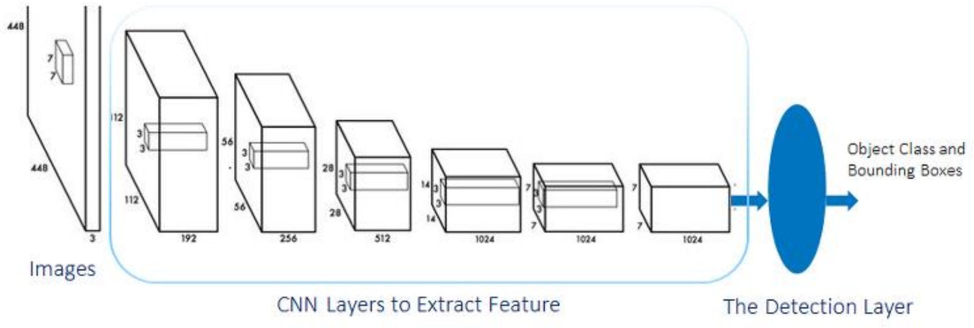
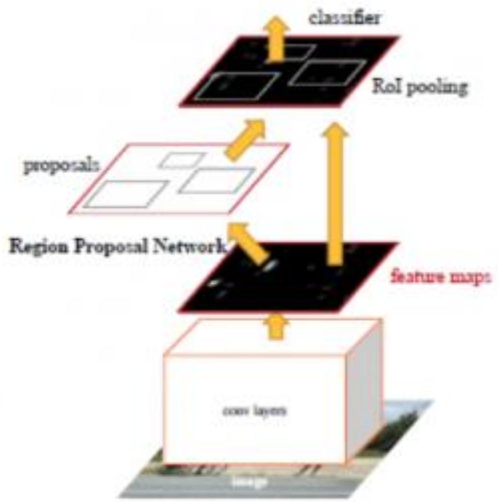
Focal Loss for Dense Object Detection

Tsung-Yi Lin Priya Goyal Ross Girshick Kaiming He Piotr Dollár
Facebook AI Research (FAIR)

1. Introduction





One Stage Object Detector(YOLO)

Regular, **dense sampling** of possible object location, scales, and aspect ratios.

Faster, Simpler but Less Accurate
(∵ Class Imbalance)

Two Stage Object Detector(Faster R-CNN)

1-stage: generate **sparse set(1~2k)** of candidate object location
2-stage: classifies each candidate location as one of foreground classes or background + **Biased Mini-Batch Sampling**

2. Class Imbalance

Occurs in one-stage detector training
$10^4 \sim 10^5$ candidate location per image
but only a few locations contain object

Why problem?

1) Training is inefficient as most locations are easy negatives that gives no useful learning signal
2) En masse, the easy negatives can overwhelm training and degenerate models

Hard negative mining
: samples only hard examples during training

Two Stage Detector:
Two-Stage Cascade + Biased mini-batch sampling
Construct mini-batch that contain ratio of pos and neg(e.g., 1:3)

➔ Focal Loss solves class imbalance problem in a one-stage detector

3. Focal Loss

Dynamically scaled Cross Entropy Loss

1) Cross Entropy Loss

$$\mathrm{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise.} \end{cases}$$

