Overlap Sampler for Region-Based Object Detection

Joya Chen[1,2], Bin Luo[1], Qi Wu[3], Jia Chen[4], and Xuezheng Peng *[1]

[1]Tencent
[2]University of Science and Technology of China
[3]Institute of Intelligent Machines, Chinese Academy of Sciences
[4]South China University of Technology

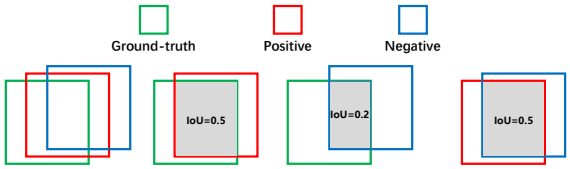
Abstract

The top accuracy of object detection to date is led by region-based approaches, where the per-region stage is responsible for recognizing proposals generated by the region proposal network. In that stage, sampling heuristics (e.g., OHEM, IoU-balanced sampling) is always applied to select a part of examples during training. But nowadays, existing samplers ignore the overlaps among examples, which may result in some low-quality predictions preserved. To mitigate the issue, we propose Overlap Sampler *that selects examples according to the overlaps among examples, which enables the training to focus on the important examples. Benefitted from it, the Faster R-CNN could obtain impressively 1.5 points higher Average Precision (AP) on the challenging COCO benchmark, a state-of-the-art result among existing samplers for region-based detectors. Moreover, the proposed sampler also yields considerable improvements for the instance segmentation task. Our code is released at* `https://github.com/ChenJoya/overlap-sampler`.

1. Introduction

Deep object detectors become prevalent since the success of Region-based CNN (R-CNN [13]). R-CNN-like detectors [2, 4, 12, 14, 23, 28, 32, 37] usually work in two stages: the region proposal network (RPN [32]) first generates some candidate regions, then followed by a per-region stage for refining the locations, classifying the categories of these candidate regions. Despite various detection frameworks proposed over years (e.g., one-stage [25, 27, 30, 31, 41, 44] and anchor-free [17, 20, 21, 29, 34, 40, 42, 43] approaches), region-based object detectors still lead the top accuracy on most benchmarks [7, 10, 26].

*Corresponding author: Reuspeng@tencent.com



(a) $IoU^{g,p} = 0.5$, $IoU^{g,n} = 0.2$, $IoU^{p,n} = 0.5$

(b) $IoU^{g,p} = 0.5$, $IoU^{g,n} = 0.2$, $IoU^{p,n} = 0.1$

Figure 1. The overlaps between ground-truths and examples are quite different from those between examples themselves. See this figure, the IoU between the ground-truth and the positive ($IoU^{g,p}$), as well as the IoU between the ground-truth and the negative ($IoU^{g,n}$) in (a), is equal to that in (b). However, the overlaps between examples themselves, i.e. the IoU between the positive and the negative ($IoU^{p,n}$) has obvious difference in (a) and (b).

Nevertheless, previous works [28, 32, 33] have demonstrated that the imbalance between positives and negatives would impede region-based detectors to attain higher accuracy. Specifically, the number of negative examples is much larger than that of positive examples during training (e.g., $100k$ vs. 100). Although the RPN could remove most negatives, they still account for ~90% in the remaining examples at the per-region stage, which may cause the training dominated by huge negatives.

To alleviate the imbalance between positives and negatives, *sampling heuristics* [19] is widely adopted for training object detectors, such as loss-based sampling [22, 25, 33] and IoU-based sampling [3, 6]. For region-based detectors, the latter shows higher efficiency as it only selects a part of examples to train, thus eliminating extra computational cost incurred by loss-based sampling. However, existing IoU-based samplers only consider the overlaps between ground-

truths and examples, but ignore the overlaps among examples themselves. As shown in Figure 1, these two types of overlaps are quite different. We believe that taking the overlaps among examples into account would be beneficial to improve the detection accuracy, which would be beneficial to the better non-maximum suppression (NMS[1]) procedure. Take the Figure 1 as an instance: the negative example has a high overlap with the positive example, which may cause the latter to be wrongly suppressed at the NMS procedure. By taking the overlaps among examples into account during sampling, we believe that the training could put more focuses on this case.

To utilize the overlaps among examples during sampling, we propose *Overlap Sampler* that selects training examples according to the overlaps among them. Current IoU-based samplers (e.g., IoU-balanced sampling [28]) always assign sampled probability by overlaps between ground-truths and examples. In contrast, the proposed overlap sampler is based on the overlaps among examples. Our analysis reveals that the overlap sampler could achieve higher upper bound in accuracy than other IoU-based samplers, as it helps the training to put more focuses on those highly overlapped cases. Therefore, a detector with the overlap sampler would tend to preserve the relatively high-quality results from multiple candidate proposals.

To validate the effectiveness of the overlap sampler, we incorporate it into two well-known region-based detectors, Faster R-CNN [32] and Mask R-CNN [14], and evaluate their performances on the challenging COCO [26] benchmark. Our experiments show that with the ResNet-50-FPN backbone [15, 24], the upgraded Faster R-CNN, Mask R-CNN could obtain 1.5 box AP, 0.8 mask AP improvements, respectively. With a strong backbone of ResNext-101-FPN [24, 39], we observed that the Faster R-CNN combined with our overlap sampler achieves 42.5 AP, surpassing existing sampling heuristics in region-based detectors.

Our main contributions are as follows:

• By a careful investigation for IoU-based sampling heuristics, we reveal the overlaps among examples have a tremendous impact on the detection accuracy.

• Motivated by this, we propose *Overlap Sampler* to improve region-based detectors, which selects training samples according to the overlaps among examples.

