Multiple Anchor Learning for Visual Object Detection

Wei Ke*[1], Tianliang Zhang*[2], Zeyi Huang[1], Qixaing Ye†[2], Jianzhuang Liu[3], and Dong Huang[1]

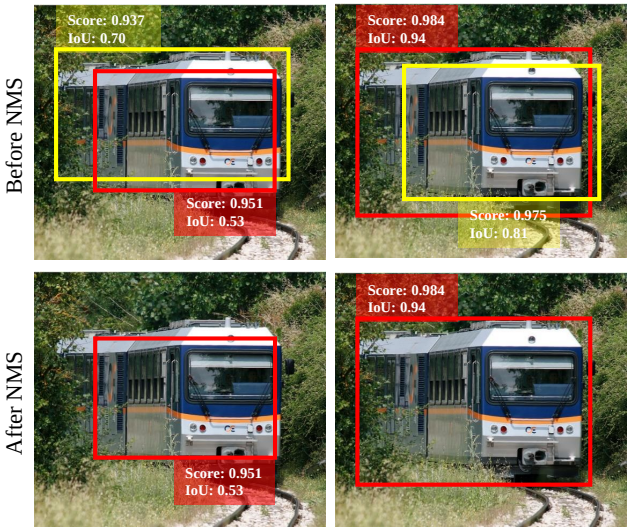
[2]Carnegie Mellon University , Pittsburgh, US
[2]University of Chinese Academy of Sciences, Beijing, China
[3]Shenzhen Institutes of Advanced Technology, Shenzhen, China
{weik, zeyih}@andrew.cmu.edu, zhangtianliang17@mails.ucas.ac.cn, jz.liu@siat.ac.cn, donghuang@cmu.edu

Abstract

Classification and localization are two pillars of visual object detectors. However, in CNN-based detectors, these two modules are usually optimized under a fixed set of candidate (or anchor) bounding boxes. This configuration significantly limits the possibility to jointly optimize classification and localization. In this paper, we propose a Multiple Instance Learning (MIL) approach that selects anchors and jointly optimizes the two modules of a CNN-based object detector. Our approach, referred to as Multiple Anchor Learning (MAL), constructs anchor bags and selects the most representative anchors from each bag. Such an iterative selection process is potentially NP-hard to optimize. To address this issue, we solve MAL by repetitively depressing the confidence of selected anchors by perturbing their corresponding features. In an adversarial selection-depression manner, MAL not only pursues optimal solutions but also fully leverages multiple anchors/features to learn a detection model. Experiments show that MAL improves the baseline RetinaNet with significant margins on the commonly used MS-COCO object detection benchmark and achieves new state-of-the-art detection performance compared with recent methods.

1. Introduction

Convolutional Neural Network (CNN) based object detectors have achieved unprecedented advances in the past few years [9, 8, 27, 25, 21, 17, 18]. In both recent two-stage and single-stage object detectors, the bounding box classification and localization modules are highly integrated: they are conducted on the shared local features, and are optimized over the sum of the loss functions.



(a) Baseline (b) MAL

Figure 1: Detection outputs of the baseline detector (RetinaNet) and the Multiple Anchor Learning (MAL), before and after NMS. The baseline detector may produce bounding boxes with high localization IoU with a low classification score (the yellow bbox), or low localization IoU with a high classification score (the red bbox), which lead to sub-optimal results after NMS. MAL produces bounding boxes with high co-occurrence of top classification and localization, leading to better detection results after NMS.

To provide rich candidates of shared local features, a prevalent approach is the introduce hand-crafted dense anchors [18] on the convolutional feature maps. These anchors create a uniform distribution of bounding box scales and aspect ratios, enabling objects with various scales and aspect ratios to be equally represented in training a detector.

However, optimization under a fixed set of hand-crafted

*indicates equal contributions
†indicates corresponding author

anchors significantly limits the possibility to jointly optimize classification and localization. During training, detectors leverage spatial alignment, $i.e.$, Intersection over Unit (IoU) between objects and anchors, as the sole criterion to assign anchors. Each assigned anchor independently supervises network learning for classification and localization. Without direct interactions of the two optimizations, the detections of accurate localization may have lower classification confidence, and be suppressed by the following Non-Maximum Suppression (NMS) procedure (see the baseline example in Fig. 1).

