Is Sampling Heuristics Necessary in Training Deep Object Detectors?

Joya Chen[1], Dong Liu[1], Tong Xu[1], Shilong Zhang[1], Shiwei Wu[1], Bin Luo[2], Xuezheng Peng[2], Enhong Chen[1]

[1]University of Science and Technology of China [2]Tencent TaiQ Team
{chenjoya,zsl1996,dwustc}@mail.ustc.edu.cn,{dongeliu,tongxu}@ustc.edu.cn,{luobinluo,reuspeng}@tencent.com

Abstract

To address the imbalance between foreground and background, various heuristic methods, such as OHEM, Focal Loss, GHM, have been proposed for biased sampling or weighting when training deep object detectors. We challenge this paradigm by discarding the sampling heuristics and focusing on other settings for training. Our empirical study reveals that the weight of classification loss and the initialization strategy have a big impact on the training stability and the final accuracy. Thus, we propose the Sampling-Free *mechanism, including three key ingredients: optimal bias initialization, guided loss weights, and class-adaptive threshold, for training deep detectors without sampling heuristics. Compared with the sampling heuristics, our Sampling-Free mechanism is fully data diagnostic and thus avoids the laborious tuning of sampling hyper-parameters. Our extensive experimental results demonstrate that the Sampling-Free mechanism can be used for one-stage, two-stage, and anchor-free object detectors, where it always achieves higher accuracy on the challenging COCO benchmark. The mechanism is also useful for the instance segmentation task. Code is at ChenJoya/sampling-free.*

1. Introduction

With the resurgence of deep learning [21, 23], recent years have seen remarkable advancement in deep object detectors. Among them, representative successes include two-stage R-CNN [13] and its successors [1, 6, 12, 15, 25, 26, 32, 35, 40]: their proposal stage first generates some candidates from dense, predefined boxes (i.e., anchors [40]), then the second stage uses by a per-region subnetwork for object classification and localization. To pursue higher computational efficiency, one-stage approaches [20, 27, 30, 37, 38, 39, 48, 51] directly recognize objects from the dense anchors without generating candidate proposals. In practice, the anchoring scheme is widely adopted in both one-stage and two-stage detectors, where massive anchors (e.g., $\sim 100k$) are uniformly sampled over an image.

Nevertheless, when training these detectors, only a few anchors that highly overlap with objects would be assigned to foreground examples (e.g., ~ 100), which always results in an extreme imbalance between foreground and background. In previous studies [24, 27, 30, 40, 42], such imbalance may impede the training from convergence, and limit the accuracy as well. More recently, anchor-free detectors [7, 17, 19, 22, 43, 45, 47, 49, 50] have gained much attention due to the replacement of anchors by points or regions (e.g., corner point, center region), but they still suffer from the imbalance caused by the overwhelming number of background points or regions.

In order to alleviate the foreground-background imbalance, numerous solutions have been proposed in recent years, which are summarized in *sampling heuristics* [34]: the hard sampling (e.g., undersampling [40], OHEM [42], IoU-balanced sampling [35]) addresses the imbalance via selecting a subset of examples, whereas the soft sampling (e.g., Focal Loss [27], GHM [24]) puts more focus on some of the examples by weighting. Nevertheless, unlike common class imbalance that is introduced by a biased dataset, the foreground-background imbalance should be attributed to the large searching space of detectors, which means it equally exists in training and inference with the same distribution. However, sampling heuristics will change this distribution, thereby resulting in a misalignment between training and inference. Moreover, they are usually heuristic and require laborious hyper-parameters tuning. For example, in [24], the authors have mentioned that the optimal strategy for GHM is hard to define.

Considering the drawbacks of sampling heuristics, a natural question is: *could the accuracy of sampling-based model be achieved without sampling heuristics?* In the past, it was believed [24, 27, 42] that detectors without sampling would suffer from extreme imbalance and yield meaningless results. Although several objectness cascaded frameworks [20, 39, 48] can solve the imbalance, they will incur

extra computational and memory costs, while most of them still use the sampling heuristics. It seems impossible to train object detectors without sampling to match the accuracy of sampling-based models.

In this paper, we successfully discard sampling heuristics for different types of object detectors, with better accuracy achieved than their vanilla models. Under our observation, the obstacle that impedes detectors without sampling from yielding high accuracy should be attributed to the instability during training. Motivated by this, we propose a *Sampling-Free* mechanism as an alternative to sampling heuristics, which manages to maintain the training stability from initialization and loss of the detector. Specifically, it consists of three schemes: (1) the optimal bias initialization scheme enables the training to be fastly converged under the imbalance; (2) the guided loss scheme avoids the classification loss to be dominated by numerous background examples; (3) the class-adaptive threshold scheme mitigates the confidence shifting problem incurred by the imbalance. Compared with sampling heuristics, the proposed sampling-free mechanism provides a new perspective to address the foreground-background imbalance.

Experiments on the challenging COCO dataset [28] have demonstrated that our sampling-free mechanism not only improves the accuracy of one-stage, two-stage, and anchor-free object detectors, but also yields considerable gains for the instance segmentation task. Moreover, it can support the state-of-the-art Cascade R-CNN [1] to attain higher accuracy. In addition, the sampling-free mechanism requires little hyper-parameters tuning.

2. Related Works

Deep Object Detectors. With the resurgence of deep learning [21, 23], deep object detectors quickly come to dominate the object detection. Among them, two-stage approaches lead the top accuracy on several object detection benchmarks, such as PASCAL VOC [8] and COCO [28]. It firstly generates some object candidates by region proposal stage [40, 44, 52], then determines the accurate object location and category by a per-region subnetwork. A large number of R-CNN variants [1, 6, 12, 15, 25, 26, 32, 35, 40] yield a large improvement in detection performance.