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise,} \end{cases}$$

$$\mathrm{CE}(p, y) = \mathrm{CE}(p_t) = -\log(p_t)$$

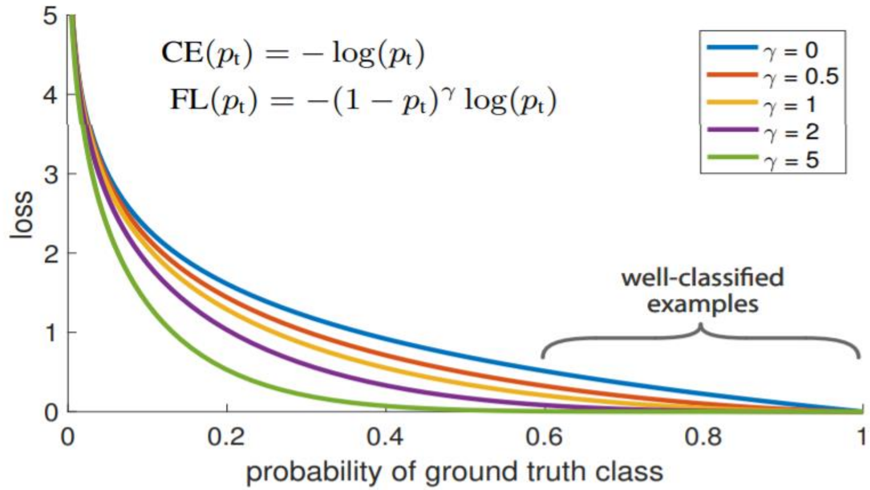


Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1 - p_t)^\gamma$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclassified examples. As our experiments will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

Easily Classified samples incur a loss with non-trivial magnitude. ☹

Especially **Easy negative samples**

3. Focal Loss

Dynamically scaled Cross Entropy Loss

2) Balanced Cross Entropy Loss

$$\mathrm{CE}(p_t) = -\boxed{\alpha_t}\log(p_t)$$

$\alpha \in [0, 1]$

$$\alpha_t = \begin{cases} \alpha, & y = 1 \\ 1 - \alpha, & otherwise \end{cases}$$

α = inverse of class frequency
or hyperparameter set by cross-validation

➔ As p_t gets bigger, $\mathrm{CE}(p_t)$ reduces smaller loss

Class imbalance between positive and negative examples is solved. ☺

Need to differentiate between easy sample and hard sample.

3. Focal Loss

Dynamically scaled Cross Entropy Loss

3) Focal Loss

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

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

Positive/Negative
(Rare/Frequent)

Easy/Hard

To down-weight easy sample and to focus on hard sample, add modulating factor

$$(1-p_t)^{\gamma}, \qquad \gamma > 0$$

1) When an example is misclassified and p_t is small, the modulating factor is near 1 and loss is unaffected

2) The focusing parameter γ smoothly adjust the rate at which easy examples are down-weighted.

➔ Modulating factor reduces the loss contribution from easy examples and extends the range in which an example receives low loss.

3. Focal Loss

Dynamically scaled Cross Entropy Loss

3) Focal Loss

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

Positive/Negative
(Rare/Frequent)

Easy/Hard

When $\gamma = 2$,

$p_t = 0.9 \rightarrow$ 100x lower loss than CE Loss
$p_t = 0.968 \rightarrow$ 1000x lower loss
$p_t \leq 0.5 \rightarrow$ at most 4x lower loss

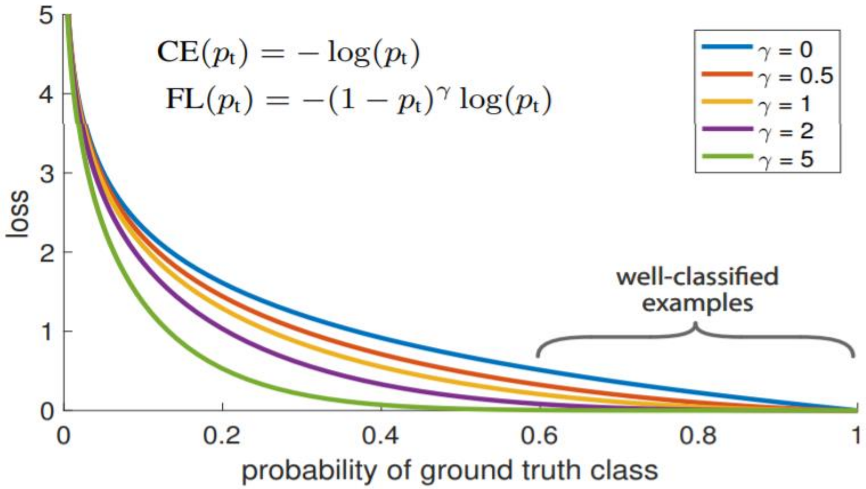


Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1 - p_t)^{\gamma}$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclassified examples. As our experiments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

