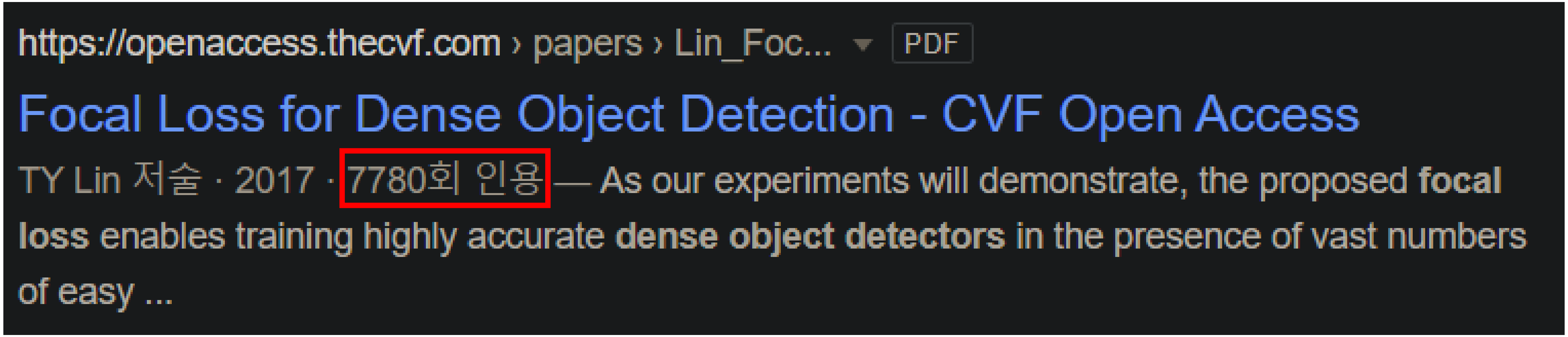
Focal Loss for Dense Object Detection

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Doll´ar

Facebook AI Research (FAIR)

Citation



https://openaccess.thecvf.com › papers › Lin_Foc... PDF

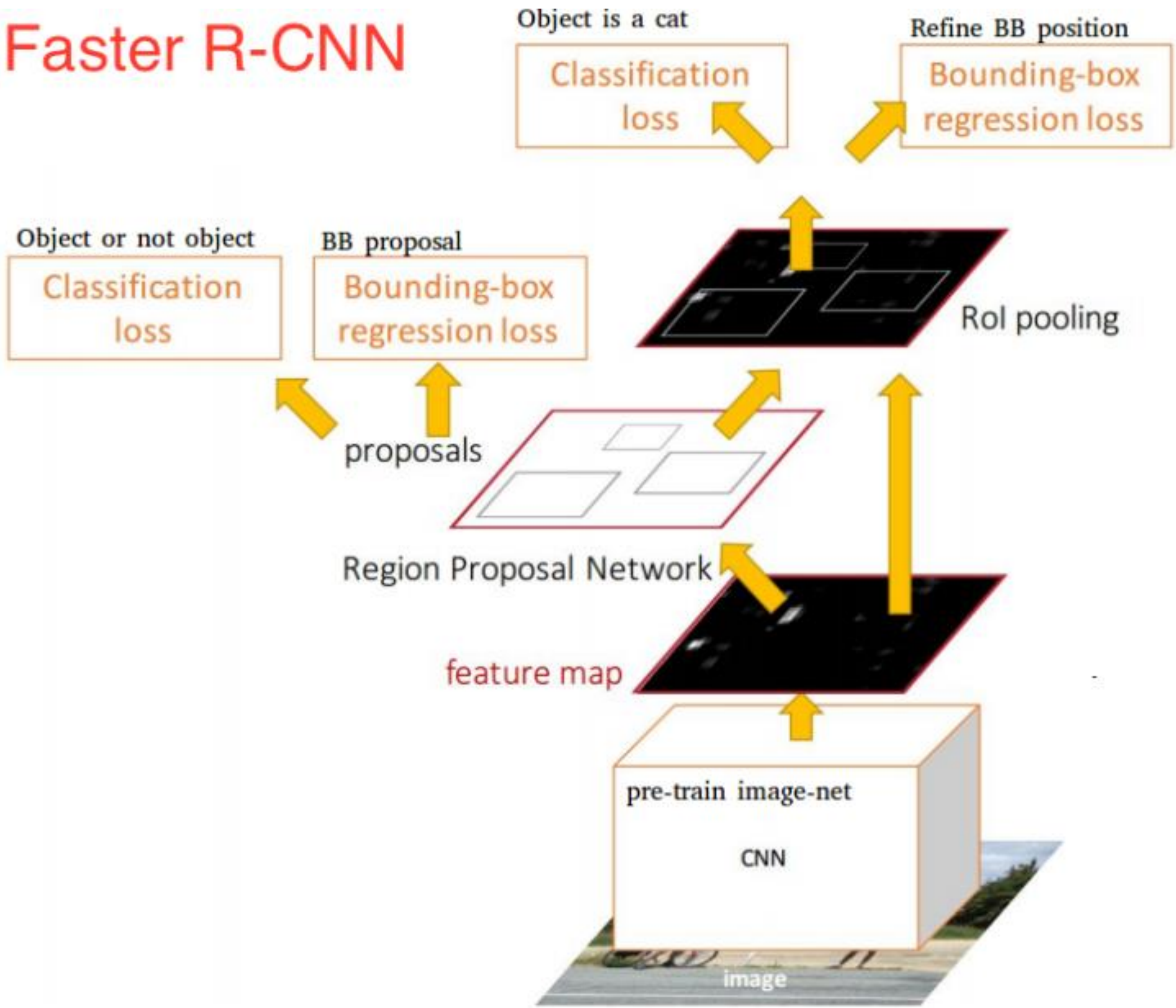
Focal Loss for Dense Object Detection - CVF Open Access

TY Lin 저술 · 2017 · 7780회 인용 — As our experiments will demonstrate, the proposed **focal loss** enables training highly accurate **dense object detectors** in the presence of vast numbers of easy ...