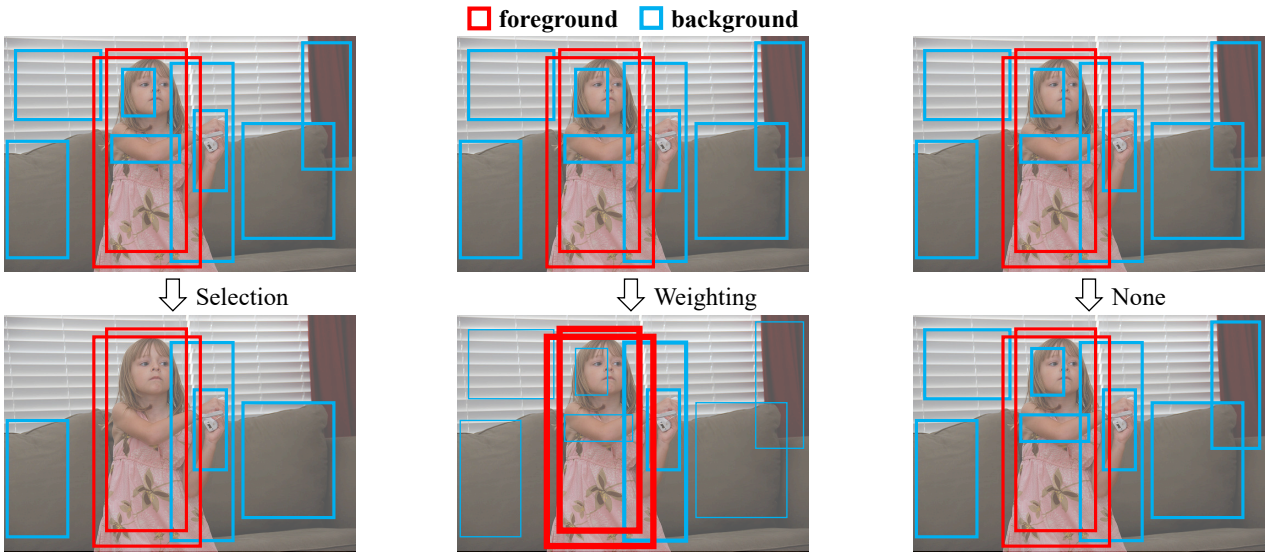
On the other hand, one-stage detectors that popularized by YOLO [37] and SSD [30] are much faster than two-stage approaches due to the elimination of the proposal stage, but have difficulties to match the accuracy of two-stage approaches. A series of advances [20, 27, 38, 39, 48, 51] promote one-stage frameworks to be more accurate. In practice, both one-stage and two-stage approaches rely on a dense anchoring scheme to cover objects, which are also known as anchor-based object detectors. For simplicity, several anchor-free object detectors [7, 19, 22, 43, 45, 47, 49, 50] originated from DenseBox [17] are proposed recently, which achieves better accuracy and computational efficiency than anchor-based detectors.

Class Imbalance. The class imbalance problem has always been an issue in machine learning applications. In such a problem, some classes have much more instances than others, where the standard models tend to be overwhelmed by the majority class. Popular solutions for addressing the imbalance could be divided to three aspects: (1) threshold moving: use lower decision threshold for the minority class [5]; (2) sampling heuristics: a biased selection for specific classes, including undersampling (e.g., EasyEnsemble [31], BalanceCascade [31]) and oversampling (e.g., SMOTE [3]), and the combination of them (e.g., RUSBoost [41]). In practice, some of them [10] also adopt the ensemble technique to obtain a more robust model.

In recent years, the imbalance between foreground and background in object detection has been extensively studied. Due to the overwhelming number of background boxes/points/regions, modern object detectors always suffer from extreme foreground-background imbalance, which results in the popularity of sampling heuristics. In [34], the authors summarizes the sampling heuristics in two groups: (1) hard sampling methods, such as undersampling [40], OHEM [42], and IoU-balanced sampling [35], are in common use in two-stage detectors; (2) soft sampling methods, such as Focal Loss [27], GHM [24], have been widely adopted in one-stage and anchor-free detectors.

However, sampling heuristics may not be the optimal strategy in all imbalanced cases. In [33], it has been demonstrated that for the metric of the ROC curve, sampling methods produce the same result as moving the filtering threshold or adjusting the cost matrix. Moreover, the common class imbalance is usually introduced by the biased dataset, whereas the foreground-background imbalance in object detection is caused by the large searching space of detection frameworks, which equally exists in training and inference. Furthermore, they are heuristic and usually require laborious hyper-parameters tuning. These observations motivate us to explore a sampling-free method.

Multi-Task Weighting. A simple way to weigh different tasks is to introduce extra weights into their loss functions. In recent years, several adaptive weighting methods are proposed. For example, Guo et al. [14] propose to weigh the losses dynamically based on a predefined key performance indicator (e.g., accuracy, average precision) for each task. Other methods can use the network outputs to weigh tasks, such as the uncertainty of the estimations [18] or their loss values [29]. Our guided loss scheme focuses on using the bounding-box regression loss to guide the weighting of classification loss, to eliminate the effects caused by foreground-background imbalance on classification loss. It seems not reported before in the literature of object detection, to our best knowledge.



(a) Hard Sampling (b) Soft Sampling (c) Sampling-Free (Ours)

Figure 1. This figure coarsely illustrates the differences between sampling heuristics and the sampling-free mechanism for addressing the foreground-background imbalance. (a) Hard sampling (e.g., OHEM [42], Undersampling [40], IoU-balanced sampling [35]) selects a subset of examples; (b) Soft sampling (e.g., Focal Loss [27], GHM [2]) uses all examples but focuses on some of them by weighting. For instance, thicker boxes in (b) mean examples with higher weights. (c) Sampling-Free uses all examples without the weighting procedure.

3. Methodology

We observe the imbalance between foreground and background equally exists in training and inference processes, so it is possible that this imbalance would not lower the detection accuracy. However, as shown in Figure 1(a) and Figure 1(b), the sampling heuristics employs the biased selection or weighting strategies for training examples, which changes the imbalanced distribution in training and results in the misalignment between training and inference. Therefore, the sampling-free method in Figure 1(c) may also achieve better accuracy than sampling-based models.