Recent remedy for the problem includes IoU-Net [13] and FreeAnchor [34]. However, it remains using independent classification and localization confidence during the training procedure. FreeAnchor selects anchors according to a joint probability over classification and localization. Nevertheless, the matching procedure based on maximum likelihood estimation (MLE) is not optimal considering the non-convexity of the problem.

In this paper, we present Multiple Anchor Learning (MAL), an automatic anchor learning approach that jointly optimizes object classification and localization from the perspective of anchor-object matching. In training phase of MAL, an anchor bag for each object is constructed by choosing the top ranked anchor with IoUs between anchors and the object bounding box. MAL evaluates positive anchors in each bag by combining their classification and localization scores. In each training iteration, MAL uses all positive anchors to optimize the training loss but selects the high/top-scored anchors as the solutions. This leads to high co-occurrence of top classification and localization (see the MAL example in Fig. 2).

MAL is optimized over an anchor selection loss based on Multiple Instance Learning (MIL) [23]. However, the iterative selection process under conventional MIL is potentially NP-hard to optimize. Selecting the top-scored instance (anchor) in each learning iteration could produce sub-optimal solutions, $e.g.$, falsely localized object parts. To address this issue, we solve MAL by repetitively depressing the confidence of top-scored anchors by perturbing their features, which guarantees that potential optimal solutions, $i.e.$, positive anchors of lower confidence, have an opportunity to participate in learning. By upgrading the supervision from independent anchors to multiple anchors, MAL fully leverages multiple anchors/features to learn a better detector. The contributions of this work include:

- We propose an Multiple Anchor Learning (MAL) approach, jointly optimizing classification and localization modules for object detection by evaluating and selecting anchors.
- We propose a selection-depression optimization strategy, providing an elegant-yet-effective way to prevent MAL from getting stuck into sub-optimal solutions during detector training.
- We improve state-of-the-arts with significant margins on the commonly used MS COCO dataset.

2. Related Work

Various taxonomies [20] have been used to categorize the large amount of CNN-based object detection methods, $e.g.$, one-stage [27] vs. two-stage [18], single-scale features [27] vs. multi-scale representation [17, 16, 24], and handcrafted architectures [21] vs. Network Architecture Search (NAS) [7]. In this paper, we review the related works from the perspective of object localization.

2.1. Anchor-Based Method

Training a detector requires to generate a set of bounding boxes along with their classification labels associated with the objects in an image. However, it is not trivial for CNNs to directly predict an order-less set of arbitrary cardinals [32]. One commonly used strategy is to introduce anchors, which employs a divide-and-conquer strategy to match objects with convolutional features, spatially.

Anchor Assignment. Anchor-based detection methods include the well-known Faster R-CNN [27], FPN [17], RetinaNet [18], SSD [21], DSSD [6], and YOLO [26]. In these detectors, a large amount of anchors are scattered over convolutional feature maps so that they can match objects of various aspect ratios and scales. During training, the anchors are assigned to objects (positive anchors) or backgrounds (negative anchors) by threshold their IoUs with the ground-truth bounding boxes [27]. During inference, anchors independently predict object bounding boxes, where the box with the highest classification score is retained after the NMS procedure.

Despite of the simplicity, these approaches rely on the assumption that anchors are optimal for both object classification and localization. For objects of partially occlusion and irregular shapes, however, such heuristics are implausible and they could miss the best anchors/features [34].

Anchor Optimization. To pursue optimal feature-object matching, MetaAnchor [32] learns to predict anchors from the arbitrary customized prior boxes with a subnet. GuidedAnchoring [31] leverages semantic features to guide the prediction of anchors while replacing dense anchors with predicted anchors. FreeAnchor [34] upgrades handcrafted anchor assignment to "free" anchor matching. This approach formulates detector training as a maximum likelihood estimation (MLE) procedure. Its goal is to learn features that best explain a class of objects in terms of both classification and localization. IoU-Net [13] selects anchors while predicting the IoU between a detected bounding box and a ground-truth box. Combined with an IoU-guided NMS, IoU-Net reduces the suppression failure