Motivation

- Despite the success of two-stage detectors, a natural question to ask is:

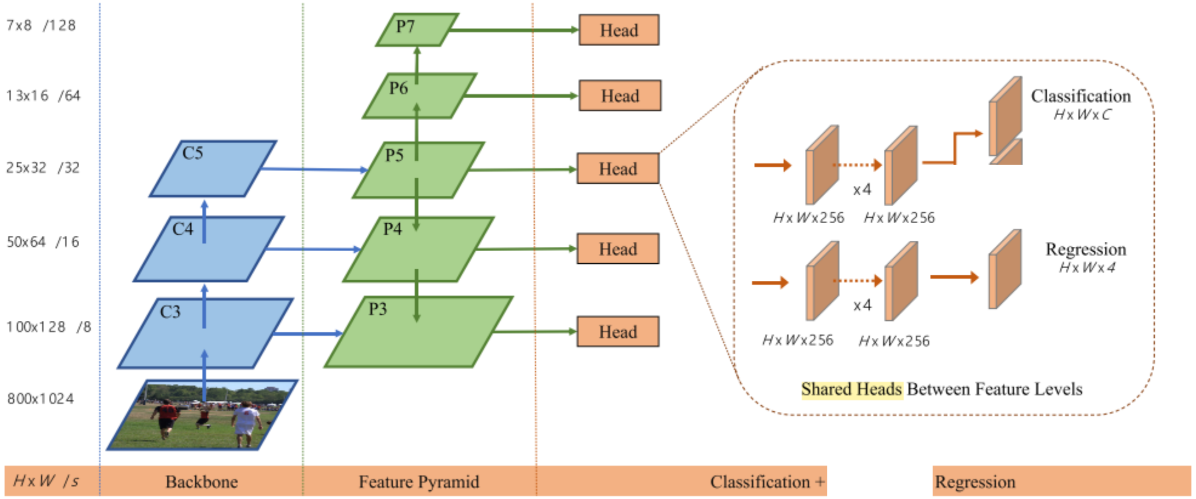
  Could a simple one-stage detector achieve similar accuracy?



Background: two-stage

- Better performance than One-stage models
- Complicate to train (backbone network, region proposal network, classifier)
- Slow, heavy