However, previous literatures [24, 27, 42] have illustrated that training an object detector without sampling heuristics (e.g., RetinaNet without Focal Loss [27]) will obtain an extremely poor accuracy. Nonetheless, the underlying reason for this phenomenon was not clearly stated before. In this section, we investigate this issue. Firstly, we perform experiments on RetinaNet [27] without Focal Loss [27] using the COCO benchmark [28], to explore the reasons for the poor accuracy. Based on the discovery of our experiments, we introduce sampling-free mechanism.

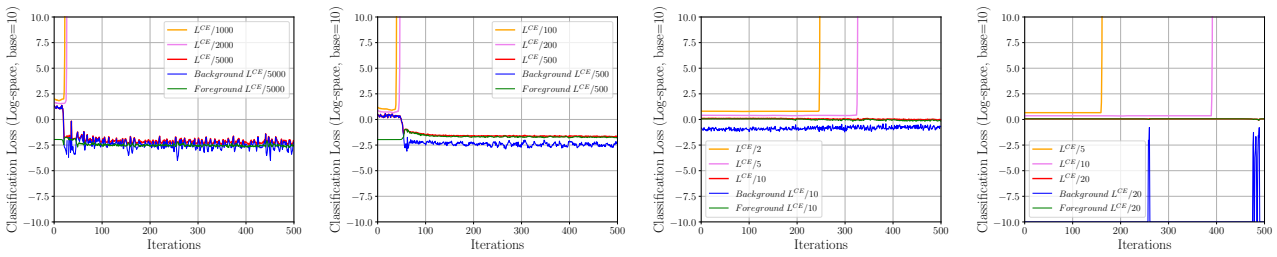
3.1. Investigation for RetinaNet without Focal Loss

In this section, we will discuss the challenges of training RetinaNet without Focal Loss. For quick evaluation, we implement baselines and models on the object detection toolbox maskrcnn-benchmark [9], with a lightweight backbone ResNet-50-FPN [16, 26] to conduct experiments. The training and inference hyper-parameters are kept unchanged from the default settings in maskrcnn-benchmark, e.g., an initial learning rate of 0.01 in $1\times$ schedule ($\sim$12 epochs on COCO), a batch size of 16, an input scale of 1333×800. For simplicity, we call RetinaNet with and without Focal Loss as *RetinaNet-FL* and *RetinaNet-None*, respectively. To implement *RetinaNet-None*, we replace the Focal Loss (L^{FL}) in *RetinaNet-FL* with the standard cross-entropy loss (L^{CE}). Specifically, their loss functions could be represented as:

$$L^{FL} = \frac{1}{N_f}\sum_{i=1}^{A}\sum_{j=1}^{C}[\mathbf{1}_{y^i>0}\alpha(1-p_{ij})^{\gamma}L_{ij}^{FG} + \mathbf{1}_{y^i=0}(1-\alpha){p_{ij}}^{\gamma}L_{ij}^{BG}], \quad (1)$$

$$L^{CE} = \frac{1}{N_f}\sum_{i=1}^{A}\sum_{j=1}^{C}[\mathbf{1}_{y^i=j}L_{ij}^{FG} + \mathbf{1}_{y^i\neq j}L_{ij}^{BG}], \quad (2)$$

where p_{ij} is denoted as the confidence score for j-th class of anchor i. L_{ij}^{FG} and L_{ij}^{BG} are denoted as the loss of a foreground and background example, i.e., $L_{ij}^{FG} = -\log(p_{ij})$ and $L_{ij}^{BG} = -\log(1-p_{ij})$, respectively. The Iverson bracket indicator function $\mathbf{1}_K$ outputs 1 when the condition K is true, otherwise $\mathbf{1}_K = 0$. For Focal Loss, α and γ are the introduced hyper-parameters, with $\alpha = 0.25$ and $\gamma = 2$ to achieve the best accuracy [27]. N_f denotes the number of foregrounds. With these preliminaries, we train *RetinaNet-None* to discover the problems without Focal Loss.



(a) $\pi = 0.5$, 6.9 AP (b) $\pi = 10^{-2}$, 17.9 AP (c) $\pi = 10^{-5}$, 35.6 AP (d) $\pi = 10^{-10}$, 33.8 AP

Figure 2. Classification loss curve of different initialization prior probability π during the warmup period, from *RetinaNet-None*. To avoid gradient exploding, the classification loss is reduced according to the specific π. For instance, the classification loss in (a) should be reduced by 5000 times when $\pi = 0.5$, otherwise the training would fail. Average precision (AP) is evaluated on COCO `minival`.

(a) Varying filtering threshold θ for *RetinaNet-FL*

Threshold	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
$\theta = 0.05$	**36.4**	55.0	39.0	19.9	40.3	48.9
$\theta = 0.01$	**36.4**	55.0	39.0	19.9	40.3	48.9
$\theta = 0.005$	**36.4**	55.0	39.0	19.9	40.3	48.9

(b) Varying filtering threshold θ for *RetinaNet-None*

Threshold	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
$\theta = 0.05$	35.6	53.9	38.2	18.6	39.2	47.4
$\theta = 0.01$	**36.2**	54.7	38.6	19.4	39.9	47.8
$\theta = 0.005$	**36.2**	54.7	38.7	19.7	39.9	47.8

Table 1. Varying filtering thresholds of *RetinaNet-FL* and *RetinaNet-None* ($\pi = 10^{-5}$, $L^{CE}/ = 10$ in Figure 2(c)) on COCO `minival`.