• Extensive experiments have demonstrated that overlap sampler is more effective than existing sampling heuristics. Without any bells and whistles, it improves the 1.5 box AP and 0.8 mask AP for Faster R-CNN and Mask R-CNN on the challenging COCO benchmark, respectively.

[1]NMS [5, 1] algorithm is widely adopted in object detection frameworks, which is responsible for removing highly overlapped bounding-boxes. While running the NMS algorithm, a bounding-box with the maximum detection score is selected and its neighboring boxes are suppressed using a predefined IoU threshold (e.g.,0.5).

2. Related Work

Classic Object Detectors. Before the boom of deep learning, the sliding-window paradigm and hand-crafted features were widely used in object detection. Well-known representatives include face detection by Viola and Jones [36] and pedestrian detection by DPM [8]. However, recent years have witnessed the outstanding performance of CNN-based general-purpose object detectors, which outperform the classic detectors by a large margin on the object detection benchmarks [7, 26].

Region-Based Detectors. Region-based detector is also termed as the two-stage detector, which is introduced and popularized by R-CNN [13]. It firstly generates a sparse set of candidates by some low-level vision algorithms [35, 45], then determines the accurate bounding boxes and the classes by convolutional networks. A number of R-CNN variations [4, 12, 14, 32] appear over years, yielding a large improvement in detection accuracy. Among them, Faster R-CNN [32] is one of the most successful approach. It introduces the region proposal network (RPN) [32, 38], which has been a standard module in region-based approaches.

Sampling Heuristics for Region-Based Detectors. Although the foreground-background class imbalance has been greatly alleviated by RPN, the overwhelming number of the negatives still dominate the training procedure. The methods for handling the imbalance can be divided into two categories: (1) loss-based sampling, such as OHEM [33], Focal Loss [25] and GHM [22]. (2) IoU-based sampling, e.g.,IoU-balanced sampling [28], ISR [3]. The loss-based sampling methods, however, require the losses of all candidate boxes, which will introduce considerable memory and computing costs. On the other hand, despite the random sampling has higher efficiency than hard mining, but as illustrated in the previous works [25, 28, 33], it usually samples excessive easy negatives such that leads to inefficient training. Recent IoU-based sampling methods managed to solve this dilemma. Specifically, the IoU-balanced sampling tends to select the negative example which has high overlap with ground-truth objects, while the ISR is likely to focus on the positive examples of high overlaps with ground-truth objects. Beyond them, our overlap sampler also considers the overlaps among examples during sampling, which has not been explored before.

Non-Maximum Suppression. Non-maximum suppression (NMS) has been an integral part of many detection algorithms. Popular greedy NMS is proposed by Dalal and Triggs [5], where a bounding box with the maximum detection score is selected and its neighboring boxes are suppressed using a predefined IoU threshold. Recently, several works [1, 16, 18] attempt to improve its performance from the perspective of the network. In contrast, we focus on the sampling procedure to avoid incorrect suppression.

Method	Algorithm	Condition	Upper bound AP
IoU-balanced [28]	*Evenly sample negatives in each IoU bin*	$S_n = 0$	44.2
ISR [3]	*Sample and reweight prime positives*	$S_p = IoU^{g,p}$	45.1
Overlap Sampler (Ours)	*Sample and reweight highly overlapped examples*	$S_{e_i} = IoU^{g,e_i}$, $S_{e_j} = IoU^{g,e_j}$	**49.5**

Table 1. An empirical analysis of the upper bound in accuracy for different IoU-based sampling heuristics. We use Faster R-CNN [32] with ResNet-50-FPN [15, 24] backbone implemented on maskrcnn-benchmark [9] to analyze the upper bound on COCO `minival` [26]. The condition to achieve the upper bound for each sampling method is described as follows: (a) IoU-balanced sampling: For any negative n, the predicted score satisfies $S_n = 0$. (b) ISR: For any positive p and the corresponding ground-truth g, the predicted score satisfies $S_p = IoU^{g,p}$. (c) Overlap sampler: For any overlapped examples e_i and e_j, the predicted score satisfies $S_{e_i} = IoU^{g,e_i}$, $S_{e_j} = IoU^{g,e_j}$. In our experiments, the overlap sampler attains the highest upper bound in accuracy.

3. Methodology

In this section, we introduce the proposed overlap sampler starting from an investigation for different IoU-based sampling heuristics, which will show the advantages of the sampling according to overlaps among examples. Specifically, we will perform an empirical analysis of the upper bound in accuracy of IoU-balanced sampling [28], ISR [3], and our overlap sampler. Based on the investigation, the overlap sampler is proposed, which could take the overlaps among examples themselves into account during sampling.

For simplicity, we follow the Figure 1 to denote the overlap[2] between example e_i and example e_j as IoU^{e_i,e_j}. Furthermore, for the sake of fairness, all of our experiments and baselines are implemented on maskrcnn-benchmark [9] with the same training and inference configurations, e.g., the backbone is ResNet-50-FPN [15, 24], the learning rate is 0.02 with $1\times$ schedule ($\sim$12 epochs on COCO [26]), the input scale is 1333×800.

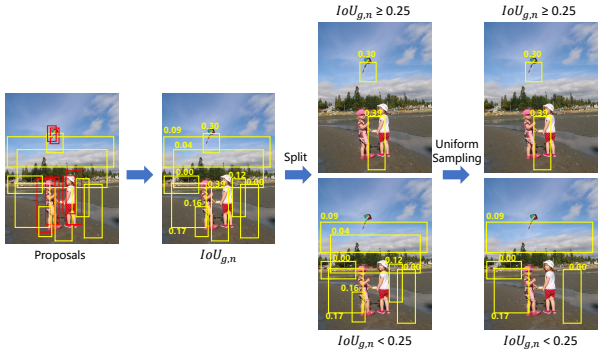
3.1. Investigation

As shown in Table 1, an empirical analysis is performed to estimate the upper bound in accuracy of different IoU-based sampling heuristics. We will describe how they select examples, then discuss the conditions for them to achieve the upper bound.