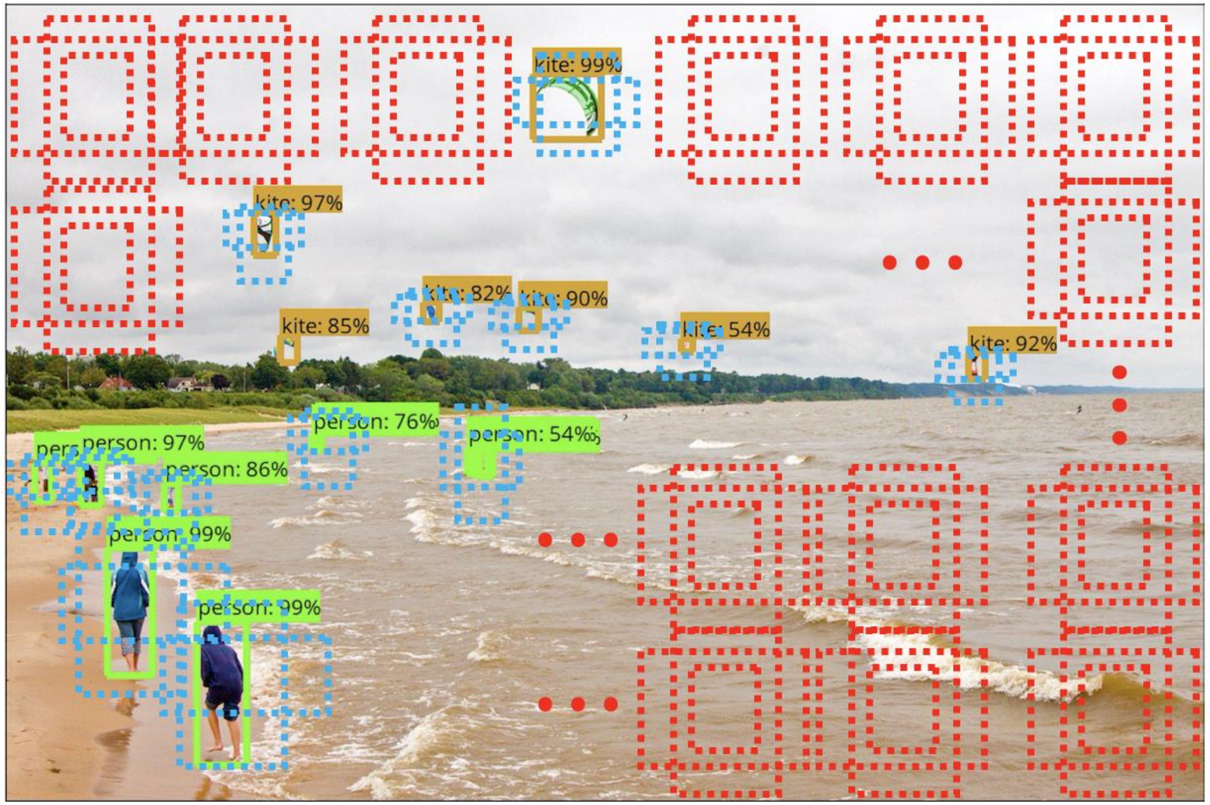
Background: one-stage

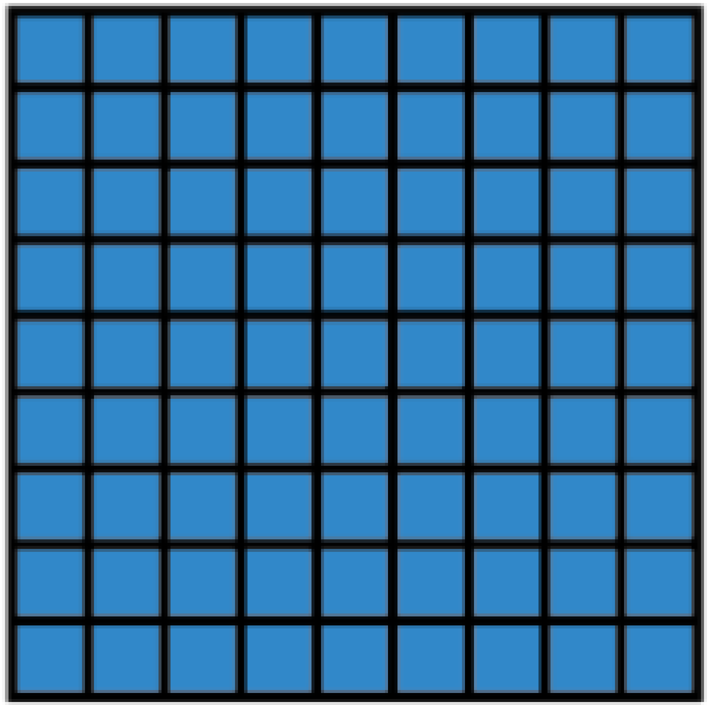


- Totally combined network
- End-to-End learning
- Lower performance than two-stage models
- Fast, lighter

Why two-stage detector get better performance? – Problem statement

- Extreme foreground-background **class imbalance** encountered **during training** of dense detectors is the **central cause**





Contribution

- One-stage detector improvement using **Focal loss**.

- Focal Loss focuses training on a sparse set of **hard examples** and prevents the vast number of **easy negatives** from overwhelming the detector during training.

Example

- For example, if there are 5 hard example which has 5 Loss value and 50 easy negative examples which have 1 Loss value, easy negative examples overwhelm hard examples!

Total Loss function

$$L = \lambda L_{loc} + \boxed{L_{cls}} \qquad (1)$$

- This paper focuses on improving classification loss

Classification loss (Focal Loss)

$$L_{cls} = -\sum_{i=1}^{K}(\boxed{y_i log(p_i)}\boxed{(1-p_i)^{\gamma}}\boxed{\alpha_i}+\boxed{(1-y_i)log(1-p_i)}\boxed{p_i^{\gamma}}\boxed{(1-\alpha_i)})$$

- Cross Entropy 🟥
  yi equals 1 if the ground-truth belongs to the i-th class and 0 otherwise; Pi is the predicted probability for the i-th class
- Balanced Cross Entropy 🟩
  αi ∈ [0,1]
- Focal Loss 🟦
  γ ∈ (0,+∞)

Focal Loss

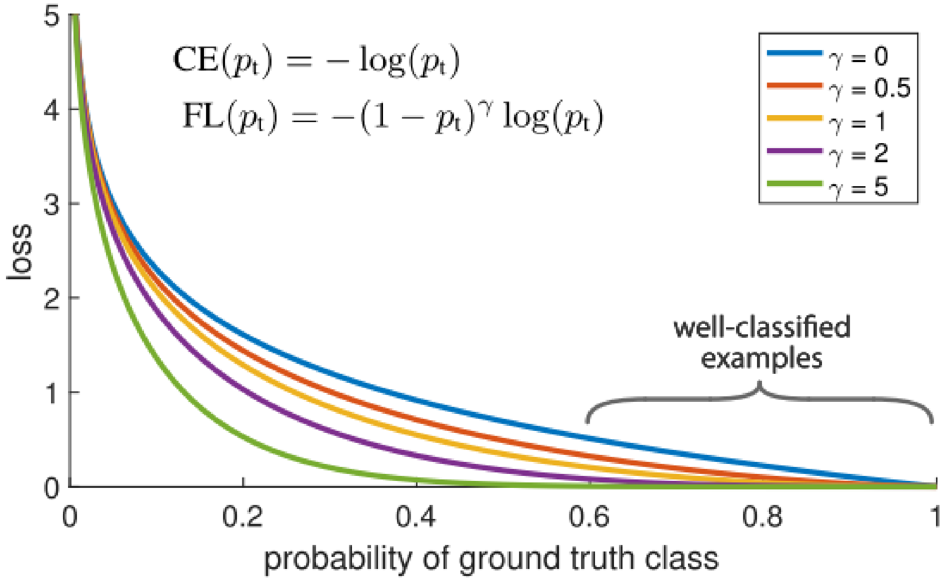
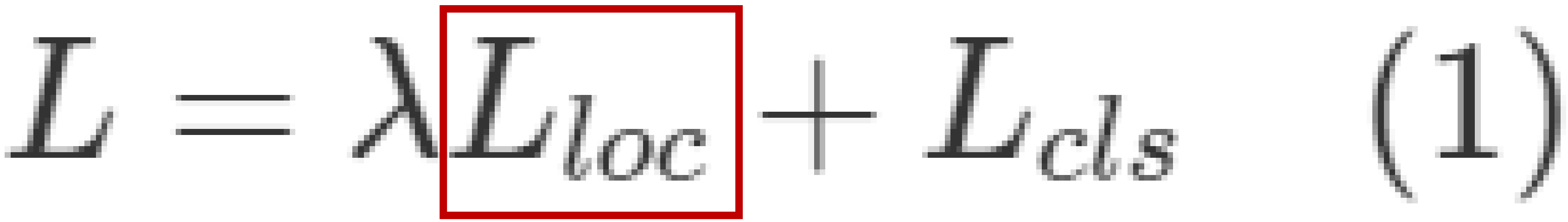


Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1 - p_t)^\gamma$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclassified examples. As our experiments will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

$$\mathrm{FL}(p_t) = -(1-p_t)^\gamma \log(p_t).$$

Higher Probability (easier),
Lower loss value

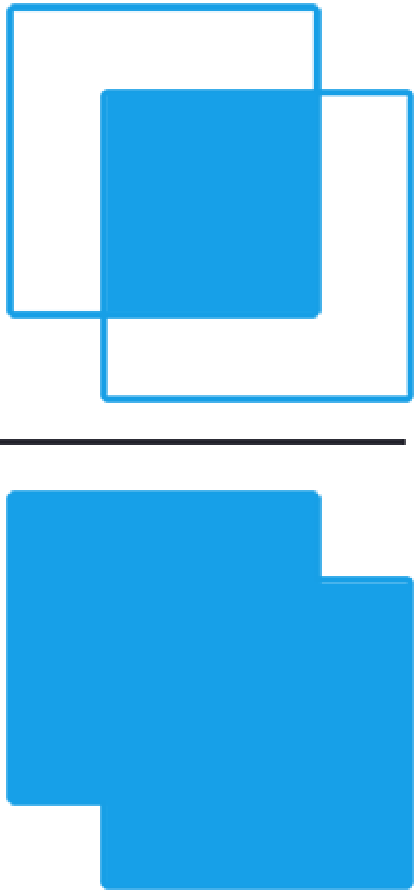
Total Loss function

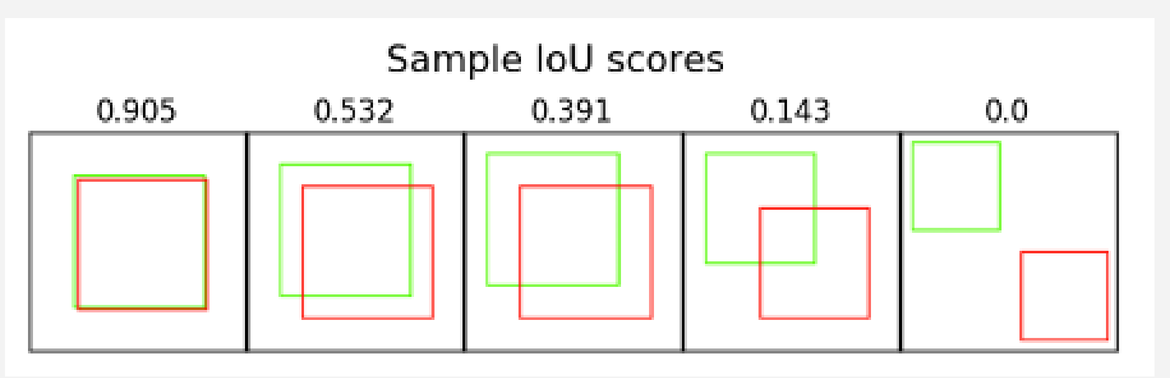


$$L = \lambda \boxed{L_{loc}} + L_{cls} \quad (1)$$

Background: Intersection over Union

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$