4. RetinaNet Architecture

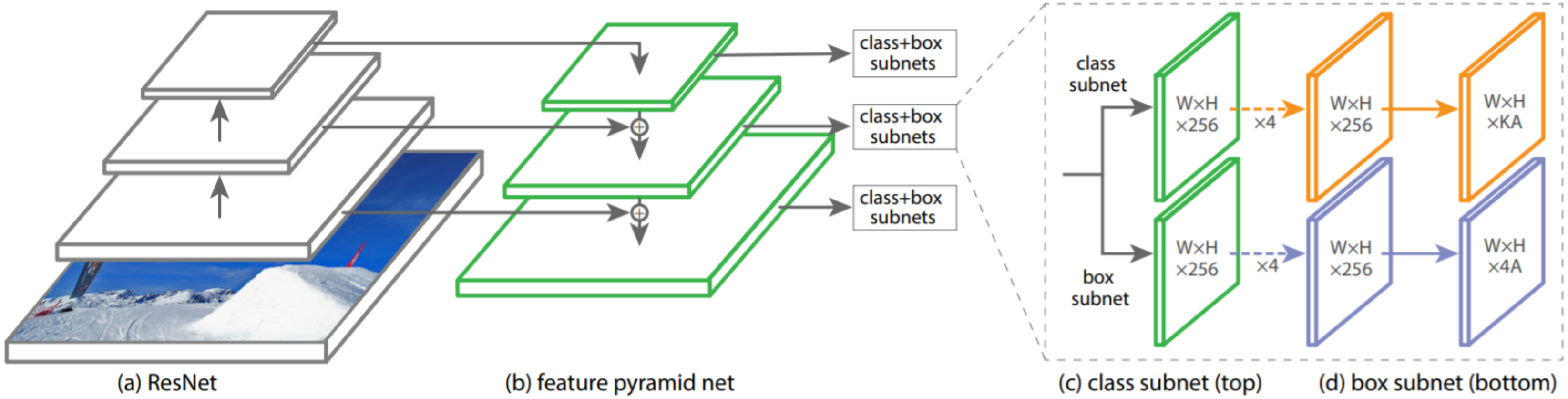


Figure 3. The one-stage **RetinaNet** network architecture uses a Feature Pyramid Network (FPN) [20] backbone on top of a feedforward ResNet architecture [16] (a) to generate a rich, multi-scale convolutional feature pyramid (b). To this backbone RetinaNet attaches two subnetworks, one for classifying anchor boxes (c) and one for regressing from anchor boxes to ground-truth object boxes (d). The network design is intentionally simple, which enables this work to focus on a novel focal loss function that eliminates the accuracy gap between our one-stage detector and state-of-the-art two-stage detectors like Faster R-CNN with FPN [20] while running at faster speeds.