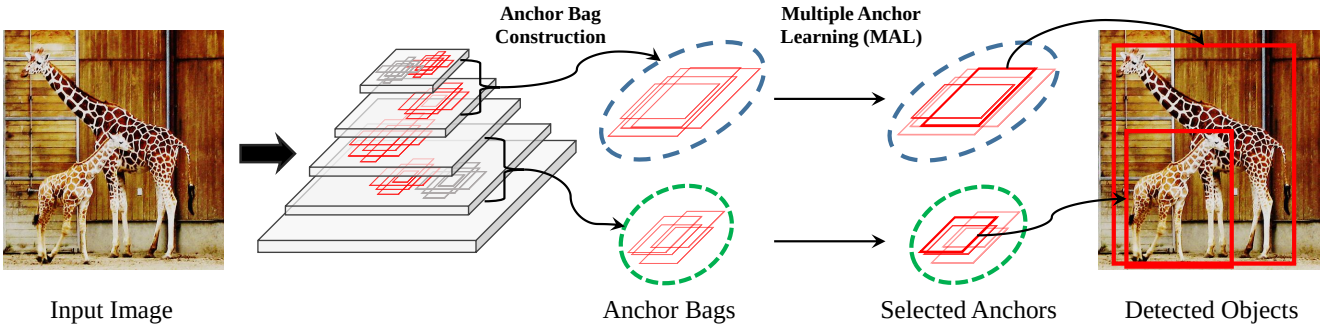


Figure 2: The main idea of MAL. In the feature pyramid network, an anchor bag A_i is constructed for each object b_i. Together with the network parameter learning, $i.e.$, back-propagation, MAL evaluates the joint classification and localization confidence of each anchor in A_i. Such confidence is used for anchor selection and indicates the importance of anchors during network parameter evolution.

caused by the misleading classification confidences. Gaussian YOLO [3] introduces localization uncertainty that indicates the reliability of anchors/bounding boxes. By using the estimated localization uncertainty during inference, this approach improves classification and localization accuracy.

All above approaches have taken some steps towards anchor learning. Nevertheless, how to efficiently select optimal anchors remains to be further elaborated. Considering a non-convex objective function which could cause sub-optimal solutions, we propose an adversarial selection-depression strategy to alleviate this issue.

2.2. Anchor-Free Method

Instead of using anchors as bases to conduct detection, researchers have recently explored anchor-free approaches, which operates on individual cells of the convolutional feature maps. FCOS leverages cell-level supervision and center-ness bounding-box regression [29] for object detection. CornerNet [15] and CenterNet [5] replace bounding box supervision with key-point supervision. Extreme point [35] and RepPoint [33] use point sets to predict object bounding boxes.

As a new direction for object detection, anchor-free methods show great potential for extreme object scales and aspect ratios, without constraints set by hand-craft anchors. However, without the anchor box as the reference point, direct regression of bounding boxes from convoltuional features remains a very challenging problem. As an anchor-based approach, MAL outperforms the current top anchor-free detectors such as CenterNet and CornerNet.

3. The Proposed Approach

MAL is implemented based on RetinaNet [18] network architecture. MAL upgrades RetinaNet by finding optimal selection of anchors/features for both classification and localization. In what follows, we briefly revisit RetinaNet on its original mechanism in object classification and localization. We then elaborate how MAL improve classification and localization by evaluating anchors. We finally propose an anchor selection-depression strategy to pursue optimal solutions of MAL.

3.1. RetinaNet Revisit

RetinaNet is a representative architecture of single-stage detectors with state-of-the-art performance. A RetinaNet detector is made up of a backbone network and two subnets, one for object classification and other for object localization. Feature Pyramid Network (FPN) is used at the end of RetinaNet backbone network. From each feature map in the feature pyramid, a classification subnet predicts category probabilities while a box regression subnet predicts object locations using anchor boxes as the reference locations. The input features of the two subnets are shared across the feature pyramid levels for efficiency. Considering the extreme imbalance of foreground-background classes, presented as positive-negative anchors after anchor-object matching, Focal Loss is adopted to prevent the vast number of easy negatives from overwhelming the detector during training.