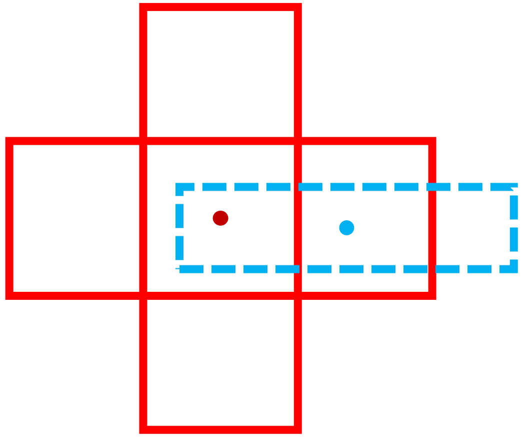
Bounding Box Regression loss

$$T_x^i = (G_x^i - A_x^i)/A_w^i \tag{2}$$

$$T_y^i = (G_y^i - A_y^i)/A_h^i \tag{3}$$

$$T_w^i = log(G_w^i/A_w^i) \tag{4}$$

$$T_h^i = log(G_h^i/A_h^i) \tag{5}$$

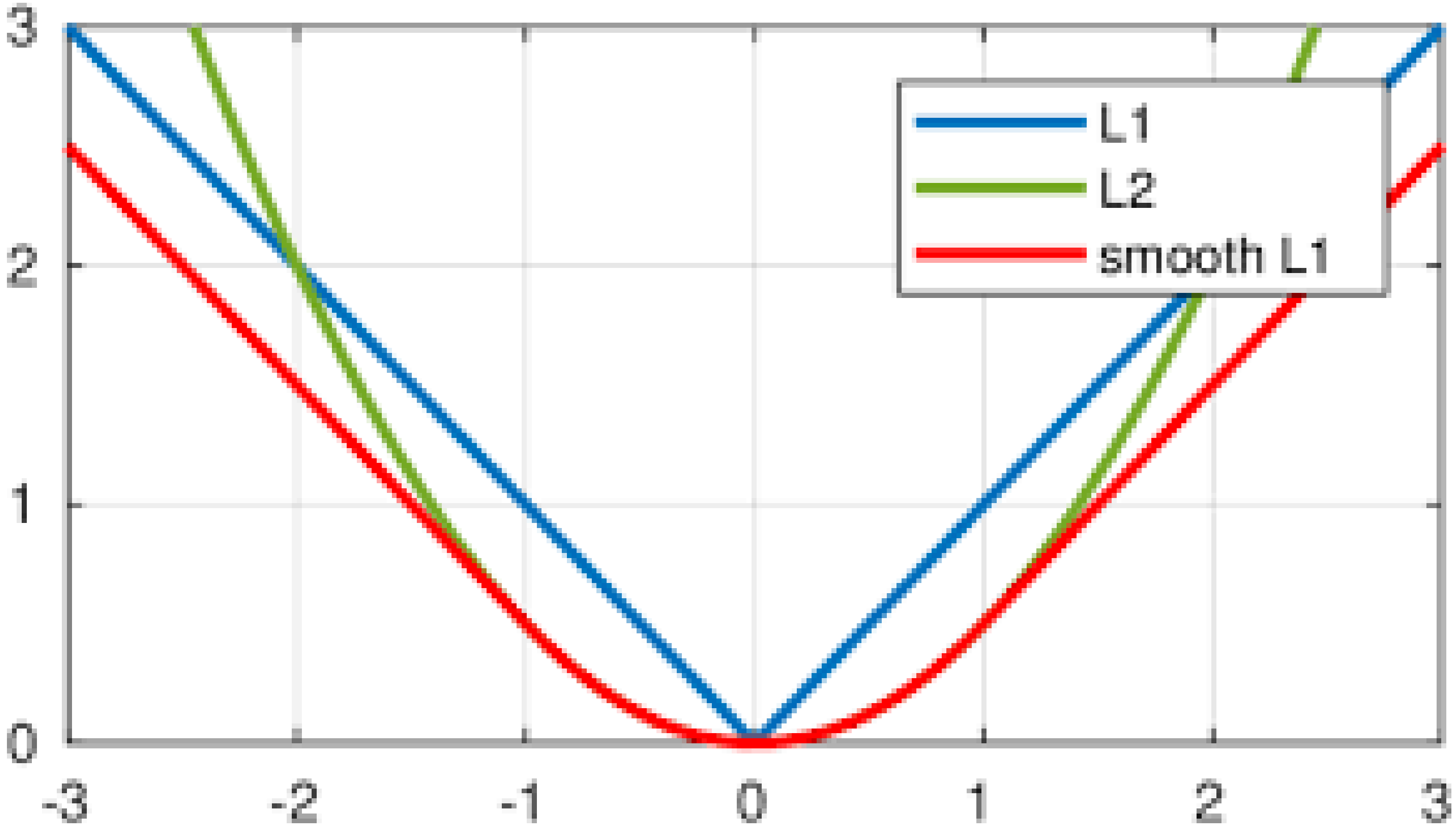


$$L_{loc} = \sum_{j\in\{x,y,w,h\}} smooth_{L1}(P_j^i - T_j^i) \tag{6}$$

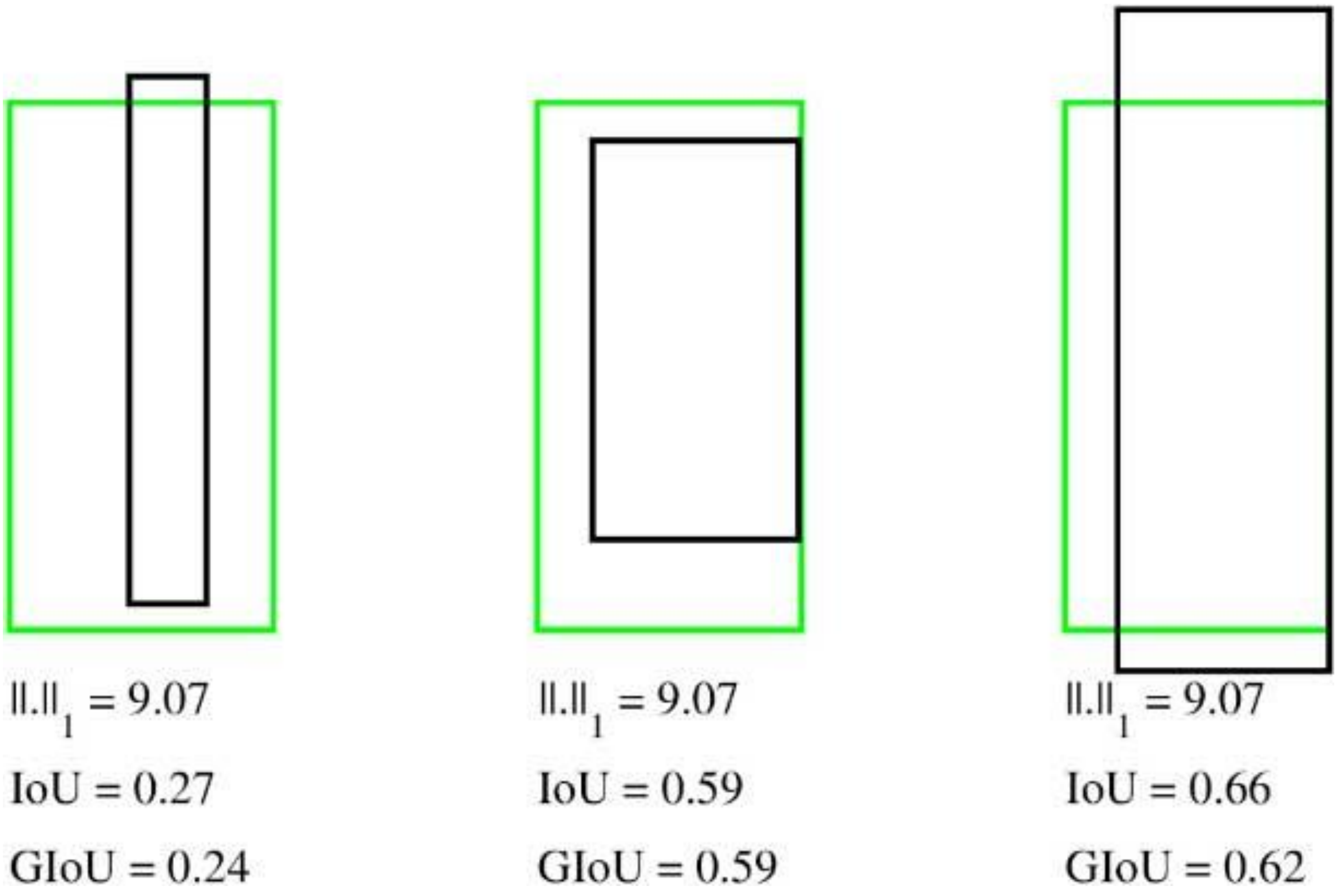
$$smooth_{L1}(x) = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| - 0.5 & |x| \geq 1 \end{cases} \tag{7}$$

- Regression loss is used only for bounding boxes which have GT class.