Gradient Exploding and Background Domination. We firstly attempt to train RestinaNet-None without modifying its initialization or loss. It fails quickly, with gradient exploding during training. To solve this, we reduce the weight of the classification loss L^{CE} and retrain the model. As shown in Figure 2(a), the gradient exploding could not be addressed until redecing L^{CE} by 5000 times. However, this setting only achieves 6.9 AP. Compared with 36.4 AP that *RetinaNet-FL* achieved in Table 1(a), *RetinaNet-None* yields a nearly meaningless result.

We observe that the classification loss in Figure 2(a) prematurely decreases into a very low level ($\approx 10^{-2.5}$), which may restrict it to achieve a high accuracy. This may be caused by the domination of the overwhelming number of background examples. At the beginning of training, they would produce a large, unstable loss value, which requires us to down-weight the classification loss to avoid gradient exploding. But after down-weighting it, it seems that the model would "utilize" the imbalanced distribution during training, i.e., the model tends to estimate all examples as background examples to obtain an extremely low loss value. That is why the curve of background loss (Background $L^{CE}/5000$) would suddenly decrease in Figure 2(a). An intuitive idea to solve the problem is to up-weight the classification loss, but this would result in the gradient exploding problem again.

To mitigate the gradient exploding issue and accelerate the learning of the model, we introduce a bias initialization method similar to Focal Loss [27] and GHM [24], to reduce the large loss from background examples at the beginning of training. It initializes the bias term b of the last layer of classification subnet to $b = -\log\frac{1-\pi}{\pi}$ with a prior probability $\pi = 0.01$, so the classification score would be similar to 0.01 after sigmoid activation. Unfortunately, this setting still incurs gradient exploding, and we have to reduce the weight of classification loss by 500 times (see Figrue 2(b)). Despite the accuracy is greatly improved to 17.9 AP, the background classification loss would still suddenly decrease, which suggests that the model still "utilizes" the imbalanced distribution to obtain a lower loss value.

Therefore, we continue to lower the initialization prior probability π to $\pi = 10^{-5}$ and $\pi = 10^{-10}$, and reduce the weight of classification loss to avoid gradient exploding. Surprisingly, as shown in Figure 2(c) and Figure 2(d), these two settings yield much better 35.6 AP and 33.8 AP, respectively. We note that the 35.6 AP of *RetinaNet-None* with $\pi = 10^{-5}$ has already been a comparable performance to 36.4 AP of *RetinaNet-FL*. This result indicates that it is possible to achieve a similar effect to Focal Loss through the adjustment of initialization and loss of the detector.

Confidence Shifting. To further improve the accuracy of *RetinaNet-None*, we analyze its average output confidence scores and found they are much lower than *RetinaNet-FL*. To preserve more foregrounds, we can set a lower filtering threshold during inference. As presented in Table 1, the *RetinaNet-None* yields 36.2 AP at most, whereas the *RetinaNet-FL* has no improvements from that.

Conclusions of Investigation. By careful investigation, we have revealed the problems incurred by the extreme imbalance. Surprisingly, with some empirically introduced techniques, the *RetinaNet-None* achieves a similar accuracy with *RetinaNet-FL*, which motivates us to develop an elegant sampling-free mechanism that enables to train object detectors without sampling heuristics.

3.2. Sampling-Free Mechanism

Inspired by the investigation, we propose sampling-free mechanism, which addresses the foreground-background imbalance from initialization, loss and inference. Our principle is to pursue a simple yet effective solution, and introduce no hyper-parameters.

Optimal Bias Initialization. Section 3.1 has demonstrated the importance of initialization, but it seems to be difficult to find the suitable value of initialization prior. We give an analysis here to determine the optimal initialization prior π.

Suppose there are N_f foregrounds in N examples, with C object categories in the dataset. At the beginning of the training, the confidence scores of examples are similar to the initialization prior probability π. Based on Equation 2, we can reformulate the L^{CE} as:

$$L^{CE} = \underbrace{-\log(\pi)}_{foreground} \underbrace{-(\frac{N}{N_f} \cdot C - 1)\log(1-\pi)}_{background}, \quad (3)$$

where $C = 80$, $\frac{N}{N_f} \approx 1000$ in our experiments. Among $\pi = 0.5, 0.01, 10^{-5}, 10^{-10}$, the setting $\pi = 10^{-5}$ yields the highest accuracy, as shown in Figure 2. A quantitative analysis for them is presented in Table 2.

Initialization Prior	Foreground Loss	Background Loss	AP
$\pi = 0.5$	0.693	55451.081	6.9
$\pi = 10^{-2}$	4.605	804.017	17.9
$\pi = 10^{-5}$	11.513	0.800	35.6
$\pi = 10^{-10}$	23.026	0.000	33.8

Table 2. The loss value of Equation 3 with different π.

We believe that $\pi = 10^{-5}$ brings better training stability than others, which may be the reason for its highest accuracy. On the one hand, it yields the lowest overall loss, which gives it the least possibility of gradient exploding. On the other hand, it prevents the model from utilizing the imbalanced distribution to obtain a lower loss. Specifically, a sudden drop of the background classification loss is only appeared when $\pi = 0.5$ and $\pi = 10^{-2}$ (see Figure 2(a) and Figure 2(b)), which illustrates that the model tends to predict all examples as background examples to reduce large, destabilizing background loss. But this would not occur when $\pi = 10^{-5}$, as it has already yielded a low background loss value. If we continue to lower the π (e.g., $\pi = 10^{-10}$), the foreground loss would be greatly improved, which is not conducive to stable training.

Based on the analysis, we propose to initialize the bias of the last convolutional layer to achieve the minimal value of the loss. The derivative of Equation 3 is:

$$\frac{\partial L^{CE}}{\partial \pi} = -\frac{1}{\pi} + (\frac{N}{N_f} \cdot C - 1)\frac{1}{1-\pi}. \quad (4)$$

Based on Equation 4, when $\pi = \frac{N_f}{N} \cdot \frac{1}{C}$, $\frac{\partial L^{CE}}{\partial \pi}$ is equal to 0, and L^{CE} attains the minimal value. As the bias term b with a sigmoid activation $\sigma()$ is similar with π at the beginning of training, i.e., $\sigma(b) \approx \pi$, we can obtain the initialization value of $b = -\log(\frac{N}{N_f} \cdot C - 1)$. The ratio $\frac{N}{N_f}$ could be calculated from examples before training, which would be efficient as it does not require network forwarding.