Let $x \in \mathcal{X}$ be an input image with label $y \in \mathcal{Y}$, where $\mathcal{X}$ is the training image set and $\mathcal{Y}$ is the label set of the categories. Without loss of generality, denote B as the ground-truth bounding boxes of the objects in a positive image. $b_i \in B$ consists of the class label b_i^{cls} and the spatial position b_i^{loc}. The classification confidence a_j^{cls} and bounding-box output a_j^{loc} of the anchor a_j are predicted by the classification and the box regression subnets, respectively. The anchors in an image are divided into positive ones a_{j+} if their IoUs with the ground-truth boxes are larger than a threshold, and negative ones a_{j-} otherwise. Anchors are

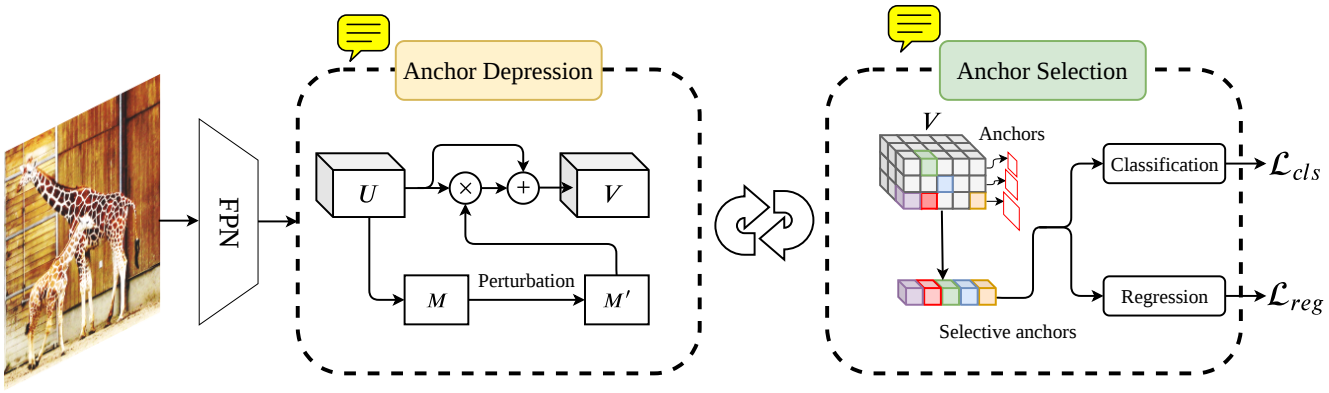


Figure 3: MAL implementation. During training, it includes the additional anchor selection and anchor depression modules added to RetinaNet. During test, it uses exactly the same architecture as RetinaNet. "U" and "V" respectively denote convolutional feature maps before and after depression. "M" and "M'" respectively denote an activation map before and after depression.

used to supervise network learning, as

$$\theta^* = \arg\max_{\theta} \left(f_\theta(a_{j+}, b_i^{cls}) - \gamma f_\theta(a_{j-}, b_i^{cls}) \right), \quad (1)$$

where $f_\theta(\cdot)$ denotes the classification procedure, and γ is a factor to balance the importance of negative/positive anchors. Simultaneously, positive anchors are used to optimize the object localization, as

$$\theta^* = \arg\max_{\theta} g_\theta(a_{j+}, b_i^{loc}), \quad (2)$$

where θ denotes the network parameters, and $g_\theta(\cdot)$ denotes the bounding-box regression procedure. Eq. 1 and Eq. 2 are actually implemented by minimizing the Focal Loss, $\mathcal{L}_{cls}(a_j, b_i^{cls})$, and the Smooth-L1 loss, $\mathcal{L}_{loc}(a_j, b_i^{loc})$,

During network learning, each assigned anchor independently supervises the learning for object classification and object localization, without considering whether the detection and localization are compatible on assigned anchors. This could cause the anchors of accurate localization but with lower classification confidence to be suppressed by the following Non-Maximum Suppression (NMS) procedure.

3.2. Multiple Anchor Learning

To alleviate the drawbacks of independent anchor optimization, we propose the Multiple Anchor Learning (MAL) approach, Fig. 2. In each learning iteration, MAL selects high-scored instances in an anchor bag to update the model. After updating, the model evaluates each instance with new confidence. Model learning and anchor selection iteratively perform towards final optimization.