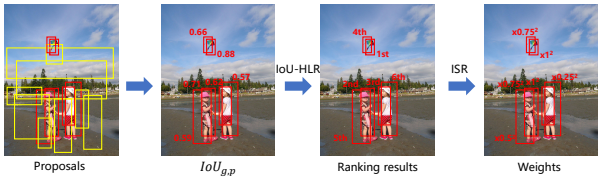
IoU-balanced sampling. IoU-balanced sampling is the sampling part of Libra R-CNN [28]. As shown in Figure 2(a), it evenly splits the sampling interval into K bins (an example of $K = 2$ is visualized in the figure) according to IoU between ground-truths and negatives and selects samples from them uniformly. Therefore, its optimal accuracy would be achieved if all negatives could be accurately recognized. In the first row of the Table 1, we set the predicted scores of all negatives to zero (i.e., $S_n = 0$) and obtain 44.2 AP on COCO `minival`.

ISR. Importance-based sample reweighting (ISR) belongs to the classification part of PISA [3]. It hopes to measure the importance of different examples, then selects the prime ones to train. As shown in Figure 2(b), an IoU-HLR algorithm [3] is developed to rank the importance of different positives. Then ISR assigns higher weights for the "prime examples" (the positives with higher overlaps to their corresponding ground-truths). Therefore, as presented in the second row of the Table 1, we set the predicted scores of all positives to $IoU^{g,p}$ to estimate the upper bound of ISR. It is observed that ISR achieves 45.1 AP on COCO `minival`, which is a 0.9 AP higher result than the upper bound of IoU-balanced sampling.

In the above, ISR has shown that different positives should be weighted according to their overlaps with the ground-truths. Compared with IoU-balanced sampling, the gain of ISR in the upper bound is from the improvement of NMS, as the weighting scheme would help the detector to output a higher score for a higher-quality positive example. Motivated by this, we propose an overlap sampler that directly focuses on those NMS-related examples.



(a) An example of IoU-balanced sampling [28].



(b) An example of Importance-based sample reweighting (ISR [3]).

Figure 2. We give two visualization examples for the pipeline of IoU-balanced sampling and ISR, to make the analysis of the upper bound more clear. IoU-balanced sampling uniformly selects negatives (yellow) from evenly split bins of overlaps between ground-truths and negatives, whereas ISR weights positives (red) according to the ranking results produced by IoU-HLR [3].

[2]To avoid conflict, we use "overlap" to refer to the intersection-over-union (IoU), but use "IoU" in the mathematical formulas.

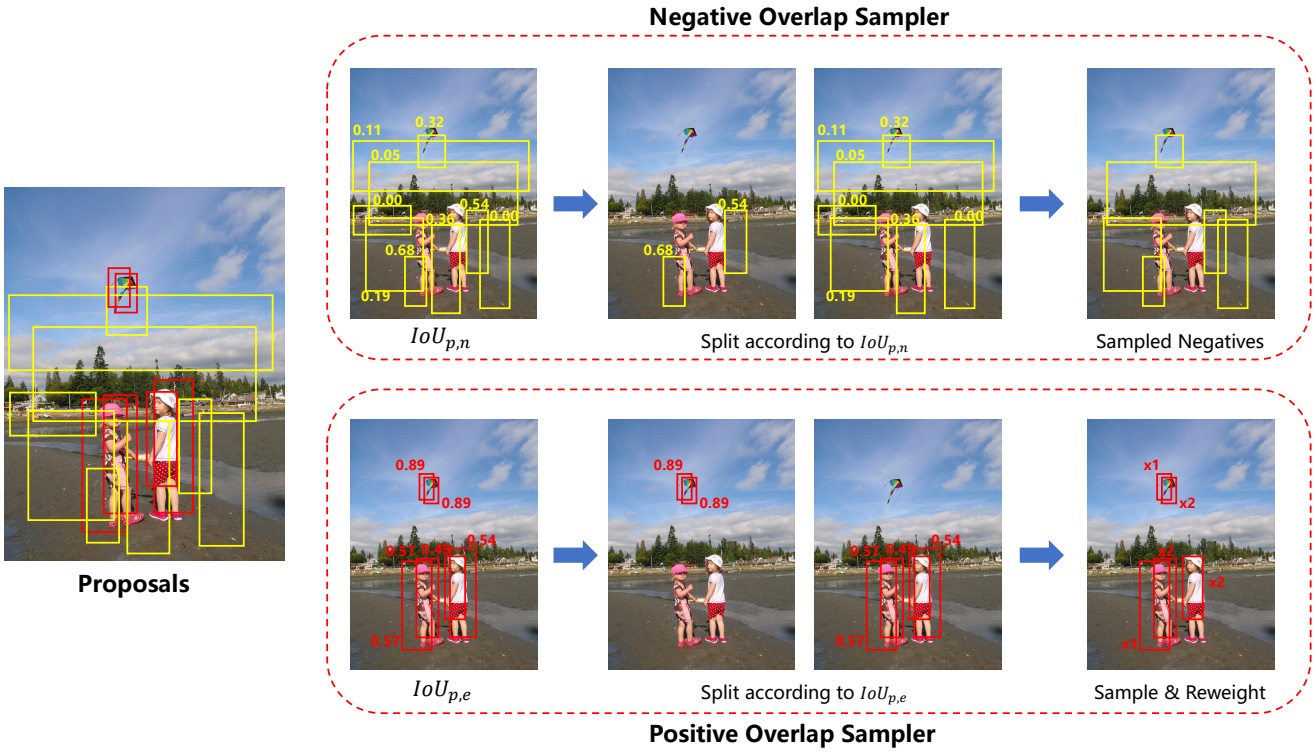


Figure 3. This figure illustrates the pipeline of our overlap sampler, which consists of the positive overlap sampler and the negative overlap sampler. Both of them select examples according to the overlaps between positives (red) and negatives (yellow), i.e., $IoU^{p,n}$. To better learn the overlapped positives, the positive sampler also applies a reweighting scheme, to focus on those examples with higher $IoU^{g,p}$.

