU values with GT boxes are negative.
- If the RPN loss function is minimized for all anchors, the negative anchors will dominate.
  - The network is learn about negative class i.e background, but will not be good at detecting objects.
- Hence, prior to loss function computation a set of anchors are chosen for which the loss function is minimized.

RPN during training

- We consider all anchors for which $IoU > \tau_u$.
  - This is because we want network to detect all objects. So, the network must learn about anchors which are positive.
- We sort the classification confidence score of all negative anchors and select those for which the classification score of being an object is highest.
  - This amounts to selecting the hardest possible examples.
- Usually the ratio of positive and negative anchors for loss function computation is considered as $1:3$.

RPN loss function again

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*).$$

- N_{cls} is total number of anchors considered for loss computation.
- N_{reg} is total number of feature map locations i.e for a feature map of $\frac{m}{s} \times \frac{n}{s}$, N_{reg} is $\frac{mn}{s^2}$
- p_i^* is 1 if the i^{th} anchor is positive else it is 0. So, regression is computed only for those anchors which are positive because for negative anchors we do not have a target i.e t_i^*.