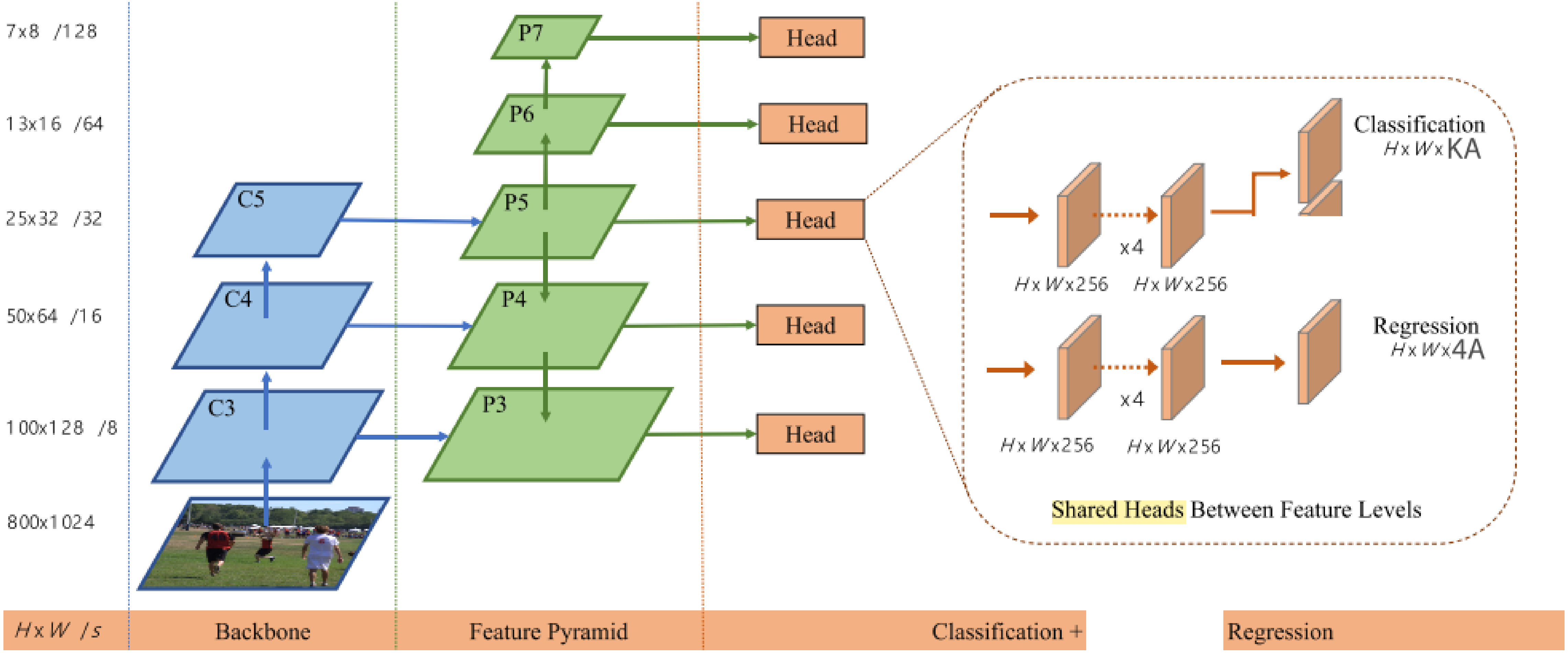
Background: smooth L1 Loss



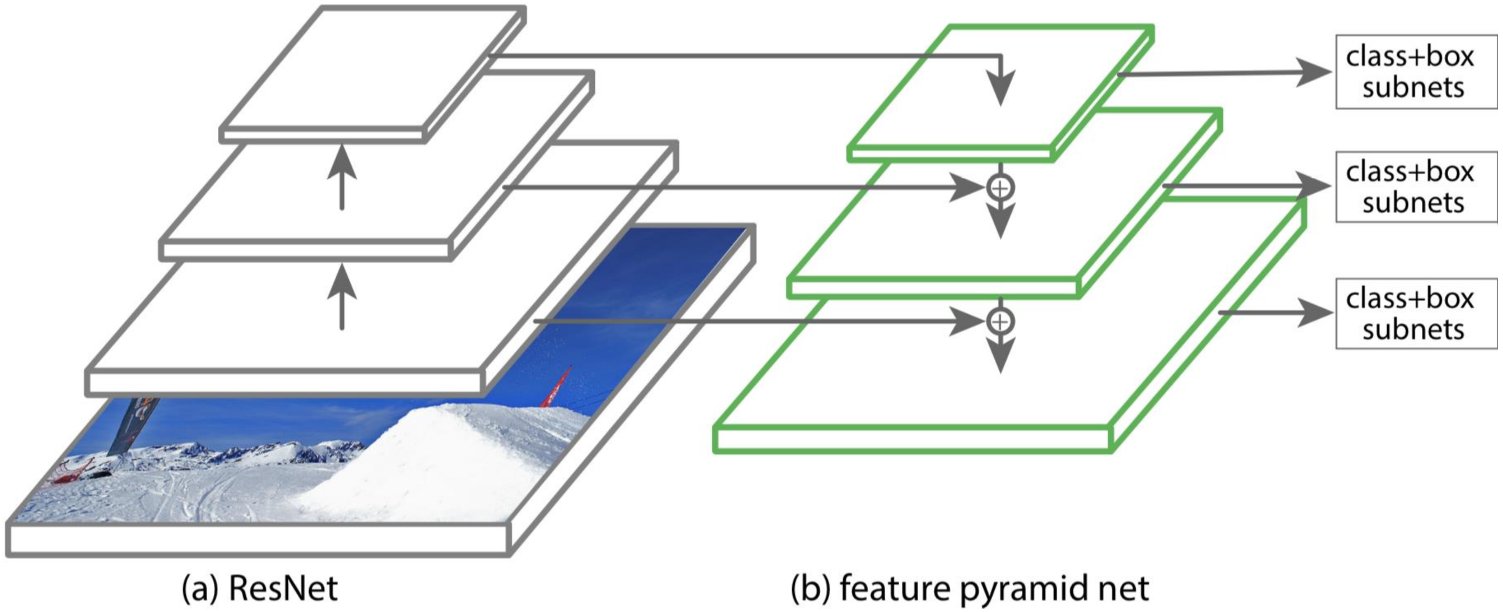
Background:
difference between L1 and IOU



RetinaNet Architecture (Proposed network)



Background: Feature Pyramid



Due to property of Receptive field,
CNN model doesn't get robust performance with **scale-variant objects**!