3.2. Overlap Sampler

In the above, we have illustrated that, by accurately recognizing the examples that highly overlapped examples, the detection accuracy would be significantly improved. To achieve this goal, it is natural to select more highly overlapped examples during sampling. In this section, we introduce the overlap sampler, which considers the overlaps among examples to sample. As shown in Figure 3, it consists of two parts, for sampling positive and negative examples. Since the number of positives is always not enough, we first introduce the main negative overlap sampler.

Negative Overlap Sampler. Let's start by revisiting the mini-batch random sampling at the per-region stage in Faster R-CNN [32]. After the proposal stage (RPN), there are ˜2000 candidate proposals, in which most of them are negatives. Rather than use all of them, a common method is to sample 512 proposals in an image to compute the loss function of a mini-batch, where the sampled positive and negative anchors have a ratio of up to 1:3. If there are fewer than 128 positive samples in an image, we pad the mini-batch with negative ones. Generally, the examples of $IoU^{g,e} >= 0.5$ and $IoU^{g,e} < 0.5$ are assigned to be positives and negatives, respectively. Our goal is to sample a subset from all negatives and combine them with the sampled positives to a mini-batch.

Different from random sampling and IoU-balanced sampling, our overlap sampler takes the overlaps among examples into account. According to the analysis in Section 3.1, we hope to sample more negatives with higher $IoU^{p,n}$ here. Suppose we need to sample N negative examples from M corresponding candidate negatives without replacement. For the i-th negative example, its sampled probability and the maximum $IoU^{p,n}$ are denoted as p_i and $IoU_i^{p,n}$, respectively. By these definitions, we design several methods to set the sampled probability.

***Uniform* probability sampling**: This strategy is completely the same as random sampling. Each example has a uniform sampled probability of $p_i = N/M$.

***Hard* probability sampling**: Analysis in Section 3.1 reveals that the highly overlapped examples should be accurately recognized. Naturally, we can sample all $IoU_i^{p,n} >= \theta$ examples to train, where θ is the NMS threshold:

$$U = \sum_{i=1}^{M} \mathbf{1}_{IoU_i^{p,n} >= \theta}. \quad (1)$$

In Equation 1, $\mathbf{1}()$ denote the indicate function, and U denote the number of sampled negatives. After that, we apply a *uniform* probability sampling to sample the $N - U$ negatives with $IoU_i^{p,n} < \theta$.

***Soft* probability sampling**: The *hard* probability sampling strategy may lead the detector excessively focus to the $IoU_i^{p,n} >= \theta$ negative examples. Furthermore, the number of $IoU_i^{p,n} >= \theta$ examples is always not enough, which would waste some examples with $IoU_i^{p,n} < \theta$. Hence, we introduce a *soft* probability sampling method here, which is similar to the IoU-balanced sampling [28]. Specifically, we first evenly split the sampling interval into K bins according to $IoU^{p,n}$. Then, we select samples from them uniformly:

$$p_i = \frac{N}{K} \cdot \frac{1}{N_k}, IoU_i^{p,n} \in [\frac{k \cdot IoU_{max}^{p,n}}{K}, \frac{(k+1) \cdot IoU_{max}^{p,n}}{K}), \tag{2}$$

where N_k denotes the number of sampling candidates in the corresponding interval and k denotes the index of each interval ($k \in [0, K)$). The $IoU_{max}^{p,n}$ is the maximum $IoU^{p,n}$ in the interval[3]. Figure 3 shows an example of the *soft* probability sampling.

***Linear* probability sampling**: It simply adopts the normalized $IoU^{p,n}$ as the sampled probability.

$$p_i = \frac{IoU_i^{p,n}}{\sum_{i=1}^{M} IoU_i^{p,n}} \cdot N. \tag{3}$$

However, this sampling method would not select the negatives with $IoU^{p,n} = 0$, which performs worse than the *hard* and *soft* probability sampling. We will further discuss them in Section 4.

Positive Overlap Sampler. The negative overlap sampler is responsible for sampling more negatives with high $IoU^{p,n}$. To better collaborate with it, as shown in Figure 3, we propose a positive overlap sampler to sample positives with high $IoU^{p,e}$, which means the overlap between positives and all examples. For simplicity, we use the same probability sampling methods in the negative overlap sampler.

Nevertheless, sampling positives is more complicated than sampling negatives. As the number of positives is often not enough during training, it is common to sample all positives. However, the sampled positives always have different quality (i.e., $IoU^{g,p}$), and the positives with higher $IoU^{g,p}$ are more important for improving accuracy, which suggests that it is not appropriate to train them equally.

To address the issue, ISR [3] proposes IoU-HLR [3] and CARL [3] techniques to focus on the prime examples with high $IoU^{g,p}$. However, they usually incur the extra computational cost. Instead, we propose a simple loss reweighting scheme to supplement the positive overlap sampler. For each ground-truth, we find its best matched positive example, and multiple their weights by a factor ϵ during loss computing. In this way, the positive overlap sampler can not only sample more positives that highly overlap with negative examples but also support the detector to predict more high-quality positives.