After initializing to the minimal value of the loss, the model can no longer "utilize" the imbalanced distribution to obtain a lower loss, which accelerates the convergence of the model. The above analysis is based on multiple independent classifier with sigmoid activation, which is commonly used in one-stage and anchor-free detectors. A softmax version of it is provided in the supplementary material.

Guided Loss. Figure 2 presents the importance of the reduction for classification loss. But the reduction ratio is difficult to be decided. We propose a simple way to determine it as follows. As the regression loss is only calculated for foregrounds, which is not affected by the foreground-background imbalance, we propose guided loss scheme that employs regression loss L^{reg} to guide the reduction of classification loss L^{cls}:

$$r = \frac{w^{reg} L^{reg}}{L^{cls}}, \quad (5)$$

where w^{reg} is the weight of regression loss. Note that Equation 5 is computed without backpropagation in a mini-batch. The total loss L could be written as:

$$L = w^{reg} L^{reg} + w^{cls} L^{cls} \cdot r, \quad (6)$$

where w^{cls} is the weight of classification loss. As the regression loss is not affected by the imbalance, we simply use the default setting $w^{reg} = 1$ in our experiments. w^{cls} needs to be tuned to find the optimal value, but it will be much easier to tune than before. As w^{reg} and w^{cls} are intrinsic parameters of the model, we actually introduce no hyper-parameters here.

Class-Adaptive Threshold. As shown in Table 1(b), moving filtering threshold is a simple yet effective technique to solve confidence shifting. Nevertheless, each class may have its optimal filtering threshold, and it is often time-consuming to adjust them. We propose a class-adaptive threshold scheme, which determines the filtering threshold according to the training data:

$$\theta_j = \frac{N_f}{N} \cdot \frac{N_j}{\sum_{j=1}^{C} N_j}, \quad (7)$$

where θ_j is denoted as the filtering threshold of j-th class, and foreground-to-all ratio $\frac{N_f}{N}$ could be reused from optimal bias initialization scheme. $\frac{N_j}{\sum_{j=1}^{C} N_j}$ means the proportion of j-th class over all classes in the dataset. Therefore, class-adaptive thresholds can be calculated before training.

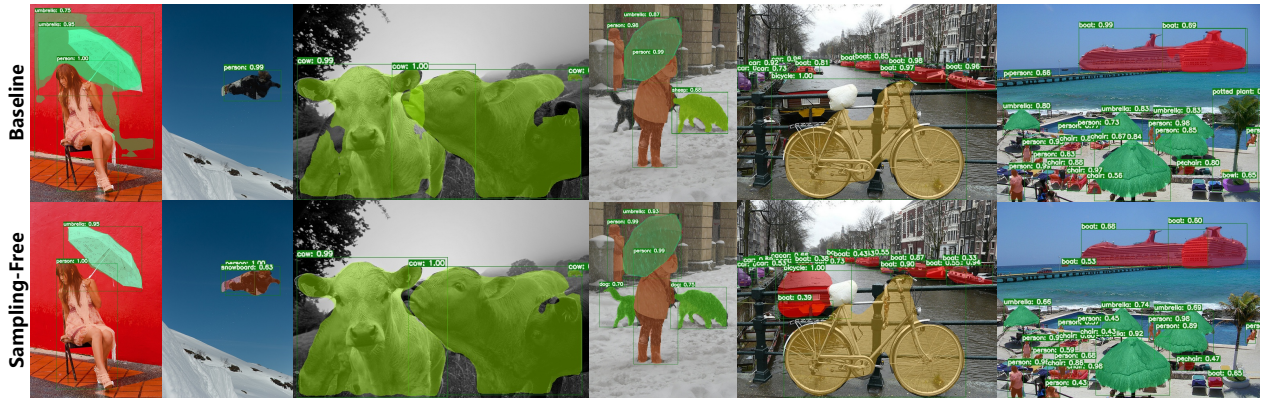


Figure 3. Mask R-CNN [15] (37.8 box AP, 34.2 mask AP on COCO `minival`) vs. Mask R-CNN with our sampling-free mechanism (39.0 box AP, 34.9 mask AP on COCO `minival`) in ResNet-50-FPN backbone. The latter exhibits better detection and segmentation results.

4. Experiments

We present experimental results for the sampling-free mechanism on object detection and instance segmentation tracks. Before that, we describe the corresponding implementation details in Section 4.1, then perform ablation studies in Section 4.2. Section 4.3 reports the improvements for various object detectors. The comparison to prevalent sampling heuristics in the metrics of accuracy, memory cost, and training speed are also discussed.

4.1. Implementation Details

Datasets and Metrics. Following previous works [1, 15, 19, 27, 39, 40], we select the challenging COCO dataset [28] for evaluation, which contains rich bounding-box and instance segmentation annotations over 80 categories. We train models on the `train2017` split ($115k$ images) and perform all ablation studies on `minival` split ($5k$ images), then submit evaluation files to the public server of `test-dev` split ($20k$ images) to report final results. COCO-style average precision (AP) is adopted as the evaluation metrics, including AP, AP_{50}, AP_{75}, AP_S, AP_M, AP_L.

Baselines and Platforms. We perform experiments on six well-known object detectors, which apply sampling heuristics as the default component:

• YOLOv3 [39] (one-stage) is an efficient object detector that utilizes the objectness branch to solve the imbalance. In our experiments, we will replace the objectness with the sampling-free mechanism to improve it.

• RetinaNet [27] (one-stage) is an accurate one-stage object detector, which applies Focal Loss to alleviate the imbalance. With the sampling-free mechanism, we will show that the standard CE Loss works better than Focal Loss.