Composed of

1. Backbone network = ResNet-101-FPN
2. Classification subnetwork
3. Box regression subnetwork

Features

i) In-network feature pyramid
ii) Use of anchor boxes

4. RetinaNet Architecture

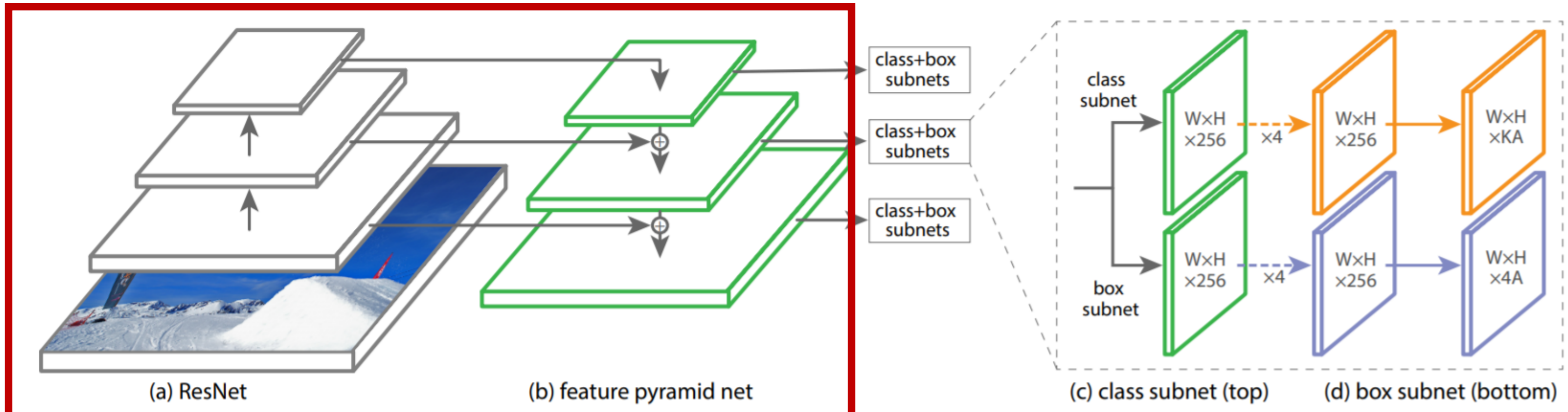


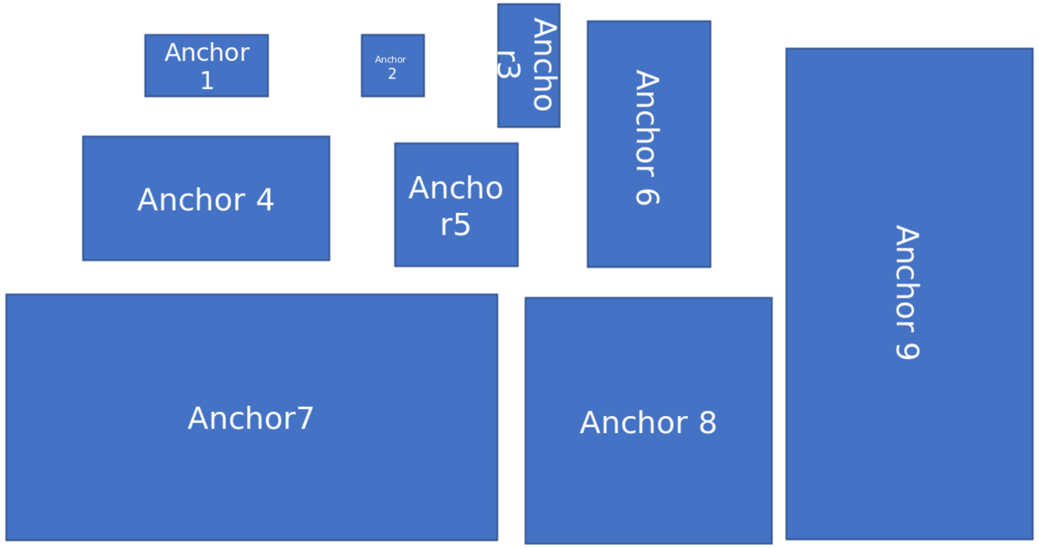
Figure 3. The one-stage **RetinaNet** network architecture uses a Feature Pyramid Network (FPN) [20] backbone on top of a feedforward ResNet architecture [16] (a) to generate a rich, multi-scale convolutional feature pyramid (b). To this backbone RetinaNet attaches two subnetworks, one for classifying anchor boxes (c) and one for regressing from anchor boxes to ground-truth object boxes (d). The network design is intentionally simple, which enables this work to focus on a novel focal loss function that eliminates the accuracy gap between our one-stage detector and state-of-the-art two-stage detectors like Faster R-CNN with FPN [20] while running at faster speeds.

Standard Convolutional Network(ResNet) + Top-down Pathway + Lateral Connection

➔ Multi-scale feature pyramid from a single resolution image
➔ **Multi-scale prediction**

# of Channel of all feature pyramid level is same. e.g., 256

4. RetinaNet Architecture



Focal Loss

1. Used as the loss on the output of the classification subnet
2. Applied to all ~ 100k anchors in each sampled image
3. Total Focal Loss =

$$\frac{\text{sum of the focal loss over all} \sim 100\text{k anchors}}{\text{the number of anchors assigned to a ground−truth box}}$$

Anchor boxes (for multi-scale object detection)

Translation-invariant for classification

Aspect ratio = {1 : 2, 1 : 1, 2 : 1}

Scale = $\{2^0,\ 2^{\frac{1}{3}},\ 2^{\frac{2}{3}}\}$

➔ 9 anchor boxes per each feature map

Each anchor has

1) **k-vector one-hot binary vector** of classification (k = # of class)