[3]The IoU interval in IoU-balanced sampling is $[0, 0.5)$, which corresponds to the interval of negatives. However, in negative overlap sampler, we consider the overlaps among examples to sample. As the RPN uses the NMS of 0.7 thresholds, the IoU interval in the per-region stage is $[0, 0.7)$.

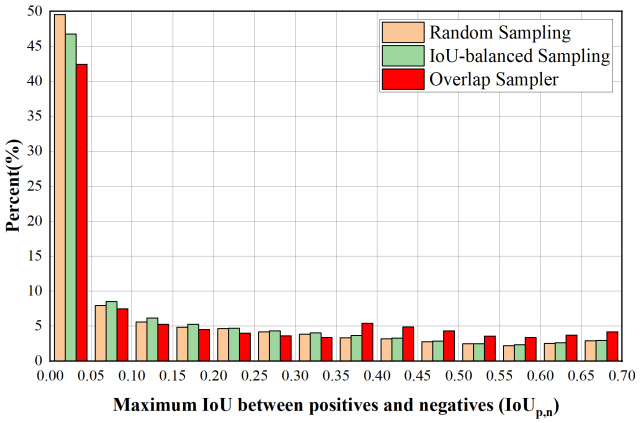


Figure 4. This figure illustrates IoU distribution of negatives from different sampling heuristics, including random sampling [32], IoU-balanced sampling [28] with 2 IoU bins, and our overlap sampler with a *soft* probability sampling strategy in $K = 2$. The IoU mentioned here denotes the maximum IoU between positives ($IoU^{g,p} \geq 0.5$) and negatives ($IoU^{g,n} < 0.5$).

Differences. To address the foreground-background imbalance, numerous sampling heuristics [3, 28, 33] and reweighting schemes [22, 25] are proposed in recent years. Although some of them are widely used in one-stage detectors, they are not popularized in region-based detectors due to the extra computational cost. Specifically, OHEM [33], Focal Loss [25] and GHM [22] are driven by the loss values of examples, which require computing the loss values for all proposals at the time-consuming per-region stage. On the other hand, random sampling has much higher efficiency, but it usually results in easy domination problem [25]. Fortunately, recent IoU-based sampling methods [3, 28] solve this dilemma, which can select more effective examples based on overlaps with ground-truths rather than introducing extra loss computational cost.

In fact, our overlap sampler can also be regarded as an IoU-based sampling method, as it selects examples according to overlaps among examples. In Table 1, we have demonstrated that the upper bound in accuracy of overlap sampler is higher than that of IoU-balanced sampling and ISR. To highlight the uniqueness of our overlap sampler, we also compare the distribution of examples produced by different sampling heuristics. As shown in Figure 4, we visualize the IoU distribution of random sampling, IoU-balanced sampling, and our overlap sampler selected negatives. It can be seen that in the sampled negatives, both random sampling and IoU-balanced sampling select little examples with high $IoU^{p,n}$. In contrast, our overlap sampler can easily sample more examples with high $IoU^{p,n}$, to help the detector to focus on them during training.

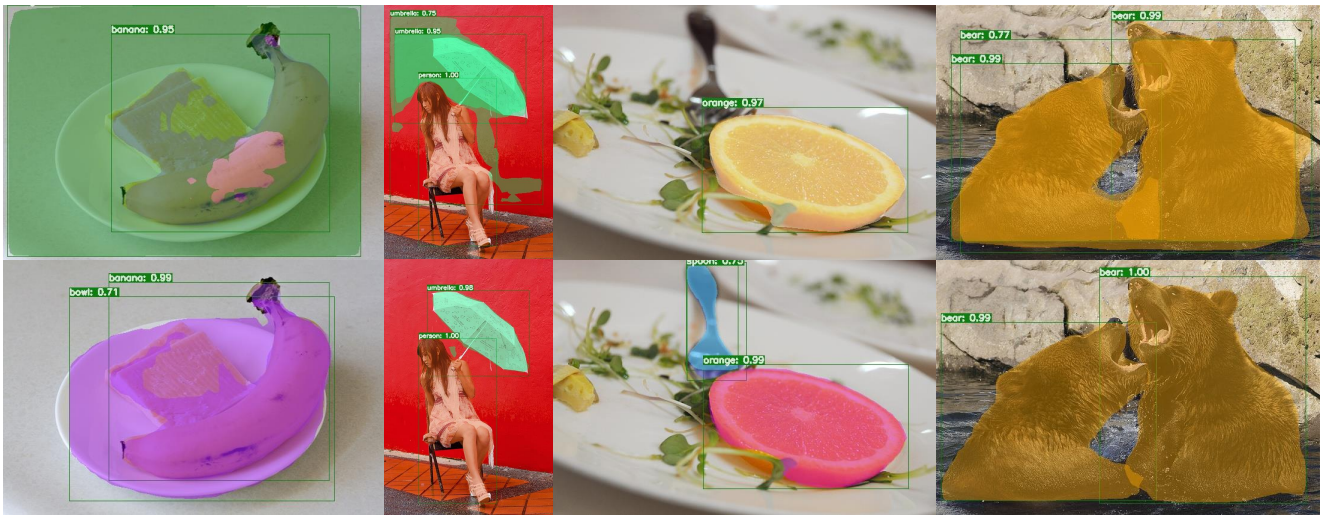


Figure 5. Mask R-CNN [14] (top) vs. Mask R-CNN with our overlap sampler (bottom) in ResNet-50-FPN [24, 15] backbone. The latter exhibits higher object detection box AP and instance segmentation mask AP on COCO `minival` [26].