• Faster R-CNN [40] (two-stage) is one of the most classical object detector, with undersampling used in the proposal and per-region stages. We will validate whether the sampling-free mechanism is effective for the two stages.

• FoveaBox [19] (anchor-free) is a simple and yet effective anchor-free object detector, using Focal Loss to alleviate the imbalance. Like RetinaNet, we will show that sampling-free works well in that.

• Mask R-CNN [15] (two-stage) is a simple, flexible, and general framework for object instance segmentation, with undersampling to address the imbalance in RPN and RoI-subnetwork. We will show that sampling-free could help it to achieve better box AP and mask AP.

• Cascade R-CNN [1] (multi-stage) has a cascaded RoI-subnetwork pipeline to achieve high-quality object detection, with undersampling to address the imbalance in each stage. Using the sampling-free mechanism in all stages, we will show that Cascade R-CNN would be more accurate.

Among them, YOLOv3 is implemented on darknet [36]; RetinaNet and Faster/Mask R-CNN are implemented on maskrcnn-benchmark [9]; FoveaBox are implemented on mmdetection [4], and Cascade R-CNN is implemented on detectron2 [46]. We follow the public standard training configurations to implement them. Specifically, YOLOv3 is trained with DarkNet-53 [39] of 608×608 input resolution configuration that provided in darknet. Other models are trained with ResNet-50-FPN [16, 26] in 1× schedule [11] (∼12 epochs on COCO) of 16 images per batch, with 1333×800 input scale. Note that these hyper-parameters are public configurations, and we have not made any changes.

4.2. Ablation Studies

Optimal Bias Initialization. Without sampling heuristics to alleviate the imbalance, the initialization would be important for training stability. As shown in Table 3(a), the π obtained from our optimal initialization scheme achieves the best performance. We note that the result 35.7 AP is similar with 35.6 AP that $\pi = 10^{-5}$ achieved in Figure 2(c). This is because the *optimal* $\pi \approx 1.113 \times 10^{-5}$ in our experiments, which is nearly the same as $\pi = 10^{-5}$.

(a) Varying initialization prior probability π

Prior	AP	AP_{50}	AP_{75}
$\pi = 10^{-2}$	17.9	27.0	18.8
$\pi = 10^{-10}$	33.8	50.7	36.1
optimal π	**35.7**	**54.1**	**38.2**

(b) Varying weight w_{cls} for guided loss

Weight	AP	AP_{50}	AP_{75}
$w_{cls} = 0.5$	35.8	54.1	38.0
$w_{cls} = 1.0$	**36.1**	**54.5**	**38.7**
$w_{cls} = 1.5$	36.0	54.3	38.6

(c) Varying filtering threshold for inference

Threshold	AP	AP_{50}	AP_{75}
$\theta = 0.05$	36.1	54.5	38.7
$\theta = 0.005$	36.5	55.4	39.0
adaptive θ	**36.6**	**55.5**	**39.0**

Table 3. Ablation studies for RetinaNet with our sampling-free mechanism. These experiments are conducted with ResNet-50-FPN backbone in $1\times$ schedule, and evaluated on COCO `minival`. (a) shows the effectiveness of the optimal bias initialization scheme, (b) explores the optimal weight for classification loss in guided loss. (c) gives an evaluation for class-adaptive threshold scheme.

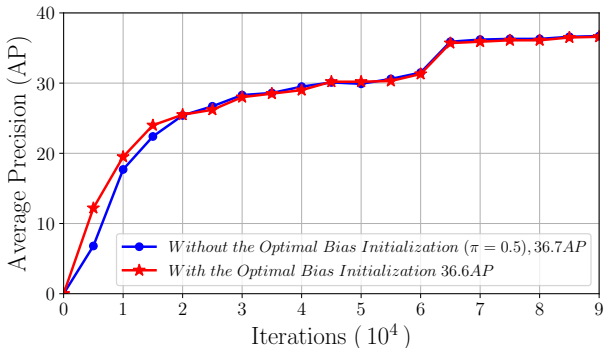


Figure 4. AP variations on COCO `minival` during training, for the sampling-free mechanism with and without bias initialization.

Stage	Sampling-Free Mechanism			
RPN	✗	✓	✗	✓
RoI-subnet	✗	✗	✓	✓
AP	36.8	37.1 (+0.3)	38.1 (+1.3)	**38.4 (+1.6)**
AP_{50}	58.4	58.8 (+0.3)	59.6 (+1.2)	**59.9 (+1.5)**
AP_{75}	40.0	40.2 (+0.2)	41.6 (+1.6)	**41.7 (+1.7)**
AP_S	20.7	21.1 (+0.4)	22.2 (+1.5)	**22.3 (+1.6)**
AP_M	39.7	40.3 (+0.6)	41.2 (+1.5)	**41.6 (+1.9)**
AP_L	47.9	48.1 (+0.2)	50.0 (+2.1)	**50.9 (+3.0)**

Table 4. Results of sampling-free mechanism for Faster R-CNN on COCO `minival`, which is trained with ResNet-50-FPN backbone in $1\times$ schedule. Better performance could be achieved by discarding sampling heuristics in these stages.

Imbalance Solution	AP	Speed (ms)	Memory (MB)
RetinaNet	36.4	171	1816
with *Sampling-Free*	**36.6**	**165**	1816
Faster R-CNN	36.8	**172**	1714
with *Sampling-Free*	**38.4**	184	**1669**

Table 5. Training speed (batch size 1 on a single *Nvidia-Titan Xp* GPU) versus accuracy on COCO `minival`.

Guided Loss. In Section 3.2, we discuss when the guided loss scheme is applied, the weight of the classification loss should be tuned. See Table 3(b), we find that $w_{cls} = 1.0$ works best for RetinaNet, and the accuracy is relatively robust to the specific value. As RPN can be considered as an one-stage detector with binary class, we set $w_{cls} = 1.0$ for all experiments related to RetinaNet and RPN. Furthermore, we discovered $w_{cls} = 1.5$ and $w_{cls} = 2.0$ achieve the best performance for FoveaBox and RoI-subnetwork, respectively. We use these settings when training them.