To fulfill this purpose, we first construct an anchor bag A_i for the i^{th} object. The anchor bag includes the top-k anchors according to the IoUs between the anchors and the ground truth. Together with network parameter learning, $i.e.$, back-propagation, MAL evaluates the joint classification and localization confidence of each anchor in A_i. Such confidence is used for anchor selection and indicates the importance of anchors during network parameter evolution. For simplicity, consider solely the learning upon positive anchors, while that for negative anchors follows Eq. 1. MAL has the following objective function:

$$\begin{aligned} \{\theta^*, a_i^*\} &= \arg\max\nolimits_{\theta, a_j \in A_i} F_\theta(a_j, b_i) \\ &= \arg\max\nolimits_{\theta, a_j \in A_i} f_\theta(a_j, b_i^{cls}) + \beta g_\theta(a_j, b_i^{loc}), \end{aligned} \quad (3)$$

where $f_\theta(.)$ and $g_\theta(.)$ give the classification and localization scores, respectively, and β is a regularization factor. It is towards selecting a best positive anchor a_i^* for the i^{th} object, as well as learning the network parameters θ^*.

The objective function defined in Eq. 3 is converted to a loss function as:

$$\begin{aligned} \{\theta^*, a_i^*\} &= \arg\min\nolimits_{\theta, a_j \in A_i} \mathcal{L}_{det}(a_j, b_i) \\ &= \arg\min\nolimits_{\theta, a_j \in A_i} \mathcal{L}_{cls}(a_j, b_i^{cls}) + \beta \mathcal{L}_{reg}(a_j, b_i^{loc}), \end{aligned} \quad (4)$$

where $\mathcal{L}_{cls}$ and $\mathcal{L}_{reg}$ are the classification and detection losses, respectively, as defined in Section 3.1. The loss for negative anchors follows the Focal Loss defined in [18].

3.3. Selection-Depression Optimization

Optimizing Eq. 3 or Eq. 4 with Stochastic Gradient Descent (SGD) is a non-convex problem, which could cause a sub-optimal anchor selection. To alleviate the problem and select optimal anchors, we propose repetitively depressing the confidence of selected anchors by perturbing their corresponding features. Such a learning strategy, referred to as

selection-depression optimization, solves the MAL problem in an adversarial manner.

Anchor Selection. According to $F_\theta(a_j, b_i)$, the conventional MIL algorithm tends to select the top-scored anchor. Nevertheless, in the context of object detection, selecting a top-scored anchor from each bag is difficult, as validated by the continuation MIL method [30]. Instead of selecting the highest-scored anchor in Eq. 3 in the training phase, we propose an "All-to-Top-1" anchor selection strategy from each anchor bag for back-propagation. When learning proceeds, we linearly decrease the number from $|A_i|$ (number of anchors in a bag) to 1. Formally, let $\lambda = t/T$, where t and T are the current and total numbers of iterations for training. Then let $\phi(\lambda)$ indicate the indices of high-ranked anchors and $|\phi(\lambda)| = |A_i| * (1 - \lambda) + 1$. Finally, Eq. 3 is re-written as:

$$\{\theta^*, a_i^*\} = \arg\max_{\theta, a_j \in A_i} \sum_{j \in \phi(\lambda)} F_\theta(a_j, b_i), \quad (5)$$

Along this pipeline, MAL leverages multiple anchors/features within the object region to learn a detection model in early training epochs, and converges to use a single optimal anchor at the last epoch.

Anchor Depression. Inspired by the inverted attention network [12], we developed an anchor depression procedure to perturb the features of selected anchors in order to decrease their confidences (see more Fig. 3). The rational is to endow unselected anchors with extra chances to participate the training. Formally, we denote the feature map and the attention map as U and M, where M is computed as $M = \sum_l w_l * U_l$, with w being the global average pooling of U and l being the channel index of U. We then generate a new depressed attention map $M' = (1 - \mathbb{1}_P) * M$ by cutting down the high values to zero, where $\mathbb{1}$ is the 0-1 indicator function. and P is the high-value position. The feature map is perturbed as:

$$V = (\mathbf{1} + M') \circ U_l, \quad (6)$$

where $\mathbf{1}$ is the identity matrix and $\circ$ denotes the element-wise multiplication. With the continuation strategy, the depression in Eq. 6 is reformulated as:

$$V = (\mathbf{1} + (1 - \mathbb{1}_{\psi(\lambda)}) * M) \circ U_l, \quad (7)$$

where $\psi(\lambda)$ indicates how many pixels to be perturbed.

3.4. Implementation

The implementation of an MAL detector is based on the RetinaNet detector where the features of the input image are extracted by a FPN backbone [17]. The anchor generation settings are the same as those of RetinaNet, *i.e.*, 9 anchors with three sizes $\{2^0, 2^{1/3}, 2^{2/3}\}$ and three aspect ratios $\{1:2, 1:1, 2:1\}$ for each pixel on the feature maps. Across the levels, the anchors cover the scale range from 32 to 813 pixels with respect to the input image.