4. Experiments

In this section, we present the experimental results of the overlap sampler on the COCO [26] dataset. Section 4.1 describes the training and evaluation configurations, as well as our implementation details for Faster R-CNN [32] and Mask R-CNN [14]. Then, we present ablation studies for our overlap sampler in Section 4.2. Finally, Section 4.3 compares the results of overlap sampler with other sampling heuristics. All of our models are implemented base on maskrcnn-benchmark [9].

4.1. Implementation Details

COCO Datasets. Following standard practice [2, 14, 24], we train the model on the COCO [26] `train2017`, and evaluate all ablations on COCO `minival`. To compare with other sampling heuristics, we also submit the detection results to the COCO `test-dev` evaluation server. As COCO applies average precision (AP) at different IoU values and sizes as the main evaluation metrics, we report the *COCO-style* AP metrics as the detection accuracy, including AP, AP_{50}, AP_{75}, and AP_S, AP_M, AP_L.

Faster R-CNN and Mask R-CNN with Overlap Sampler. To validate the effectiveness of the proposed overlap sampler, we incorporate it into the well-known Faster R-CNN [32] and Mask R-CNN [14]. Then we train them and evaluate their box AP and mask AP on COCO, respectively. We use ResNet-50-FPN [15, 24] as the backbone for ablation studies, while the heavier backbone [39] is also used to report the performance. For better coordinating the detectors and the sampler, we carefully tune the hyper-parameters, i.e., probability sampling strategy (*hard*, *soft*, *linear*), loss reweighting parameter (ϵ) and NMS threshold (θ). They will be further discussed in Section 4.2.

Other Hyper-parameters. To keep the consistency with maskrcnn-benchmark [9], we follow their configurations for training Faster R-CNN and Mask R-CNN. Specifically, we set the batch size as 16 with the weight decay of 0.0001 and momentum of 0.9 and set the initial learning rate as 2×10^{-2} in the first 60k iterations, then decay it by to 10 and 10^2 for training another 20k and 10k iterations, which is called "1$\times$" training schedule [11].

4.2. Ablation Study

Negative Overlap Sampler. As presented in Table 2(a), (b) and (c), we do experiments for negative overlap sampler to determine the optimal probability sampling strategy and hyper-parameters. We discuss them as follows.

• **Probability Sampling Strategy**: As illustrated in Section 3.2, there are several probability sampling strategies for the negative overlap sampler, including *hard*, *soft* and *linear* probability sampling. The *hard* sampling simply selects all negatives with $IoU^{p,n} >= \theta$, then randomly samples in $IoU^{p,n} < \theta$ negatives. Similarly, the *soft* sampling evenly splits negatives to K groups according to the interval of $IoU^{p,n}$, and assigns uniform probabilities for negatives in each group. The *linear* sampling directly assigns probability for each negative according to normalized $IoU^{p,n}$.

We compare their performance in Table 2(a). First, the *uniform* sampling means baseline model, which yields 36.8 AP on COCO `minival`. Among *hard*, *soft* and *linear* probability sampling strategies, it is shown that the *soft* sampling could achieve the highest 37.7 AP, which is 0.2 AP and 0.7 AP higher than *hard* sampling and *linear* sampling, respectively. It is worth noting that the performance of *linear* sampling is obvious lower than *hard* and *soft* sampling. In Section 3.2, We have discussed this problem that

(a) Sampling strategy in negative overlap sampler

Strategies	AP	AP_{50}	AP_{75}
uniform (baseline)	36.8	58.4	40.0
hard	37.5	58.6	**41.1**
soft	**37.7**	**58.9**	41.0
linear	37.0	58.4	40.2

(b) The number of bins in *soft* strategy

Number	AP	AP_{50}	AP_{75}
$K = 1$	36.8	58.4	40.0
$K = 2$	**37.7**	**58.9**	**41.0**
$K = 3$	**37.7**	**58.9**	**41.0**
$K = 4$	**37.7**	**58.9**	**41.0**

(c) Varying NMS threshold during inference

NMS	AP	AP_{50}	AP_{75}
$\theta = 0.45$	37.5	58.7	40.9
$\theta = 0.50$	37.7	**58.9**	41.0
$\theta = 0.55$	**37.9**	**58.9**	**41.5**
$\theta = 0.60$	37.8	58.7	41.5

(d) Sampling strategy in positive overlap sampler

Strategies	AP	AP_{50}	AP_{75}
uniform (baseline)	36.8	58.4	**40.0**
hard	37.1	58.9	39.9
soft	**37.2**	**59.0**	39.9
linear	37.0	58.7	39.9

(e) Varying loss reweighting factor

ϵ	AP	AP_{50}	AP_{75}
$\epsilon = 1$	37.2	**59.0**	39.9
$\epsilon = 2$	**37.4**	**59.0**	40.5
$\epsilon = 3$	**37.4**	58.9	40.5
$\epsilon = 5$	37.2	58.3	**40.6**

(f) Box AP and mask AP of various combinations.