As alleviates this problem,
increase or decease receptive field size using interpolate feature map

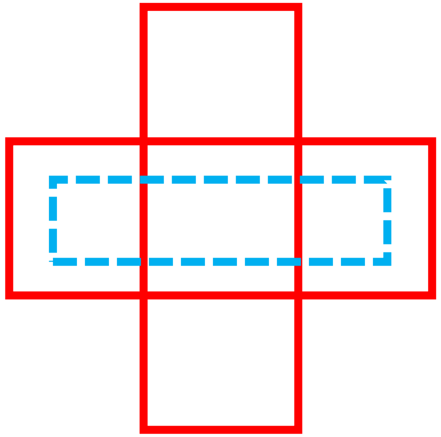
Training

- IOU between GT box and Anchor box

[0.5, 1.0] : assign to GT object boxes (L_{loc}, L_{cls})

[0.4, 0.5) : ignore

[0, 0.4) : background (L_{cls})

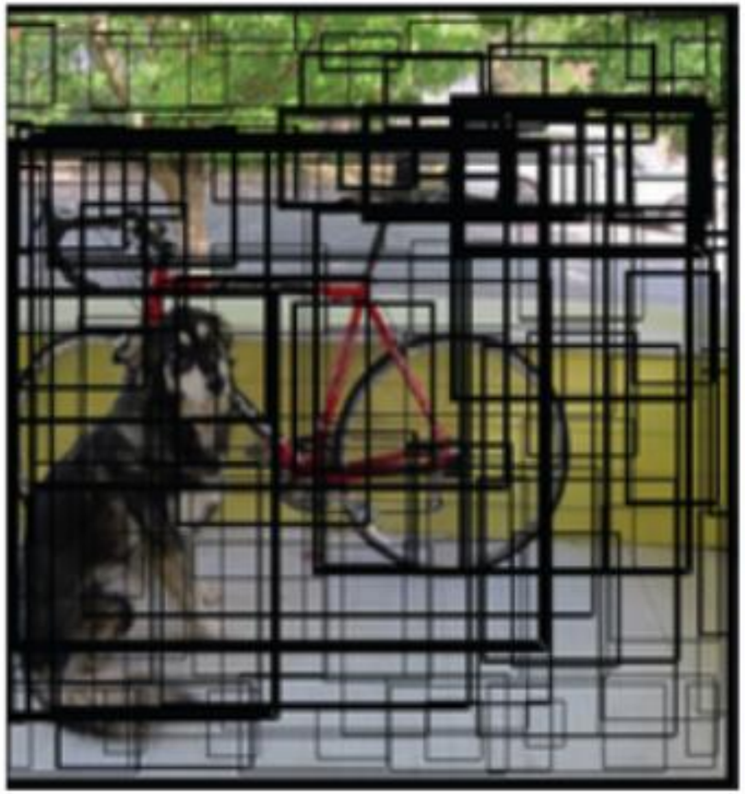


Inference

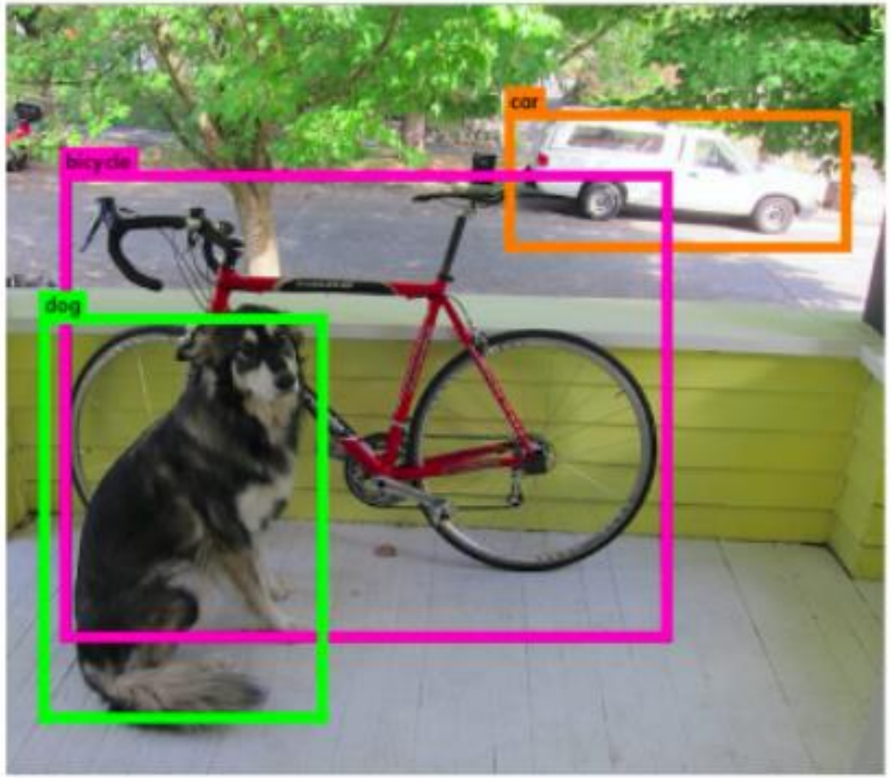
- We only decode box predictions from at **most 1k top-scoring predictions per FPN level**, after thresholding detector confidence at 0.05.

- The top predictions from all levels are merged and non-maximum suppression with a threshold of 0.5 is applied to yield the final detections.