2) **4-vector** of box regression

And follows **RPN rule**

4. RetinaNet Architecture

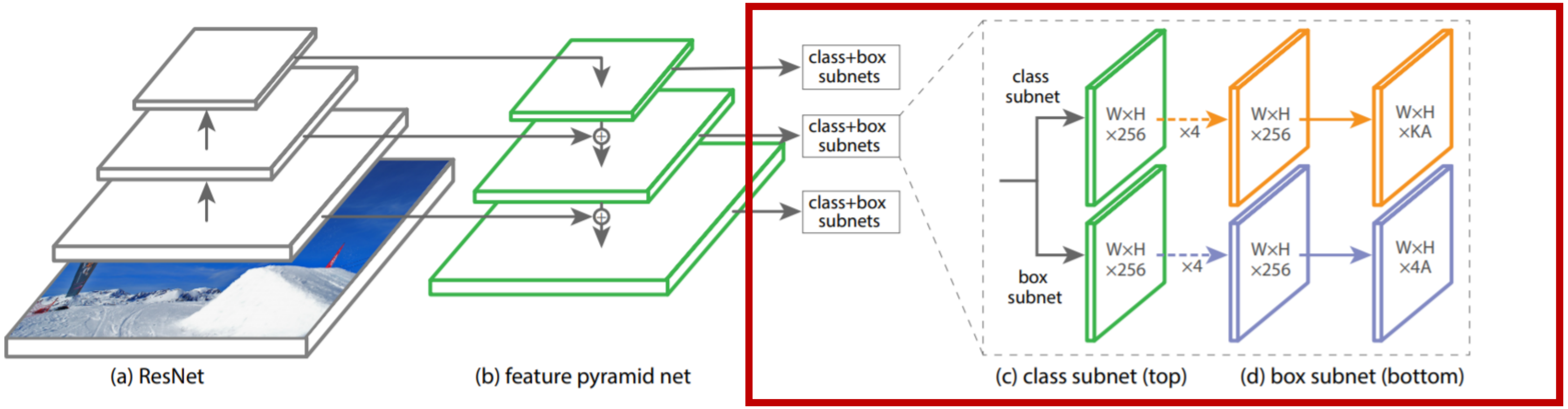


Figure 3. The one-stage **RetinaNet** network architecture uses a Feature Pyramid Network (FPN) [20] backbone on top of a feedforward ResNet architecture [16] (a) to generate a rich, multi-scale convolutional feature pyramid (b). To this backbone RetinaNet attaches two subnetworks, one for classifying anchor boxes (c) and one for regressing from anchor boxes to ground-truth object boxes (d). The network design is intentionally simple, which enables this work to focus on a novel focal loss function that eliminates the accuracy gap between our one-stage detector and state-of-the-art two-stage detectors like Faster R-CNN with FPN [20] while running at faster speeds.

Classification subnet

Predicts the probability of object presence at each spatial position for each of the A anchors and K object classes.

$$sigmoid(ReLU(3\ x\ 3\ conv, 256) * 4 + 3\ x\ 3\ conv, KA)$$
$$= KA\ binary\ prediction\ per\ spatial\ location$$

Box regression subnet

Output = 4A linear (4 = relative offset of bounding box)

5. Inference and Training

Inference

1. Threshold detector confidence at 0.05
2. Decoding box predictions from at most 1k top-scoring predictions per FPN level
3. Merge top predictions from all levels
4. Apply Non-Maximum Suppression with a threshold of 0.5

Every anchor should be labeled as foreground with confidence of π

Initialization

Except final conv layer of the classification subnet, bias = 0, sigma = 0.01

Final conv layer of the classification subnet, bias is initialized as b = $-\log\left(\frac{1-\pi}{\pi}\right)$, $\pi = 0.01$

➔ Prevents the large number of background anchors from generating a large, destabilizing loss value in the first iteration of training