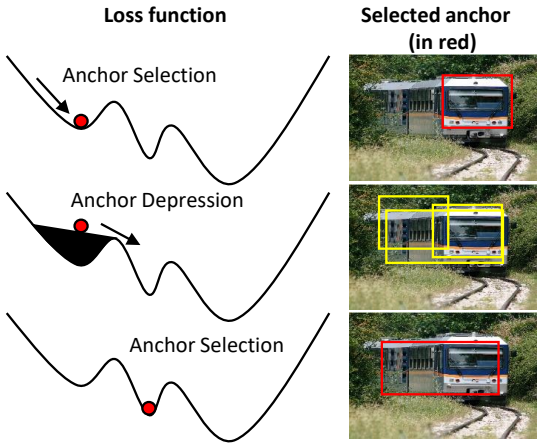


Figure 4: Optimization analysis. In the first curve, MAL selects a sub-optimal anchor and gets stuck into a local minimum. In the second curve, anchor depression increases the loss so that MAL continues the optimization. In this way, MAL has a greater chance to find optimal solutions.

During the feed-forward procedure of the network training, we calculate the detection confidence of each anchor, $F_\theta(a_j, b_i)$, to minimize the detection loss defined in Eq. 4. According to the confidence, top-k anchors are selected. The network parameters are then updated under the supervision of the selected anchors. After anchor selection, anchor depression is carried out, as described in Section 3.3. In the next iteration, anchor selection is carried out again to select high-scored anchors.

The inference procedure of our approach is exactly the same as RetinaNet, *i.e.*, we use the learned network parameters to predict classification scores and object bounding boxes, which are fed to a NMS procedure for object detection. As MAL is only applied in the detector training procedure to learn more representative features, our learned detector achieves performance improvement with negligible additional computation cost.

3.5. Optimization Analysis

The anchor selection-depression strategy approximates an adversarial procedure. The selection operation finds top-scored anchors that minimize the detection loss $\mathcal{L}_{det}$. The depression operation perturbs the corresponding features of selected anchors so that their confidence decreases and the detection loss increases again. The selection-depression strategy helps the learner find better solutions for the non-convex objective function of MAL. As illustrated by the first

curve of Fig. 4, MAL selects a sub-optimal anchor and gets stuck into a local minimum of the loss function. In the second curve, the anchor depression increases the loss, so that the local minimum is "filled". Consequently, MAL continues to find the next local minimum. After learning converges, MAL has a better chance to find optimal solutions.

4. Experiments

In this section, we present experimental results of the proposed Multiple Anchor Learning approach on the bounding-box detection track of the challenging COCO benchmark [19]. We follow the common practice and use ∼118k images for training, 5k for validation and ∼20k for testing without provided annotations (test-dev). AP is computed over ten different IoU thresholds, *i.e.*, 0.5: 0.05: 0.95, with all categories. It is the commonly used evaluation metric for object detection.

4.1. Experimental Setting

We utilize ResNet-50, ResNet-101, and ResNeXt-101 with FPN as backbones. The batch normalization layers are fixed to be frozen in the training phase. We use a mini-batch of 2 images per GPU, thus making a total mini-batch of 16 images on 8 GPUs. The initial learning rate is set to 0.01 and decreased by a factor of 10 after 90k and 120k for the 135k setting (ResNet-50), and 120k and 160k for the 180k setting (ResNet-101 and ResNeXt-101). The synchronized Stochastic Gradient Descent (SGD) is adopted for network optimization. The weight decay of 0.0001 and the momentum of 0.9 are used. A linear warmup strategy is adopted in the first 500 iterations. We set the regularization factor $\beta = 0.75$ experimentally. Following [34], we assign anchors to ground-truth using IoU threshold of 0.5, and to background if their IoUs are in [0, 0.4).

4.2. Ablation Study

For ablation study, we used ResNet-50 as the backbone. All detection performances were evaluated on the COCO-minval dataset ($5k$ images). Firstly, we visualize the effectiveness of MAL in Fig. 5 on feature activation maps. Comparing MAL with RetinaNet, MAL activates more parts on the object and suppresses the more parts in the background. It demonstrates that MAL improved features for better object detection.