Class-Adaptive Threshold. Table 3(c) showed that imbalance decision scheme enable the RetinaNet to yield the higher accuracy than setting specific filtering threshold. It can be seen that RetinaNet with sampling-free mechanism (36.6 AP in Table 3(c)) outperforms the RetinaNet with Focal Loss (36.4 AP in Table 1(a)).

Without Bias Initialization. We note that the guided loss could adaptively down-weight the classification loss to the bounding-box regression loss. Therefore, we compare the accuracy of the sampling-free mechanism with and without the optimal bias initialization scheme. As shown in Figure 4, their final accuracy is very similar, but the bias initialization could help the model to be fastly converged. Hence, we still recommend using the optimal initialization in the sampling-free mechanism.

Sampling-Free for Proposal and Per-Region Stages. As shown in Table 4, the vanilla Faster R-CNN achieves 36.8 AP, which uses undersampling in proposal and per-region stages. We gradually incorporate the sampling-free mechanism into different stages and observe the changes in performance. By replacing undersampling with sampling-free mechanism in RPN or RoI-subnetwork, better accuracy could be always obtained. Finally, we apply the sampling-free mechanism for all stages, which yields an impressive 1.6 AP improvements, with the gains from all AP metrics.

Training Speed and Memory Cost. As sampling heuristics would introduce extra computation on loss function, or drop some examples during training, sampling-free mechanism may have some impacts on training speed and memory cost. We gave a quantitative analysis in Table 5. Collaborated with sampling-free rather than Focal Loss, the training of RetinaNet becomes faster, with the equivalent memory cost. Interestingly, although the training speed becomes slower due to the time-consuming RoI-subnetwork for Faster R-CNN with sampling-free, its memory cost could be reduced. The reason is that the sampling procedure also requires considerable memory costs, which can be ignored when the sampling-free mechanism is applied.

Method	Platform	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv3 [39]	darknet	DarkNet-53	33.0	57.9	34.4	18.3	35.4	41.9
YOLOv3 (*Sampling-Free*)			**33.9**	**58.4**	**35.5**	**18.4**	**35.7**	**43.2**
RetinaNet [27]	maskrcnn-benchmark	ResNet-50-FPN	36.6	56.0	39.1	19.5	38.9	**47.1**
RetinaNet (*Sampling-Free*)			**36.8**	56.0	**39.4**	**19.9**	**39.7**	46.6
FoveaBox [19]	mmdetection		36.7	56.6	39.2	20.3	39.9	45.3
FoveaBox (*Sampling-Free*)			**36.8**	**56.8**	39.2	**20.6**	**40.2**	**45.4**
Faster R-CNN [40]	maskrcnn-benchmark		37.2	59.3	40.3	21.3	39.5	46.9
Faster R-CNN (*Sampling-Free*)			**38.7**	**60.5**	**42.2**	**22.2**	**41.0**	**49.1**
Cascade R-CNN [1]	detectron2		41.8	59.8	45.2	24.3	44.3	52.5
Cascade R-CNN (*Sampling-Free*)			**42.8**	**60.3**	**46.9**	**25.2**	**45.7**	**54.1**
RetinaNet [27]	maskrcnn-benchmark	ResNet-101-FPN	38.8	58.4	41.7	20.9	41.7	**49.5**
RetinaNet (*Sampling-Free*)			**39.0**	58.4	**41.8**	**21.3**	**42.0**	49.1
FoveaBox [19]	mmdetection		38.4	58.2	41.1	**21.5**	42.0	47.2
FoveaBox (*Sampling-Free*)			**38.6**	**58.6**	**41.3**	21.4	**42.2**	**47.6**
Faster R-CNN [40]	maskrcnn-benchmark		39.3	61.4	42.7	22.1	41.9	50.1
Faster R-CNN (*Sampling-Free*)			**40.7**	**62.4**	**44.3**	**23.0**	**43.5**	**51.8**
Cascade R-CNN [1]	detectron2		43.4	61.6	47.1	24.7	46.2	54.4
Cascade R-CNN (*Sampling-Free*)			**44.4**	**62.4**	**48.3**	**26.0**	**46.8**	**55.9**

Table 6. Bounding-box AP results on COCO `test-dev`. For various backbones and detectors, sampling-free mechanism could yield improvements, especially for two-stage and multi-stage approaches.

4.3. Results

Sampling-Free for Various Object Detectors. As presented in Table 6, we apply the sampling-free mechanism for various object detectors to validate its effectiveness. For both lightweight (DarkNet-53 [36], ResNet-50-FPN [16, 26]) and heavy backbones (ResNet-101-FPN [16, 26]), we observe that the detectors combined with the sampling-free mechanism always yield better performance than the vanilla models. Among them, the two-stage and multi-stage approaches obtain large gains, where the sampling-free mechanism yield $1.4 \sim 1.5$ AP improvements for Faster R-CNN, 1.0 AP improvements for Cascade R-CNN, respectively. Although the improvements on one-stage and anchor-free object detectors are not impressive, our focus is to show the foreground-background imbalance is not the obstacle that impedes them from achieving high accuracy. Moreover, sampling heuristics usually requires much more hyper-parameters tuning than our method.

However, it seems like GHM [24] shows better accuracy than Focal Loss [27] and our sampling-free mechanism on COCO. We observe the GHM-R scheme in GHM is not for solving the foreground-background imbalance, so sampling-free with GHM-R may obtain similar results. Table 7 proves this, and the sampling-free mechanism produces a 27% acceleration during training.

Sampling-Free for Instance Segmentation. In Table 8, when we apply the sampling-free mechanism for Mask R-CNN [15], 1.2 box AP and 0.7 mask AP gains are obtained. We also visualize the its results in Figure 3, which shows that the Mask R-CNN with the sampling-free mechanism exhibits better detection and segmentation effects.