Optimization

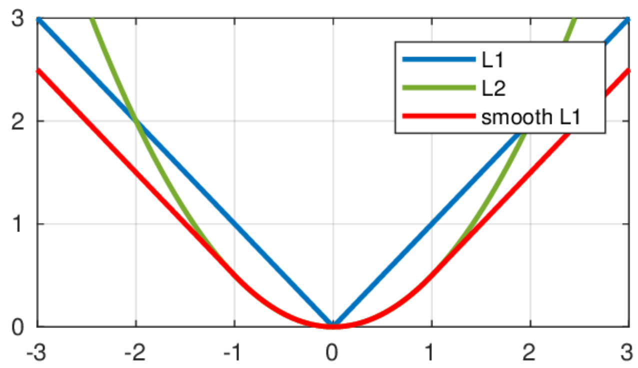
<u>Training loss = Focal Loss + Standard Smooth L_1 loss used for box regression</u>

Initial learning rate : 0.01, divided by 10 at 60k and again at 80k iterations

Data augmentation : horizontal image flipping

Weight decay : 0.0001

Momentum : 0.9



6. Experiments (COCO datset)

α	AP	AP_{50}	AP_{75}
.10	0.0	0.0	0.0
.25	10.8	16.0	11.7
.50	30.2	46.7	32.8
.75	31.1	49.4	33.0
.90	30.8	49.7	32.3
.99	28.7	47.4	29.9
.999	25.1	41.7	26.1

(a) **Varying** α **for CE loss** ($\gamma = 0$)

γ	α	AP	AP_{50}	AP_{75}
0	.75	31.1	49.4	33.0
0.1	.75	31.4	49.9	33.1
0.2	.75	31.9	50.7	33.4
0.5	.50	32.9	51.7	35.2
1.0	.25	33.7	52.0	36.2
2.0	.25	**34.0**	**52.5**	**36.5**
5.0	.25	32.2	49.6	34.8

(b) **Varying** γ **for FL** (w. optimal α)

#sc	#ar	AP	AP_{50}	AP_{75}
1	1	30.3	49.0	31.8
2	1	31.9	50.0	34.0
3	1	31.8	49.4	33.7
1	3	32.4	52.3	33.9
2	3	**34.2**	**53.1**	**36.5**
3	3	34.0	52.5	**36.5**
4	3	33.8	52.1	36.2

(c) **Varying anchor scales and aspects**

6. Experiments (COCO datset)

method	batch size	nms thr	AP	AP_{50}	AP_{75}
OHEM	128	.7	31.1	47.2	33.2
OHEM	256	.7	31.8	48.8	33.9
OHEM	512	.7	30.6	47.0	32.6
OHEM	128	.5	32.8	50.3	35.1
OHEM	256	.5	31.0	47.4	33.0
OHEM	512	.5	27.6	42.0	29.2
OHEM 1:3	128	.5	31.1	47.2	33.2
OHEM 1:3	256	.5	28.3	42.4	30.3
OHEM 1:3	512	.5	24.0	35.5	25.8
FL	n/a	n/a	**36.0**	**54.9**	**38.7**

(d) **FL** ***vs*****. OHEM** baselines (with ResNet-101-FPN)

Online Hard Exampling Mining(OHEM)
Puts more emphasis on misclassified examples
Discard easy examples
Used in Two Stage Object Detector

6. Experiments (COCO datset)

depth	scale	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	time
50	400	30.5	47.8	32.7	11.2	33.8	46.1	64
50	500	32.5	50.9	34.8	13.9	35.8	46.7	72
50	600	34.3	53.2	36.9	16.2	37.4	47.4	98
50	700	35.1	54.2	37.7	18.0	39.3	46.4	121
50	800	35.7	55.0	38.5	18.9	38.9	46.3	153
101	400	31.9	49.5	34.1	11.6	35.8	48.5	81
101	500	34.4	53.1	36.8	14.7	38.5	49.1	90
101	600	36.0	55.2	38.7	17.4	39.6	49.7	122
101	700	37.1	56.6	39.8	19.1	40.6	49.4	154
101	800	37.8	57.5	40.8	20.2	41.1	49.2	198

(e) **Accuracy/speed trade-off** RetinaNet (on `test-dev`)