Component	Settings			
Negative Overlap Sampler (*soft*, $K = 2, \theta = 0.55$)	✗	✓	✗	✓
Positive Overlap Sampler (*soft*, $\epsilon = 2$)	✗	✗	✓	✓
box AP	36.8	37.9 (+1.1)	37.4 (+0.6)	**38.3 (+1.5)**
box AP_{50}	58.4	58.9 (+0.5)	59.0 (+0.6)	**59.4 (+1.0)**
box AP_{75}	40.0	41.5 (+1.5)	40.5 (+0.5)	**41.9 (+1.9)**
mask AP	34.2	34.8 (+0.6)	34.5 (+0.3)	**35.0 (+0.8)**
mask AP_{50}	56.0	56.3 (+0.3)	56.5 (+0.5)	**56.7 (+0.7)**
mask AP_{75}	36.3	37.2 (+0.9)	36.5 (+0.2)	**37.4 (+1.1)**

Table 2. Ablation studies for our overlap sampler. If not specified, default values are: *soft* probability sampling strategy with $K = 2$, loss reweighting factor $\epsilon = 1$, NMS threshold $\theta = 0.5$. (a), (b), (c) show the ablations for the negative overlap sampler, which achieves 37.9 AP by *soft* sampling strategy with $K = 2$, $\theta = 0.55$. (d) and (e) are the ablations for the positive overlap sampler, which obtains 37.4 AP by *soft* sampling strategy with $\epsilon = 2$. Moreover, we also construct three variants for comparing their box AP and mask AP improvements in Table 2 (f). Note that these experiments are conducted with Faster R-CNN (box AP) [32] and Mask R-CNN (mask AP) [14] of ResNet-50-FPN [15, 24] backbone on COCO `minival` [26], which are implemented on maskrcnn-benchmark [9].

the *linear* sampling would not select the negatives with $IoU^{p,n} = 0$. However, these negatives occupy the main part of all negatives, which may be detrimental to learning negative examples. In conclusion, it suggests that the sampler can not fully ignore low $IoU^{p,n}$ negatives.

• **IoU Bins**: In *soft* probability sampling, we evenly split the IoU interval to K bins, and selects examples uniformly from the bins. The default value of K is 2, but we hope to find the most suitable K to improve our negative overlap sampler. Unfortunately, it is shown in the Table 2(b) that the performance of *soft* sampling is not sensitive to K.

• **NMS Threshold**: During inference, we tune the NMS threshold θ to find the optimal performance of the negative overlap sampler in Table 2(c). By setting $\theta = 0.55$, it achieves the best 37.9 AP on COCO `minival`.

Positive Overlap Sampler. Now we discuss the experiments for positive overlap sampler in Table 2(d) and (e).

• **Probability Sampling Strategy**: We use the same sampling strategies of negative overlap sampler for positive overlap sampler. Among them, the *soft* sampling still performs best, which yields 0.4 higher AP than baseline.

• **Loss Reweighting Factor**: The loss reweighting factor ϵ controls the loss weights of the positives with a maximum $IoU^{g,p}$ for each ground-truth, which has been discussed in Section 3.2. In Table 2(e), as ϵ increases, we can see the AP_{50} drops but AP_{75} keeps improving, which illustrates that increasing ϵ is beneficial to yield high-quality predictions. By setting $\epsilon = 2$, we could achieve the optimal 37.4 AP here, which improves the *soft* sampling by 0.2 AP.

Box AP and Mask AP of Various Combinations. The performances of negative overlap sampler and positive overlap sampler for Faster R-CNN, which have been presented in Table 2, are achieved 37.9 AP and 37.4 AP at most, respectively. When we combine them as shown in Table 2(f), we obtain an impressive 38.3 AP, which is 1.5 AP higher than baseline. To evaluate the generalization for our overlap sampler, we also train the Mask R-CNN with overlap sampler for instance segmentation task. It helps the Mask R-CNN to achieve 35.0 mask AP, which is 0.8 AP higher than the original model. What can be observed is that the improvements on Faster R-CNN and Mask R-CNN mainly come from AP_{75}, which indicates the sampler helps the model to yield higher-quality predictions.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster R-CNN [9, 32]	ResNet-50-FPN [15, 24]	37.2	59.3	40.3	21.3	39.5	46.9
OHEM* [33]		37.4	59.5	40.3	21.2	40.3	47.1
PISA [3]		37.8	58.0	41.7	22.1	40.8	46.6
Libra R-CNN [28]		**38.7**	59.9	42.0	22.5	**41.1**	48.7
Overlap Sampler		38.6	**60.2**	41.9	22.4	**41.1**	48.6
Faster R-CNN [9, 32]	ResNet-101-FPN [15, 24]	39.3	61.4	42.7	22.1	41.9	50.1
Libra R-CNN [28]		40.3	61.3	43.9	22.9	43.1	51.0
Overlap Sampler		**40.6**	**62.0**	**44.0**	**23.7**	**43.5**	**51.1**
Faster R-CNN [9, 32]	ResNext-101-FPN-32x4d [39, 24]	41.3	61.9	44.9	24.3	44.4	51.8
PISA [3]		41.5	61.8	45.8	24.7	44.7	51.9
Overlap Sampler		**42.5**	**62.4**	**46.2**	**25.1**	**45.0**	**52.3**

Note: The symbol "*" means our re-implemented results.

Table 3. Comparisons with existing sampling heuristics designed for Faster R-CNN on COCO `test-dev` (single model, without bells and whistles). Note, in the literature [28], the Libra R-CNN can achieve 41.1 AP with "2×" training schedules. As we use "1×" training schedule in all experiments, for a fair comparison, we only show the accuracy of Libra R-CNN with "1×" schedule. Furthermore, the ISR and Libra R-CNN do not report their result in ResNet-101-FPN and ResNext-101-FPN-32x4d, respectively. Therefore, we only report their COCO AP with the corresponding backbone.