Regression	Classification	AP	Speed (ms)
Smooth L1 [12]	GHM-C [24]	36.3	328
	Sampling-Free	**36.4**	**238**
GHM-R [24]	GHM-C [24]	36.9	336
	Sampling-Free	36.9	**244**

Table 7. Results of GHM [24] and sampling-free mechanism on COCO `minival`. "Speed (ms)" refers to training speed. Implementation is based on the code of GHM in mmdetection [4].

(a) Box AP of Mask R-CNN on COCO `minival`

Model	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN [15]	37.8	59.3	41.1	21.5	41.1	49.9
with *Sampling-Free*	**39.0**	**60.3**	**42.5**	**22.5**	**41.9**	**51.2**

(b) Mask AP of Mask R-CNN on COCO `minival`

Model	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN [15]	34.2	55.9	36.3	15.6	36.8	50.6
with *Sampling-Free*	**34.9**	**56.8**	**37.1**	**16.2**	**37.3**	**51.2**

Table 8. Results of sampling-free mechanism for Mask R-CNN.

5. Conclusion

In this paper, we explore whether sampling heuristics is necessary for training object detectors. Our empirical study reveals that weight of classification loss and the initialization strategy is the key to maintain the training stability and achieve the accuracy of sampling-based models. Inspired by this, we propose the *Sampling-Free* mechanism as an alternative to sampling heuristics. Extensive experiments demonstrate that the sampling-free mechanism could help various object detectors to achieve better accuracy on the challenging COCO benchmark. Moreover, it also yields considerable gains for instance segmentation task.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018.

[2] Yuhang Cao, Kai Chen, Chen Change Loy, and Dahua Lin. Prime sample attention in object detection. *CoRR*, abs/1904.04821, 2019.

[3] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority oversampling technique. *J. Artif. Intell. Res.*, 16:321–357, 2002.

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Mmdetection: Open mmlab detection toolbox and benchmark. *CoRR*, abs/1906.07155, 2019.

[5] Guillem Collell, Drazen Prelec, and Kaustubh R. Patil. A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data. *Neurocomputing*, 275:330–340, 2018.

[6] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: object detection via region-based fully convolutional networks. In *NIPS*, pages 379–387, 2016.

[7] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, pages 6569–6578, 2019.

[8] Mark Everingham, Luc J. Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[9] Massa Francisco and Girshick Ross. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. `https://github.com/facebookresearch/maskrcnn-benchmark`, 2018.

[10] Mikel Galar, Alberto Fernández, Edurne Barrenechea Tartas, Humberto Bustince Sola, and Francisco Herrera. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Trans. Systems, Man, and Cybernetics, Part C*, 42(4):463–484, 2012.

[11] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. `https://github.com/facebookresearch/detectron`, 2018.

[12] Ross B. Girshick. Fast R-CNN. In *ICCV*, pages 1440–1448, 2015.

[13] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[14] Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In *ECCV*, pages 282–299, 2018.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[17] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. Densebox: Unifying landmark localization with end to end object detection. *CoRR*, abs/1509.04874, 2015.

[18] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pages 7482–7491, 2018.

[19] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, and Jianbo Shi. Foveabox: Beyond anchor-based object detector. *CoRR*, abs/1904.03797, 2019.

[20] Tao Kong, Fuchun Sun, Anbang Yao, Huaping Liu, Ming Lu, and Yurong Chen. RON: reverse connection with objectness prior networks for object detection. In *CVPR*, pages 5244–5252, 2017.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, 2017.

[22] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 765–781, 2018.

[23] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[24] Buyu Li, Yu Liu, and Xiaogang Wang. Gradient harmonized single-stage detector. In *AAAI*, pages 8577–8584, 2019.

[25] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *ICCV*, pages 6054–6063, 2019.

[26] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944, 2017.

[27] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2999–3007, 2017.

[28] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014.

[29] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *CVPR*, pages 1871–1880, 2019.

[30] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. In *ECCV*, pages 21–37, 2016.

[31] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Trans. Systems, Man, and Cybernetics, Part B*, 39(2):539–550, 2009.

[32] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid R-CNN. In *CVPR*, pages 7363–7372, 2019.

[33] M. Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. In *ICML 2003 Workshop on Learning from Imbalanced Data Sets II*, pages 73–80, 2003.

[34] Kemal Oksuz, Baris Can Cam, Sinan Kalkan, and Emre Akbas. Imbalance problems in object detection: A review. *CoRR*, abs/1909.00169, 2019.

[35] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra R-CNN: towards balanced learning for object detection. In *CVPR*, pages 821–830, 2019.

[36] Joseph Redmon. Darknet: Open source neural networks in c. `http://pjreddie.com/darknet/`, 2013–2016.

[37] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016.

[38] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *CVPR*, pages 6517–6525, 2017.

[39] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[40] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017.

[41] Chris Seiffert, Taghi M. Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Rusboost: A hybrid approach to alleviating class imbalance. *IEEE Trans. Systems, Man, and Cybernetics, Part A*, 40(1):185–197, 2010.

[42] Abhinav Shrivastava, Abhinav Gupta, and Ross B. Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, pages 761–769, 2016.

[43] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: fully convolutional one-stage object detection. In *ICCV*, pages 9627–9636, 2019.

[44] Jasper R. R. Uijlings, Koen E. A. van de Sande, Theo Gevers, and Arnold W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

[45] Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. Region proposal by guided anchoring. In *CVPR*, pages 2965–2974, 2019.

[46] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[47] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *ICCV*, pages 9657–9666, 2019.

[48] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z. Li. Single-shot refinement neural network for object detection. In *CVPR*, pages 4203–4212, 2018.

[49] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *CoRR*, abs/1904.07850, 2019.

[50] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krähenbühl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, pages 850–859, 2019.

[51] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *CVPR*, pages 840–849, 2019.

[52] C. Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, pages 391–405, 2014.