Background: non-maximum suppression

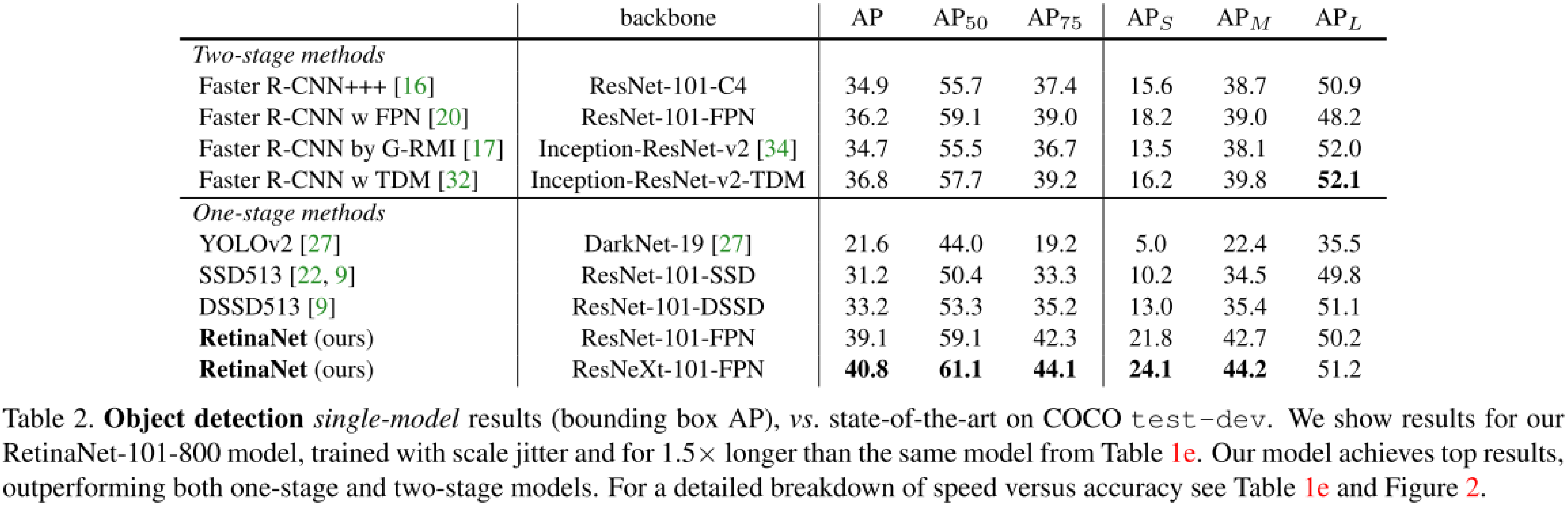


Multiple Bounding Boxes



Final Bounding Boxes

Results



	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [16]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [20]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [17]	Inception-ResNet-v2 [34]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [32]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
One-stage methods							
YOLOv2 [27]	DarkNet-19 [27]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [22, 9]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [9]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet (ours)	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet (ours)	ResNeXt-101-FPN	**40.8**	**61.1**	**44.1**	**24.1**	**44.2**	51.2

Table 2. **Object detection** *single-model* results (bounding box AP), *vs*. state-of-the-art on COCO `test-dev`. We show results for our RetinaNet-101-800 model, trained with scale jitter and for 1.5× longer than the same model from Table 1e. Our model achieves top results, outperforming both one-stage and two-stage models. For a detailed breakdown of speed versus accuracy see Table 1e and Figure 2.

End-to-End Object Detection with Transformers

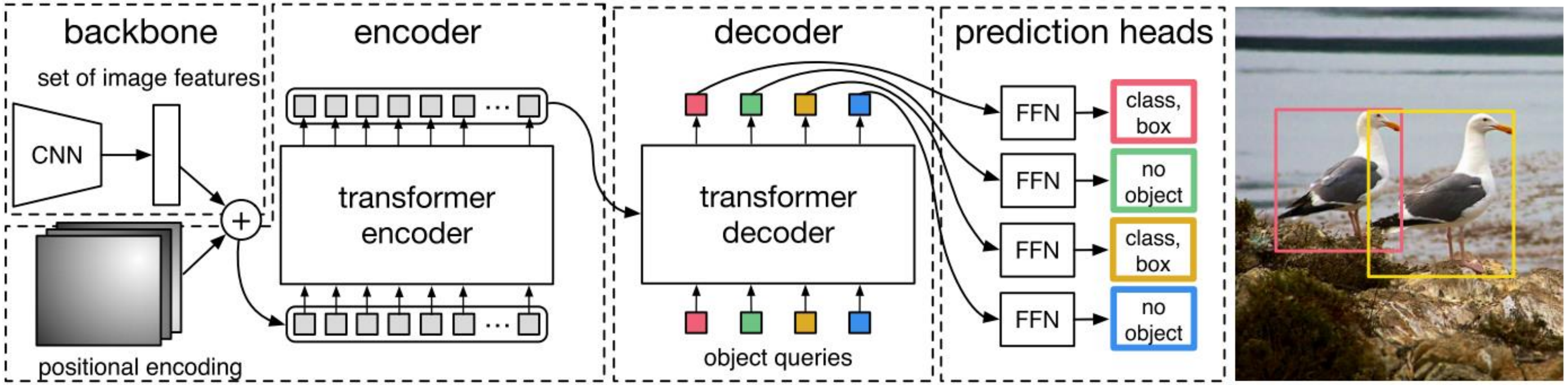


Fig. 2: DETR uses a conventional CNN backbone to learn a 2D representation of an input image. The model flattens it and supplements it with a positional encoding before passing it into a transformer encoder. A transformer decoder then takes as input a small fixed number of learned positional embeddings, which we call *object queries*, and additionally attends to the encoder output. We pass each output embedding of the decoder to a shared feed forward network (FFN) that predicts either a detection (class and bounding box) or a "`no object`" class.

Reference

- RetinaNet Explained and Demystified | NickZeng|曾广宇 (zenggyu.com)

- H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized intersection over union: A metric and a loss for bounding box regression,” 2019. doi: 10.1109/CVPR.2019.00075.

- S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” 2017. doi: 10.1109/TPAMI.2016.2577031.

- [1] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully convolutional one-stage object detection,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, vol. 2019-Octob, pp. 9626–9635, doi: 10.1109/ICCV.2019.00972.

- N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-End Object Detection with Transformers.”