6. Experiments (COCO datset)

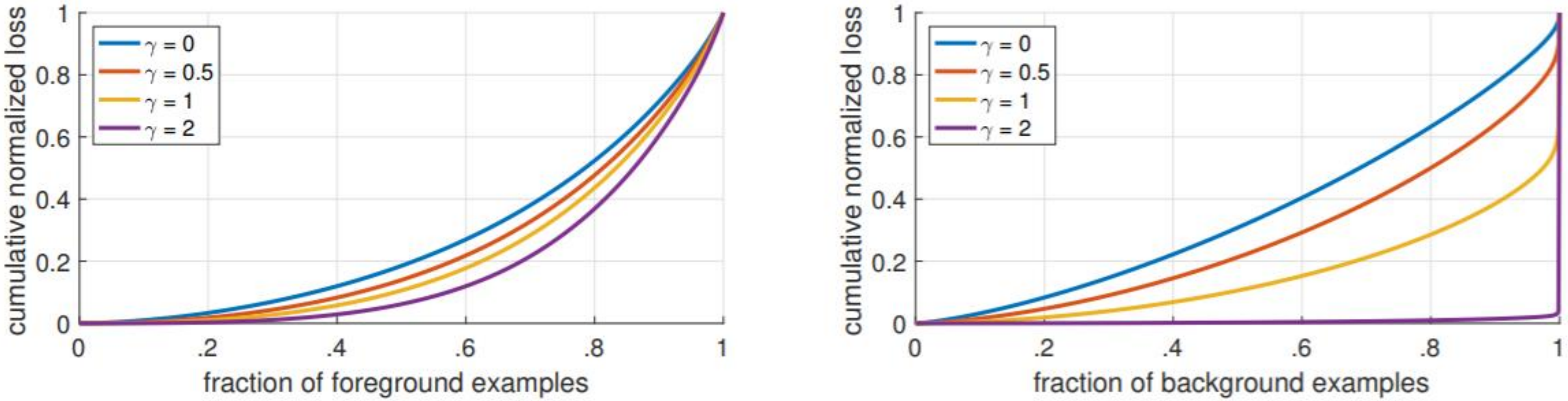


Figure 4. Cumulative distribution functions of the normalized loss for positive and negative samples for different values of γ for a *converged* model. The effect of changing γ on the distribution of the loss for positive examples is minor. For negatives, however, increasing γ heavily concentrates the loss on hard examples, focusing nearly all attention away from easy negatives.

6. Experiments (COCO datset)

	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.

6. Experiments (COCO datset)

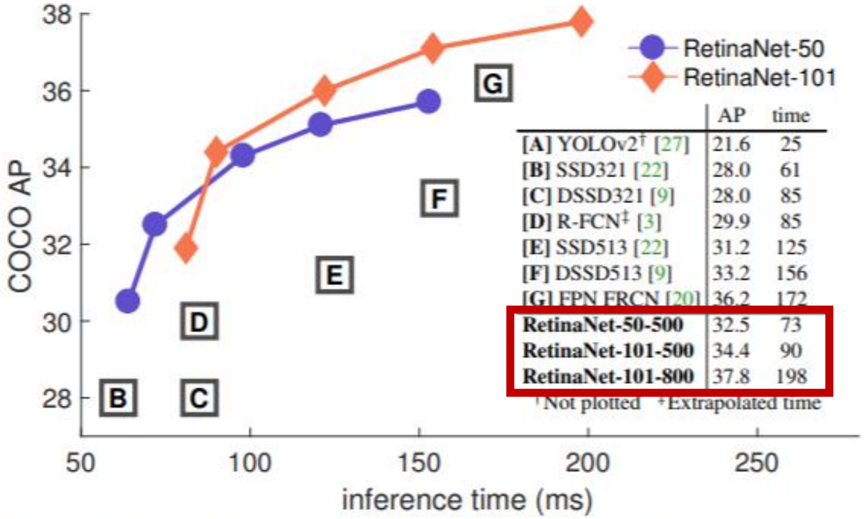


Figure 2. Speed (ms) versus accuracy (AP) on COCO `test-dev`. Enabled by the focal loss, our simple one-stage *RetinaNet* detector outperforms all previous one-stage and two-stage detectors, including the best reported Faster R-CNN [28] system from [20]. We show variants of RetinaNet with ResNet-50-FPN (blue circles) and ResNet-101-FPN (orange diamonds) at five scales (400-800 pixels). Ignoring the low-accuracy regime (AP<25), RetinaNet forms an upper envelope of all current detectors, and an improved variant (not shown) achieves 40.8 AP. Details are given in §5.

Thank You ☺