4.3. Results on COCO Test Set.

To further validate the effectiveness of our overlap sampler, we train Faster R-CNN [32] with various backbones, and submit their results to COCO `test-dev` evaluation server, to compare it with other sampling heuristics. As shown in Table 3, with ResNet-50-FPN [15, 24] backbone, the Faster R-CNN (baseline) could achieve 37.2 AP. We also implement OHEM [33] for Faster R-CNN that selects the hard examples per image during training. However, this scheme has brought little improvement (37.4 AP vs. 37.2 AP). Then, we present the COCO AP results of Libra R-CNN [28] and ISR [3]. It is shown that they can get an obviously higher 37.8 AP and 38.7 AP, respectively.

Finally, we incorporate our overlap sampler into the Faster R-CNN, with the hyper-parameters determined in Section 4.2. In the fifth row of Table 3, we can see the detector yields 38.6 AP, which is 1.4 AP higher than the original Faster R-CNN. Unfortunately, it is slightly worse than Libra R-CNN [28]. This is because Libra R-CNN benefits from feature pyramid [28] and balanced L1 loss [28] that have been proven to be effective, which are not sampling heuristics. We believe our overlap sampler combined with these methods could achieve better performance. Nevertheless, with a large ResNet-101-FPN [15, 24] backbone, our overlap sampler achieves higher COCO AP results (40.6 AP) than Libra R-CNN (40.3 AP). It suggests that our method can still work even on a stronger baseline.

As PISA [3] does not report its performance in ResNet-101-FPN [15, 24], we also adopt ResNext-101-FPN [24, 39] to do the comparison. Table 3 shows that the Faster R-CNN with our overlap sampler achieves an impressive 42.5 AP, which is 1.0 AP and 1.2 AP higher than PISA and the baseline, respectively.

5. Conclusion

In this paper, we carefully investigate different sampling heuristics for region-based detectors, and discover the overlaps among examples is crucial for improving the IoU-based sampling heuristics. Motivated by the analysis, a novel *Overlap Sampler* is proposed, which samples according to IoU between examples themselves, rather than IoU between ground-truth and example. With the overlap sampler, we upgrade the two well-known region-based detectors Faster R-CNN and Mask R-CNN. Extensive experiments present that overlap sampler is more effective than the random sampling and IoU-balanced sampling, which yields 1.5 higher box AP and 0.8 higher mask AP for Faster R-CNN and Mask R-CNN on COCO `minival`, respectively. Given the performance of the proposed method that surpassing ISR [3] and Libra R-CNN [28] on COCO `test-dev`, we expect overlap sampler could be adopted in other region-based detectors.

References

[1] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-nms - improving object detection with one line of code. In *ICCV*, pages 5562–5570, 2017.

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

[3] Y. Cao, K. Chen, C. C. Loy, and D. Lin. Prime sample attention in object detection. *CoRR*, abs/1904.04821, 2019.

[4] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: object detection via region-based fully convolutional networks. In *NIPS*, pages 379–387, 2016.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[6] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, pages 6569–6578, 2019.

[7] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[8] P. F. Felzenszwalb, R. B. Girshick, and D. A. McAllester. Cascade object detection with deformable part models. In *CVPR*, pages 2241–2248, 2010.

[9] M. Francisco and G. Ross. maskrcnn-benchmark. github.com: facebookresearch/maskrcnn-benchmark, 2018.

[10] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, pages 3354–3361, 2012.

[11] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. Detectron. github.com: facebookresearch/detectron.

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

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

[14] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, 2017.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[16] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang. Bounding box regression with uncertainty for accurate object detection. In *CVPR*, pages 2888–2897, June 2019.

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

[18] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang. Acquisition of localization confidence for accurate object detection. In *ECCV*, pages 816–832, 2018.

[19] Kemal, B. C. Cam, S. Kalkan, and E. Akbas. Imbalance problems in object detection: A review. *CoRR*, abs/1909.00169, 2019.

[20] T. Kong, F. Sun, H. Liu, Y. Jiang, and J. Shi. Foveabox: Beyond anchor-based object detector. *CoRR*, abs/1904.03797, 2019.

[21] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 765–781, 2018.

[22] B. Li, Y. Liu, and X. Wang. Gradient harmonized single-stage detector. In *AAAI*, pages 8577–8584, 2019.

[23] Y. Li, Y. Chen, N. Wang, and Z. Zhang. Scale-aware trident networks for object detection. In *ICCV*, pages 6054–6063, 2019.

[24] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944, 2017.

[25] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, pages 2999–3007, 2017.

[26] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014.

[27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. In *ECCV*, pages 21–37, 2016.

[28] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin. Libra R-CNN: towards balanced learning for object detection. In *CVPR*, pages 821–830, 2019.

[29] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016.

[30] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. In *CVPR*, pages 6517–6525, 2017.

[31] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

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

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

[34] Z. Tian, C. Shen, H. Chen, and T. He. FCOS: fully convolutional one-stage object detection. In *ICCV*, pages 9627–9636, 2019.

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

[36] P. A. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[37] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin. Region proposal by guided anchoring. In *CVPR*, pages 2965–2974, 2019.

[38] Y. Xiang, W. Choi, Y. Lin, and S. Savarese. Subcategory-aware convolutional neural networks for object proposals and detection. In *WACV*, pages 924–933, 2017.

[39] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 5987–5995, 2017.

[40] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin. Reppoints: Point set representation for object detection. In *ICCV*, pages 9657–9666, 2019.

[41] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li. Single-shot refinement neural network for object detection. In *CVPR*, pages 4203–4212, 2018.

[42] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points. *CoRR*, abs/1904.07850, 2019.

[43] X. Zhou, J. Zhuo, and P. Krähenbühl. Bottom-up object detection by grouping extreme and center points. In *CVPR*, pages 850–859, 2019.

[44] C. Zhu, Y. He, and M. Savvides. Feature selective anchor-free module for single-shot object detection. In *CVPR*, pages 840–849, 2019.

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