Anchor Selection: Without the depression component of MAL, we evaluate the selection component individually first. We compare the results of different k for anchor bag construction, as shown in Table 1a. The AP is stable when $k = 40$, 50, or 60. We choose 50 anchors in the following experiments. The results of different anchor selection strategies are shown in Table 1b. It improves AP from 35.46% to 38.14% when anchor bags are used instead of the scattered anchors in RetinaNet, as MAL+S(all) in Table 1b. In the RetinaNet, if an anchor is with good localization but without the highest score, it does not affect the network parameters. While using the anchor bags, this kind of anchor has potential to be selected for detector learning. By the continuation optimization which selects all anchors at the beginning and gradually reduces the selected anchors to the top-1, the performance is further improved to 38.39%, as MAL+S(all-top1) in Table 1b. It verifies that continuation optimization is also efficient in MAL.

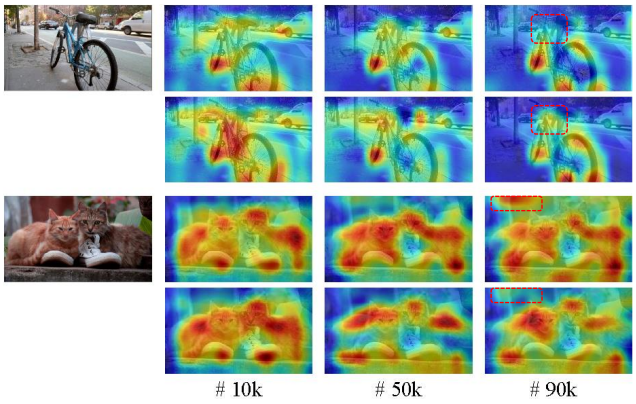


Figure 5: The activation map comparison between RetinaNet (the first and third rows) and MAL (the second and fourth rows). The attention maps at the 10k, 50k and 90k iterations are overlaid on input images. As highlighted by red boxes at the 90k th iteration, MAL gets better attention maps which activate more parts in the bicycle image and suppress irrelevant parts in the cat image.

Anchor Depression: We only add the depression component to RetinaNet to find the preferable indicator function $\psi(\lambda)$. We employ three kinds of indicator function. The first one is the constant function, which means keeping the same depression ratio in the whole training phase. We depress the top 50% pixels in the attention map. The AP decreases a little from 35.46% to 35.25%, as MAL+D(constant) shown in Table 1c. The reason is that at the beginning of the training phase, the parameters of the network are randomly initialized, and the depression is meaningless for the adversarial learning. If a step function is utilized for $\psi(\lambda)$, which increases the depression part from 0.0% to 50.0% by step, the performance is increased to 35.88%, as MAL+D(step) shown in Table 1c. It illustrates that the detector should be optimized in a way before depression. The third one is the symmetric step function, which increases the depression part from 0.0% to 50% and then decreases it from 50% to 0.0%. It achieves the best performance of 36.18%, as MAL+D(symmetric step) shown in Table 1c.

Selection-Depression: The efficient combination of selection and depression is shown in Fig. 6. We compare the

Method	AP	AP_{50}	AP_{75}
MAL(k=40)	38.27	56.67	40.81
MAL(k=50)	**38.39**	**56.81**	**41.14**
MAL(k=60)	38.08	56.11	40.18

(a) Detection performance upon different anchor numbers k in each anchor bag.

Method	AP	AP_{50}	AP_{75}
RetinaNet	35.46	51.61	39.37
MAL+S(all)	38.14	56.81	40.81
MAL+S(all-top1)	**38.39**	**56.81**	**41.14**

(b) Anchor selection strategy $\phi(\lambda)$. "S" denotes "Selection". We compare selecting all instances and all-top1 instance.

Method	AP	AP_{50}	AP_{75}
MAL+D(constant)	35.25	51.72	38.92
MAL+D(step)	35.88	52.34	39.63
MAL+D(symmetric step)	**36.18**	**52.66**	**39.88**

(c) Depression strategy $\psi(\lambda)$. "D" denotes "Depression". The constant function, step function, and symmetric step function are compared.

Table 1: Ablation study on the COCO minval dataset with the backbone ResNet50. We show the AP, AP_{50}, and AP_{75} (%).

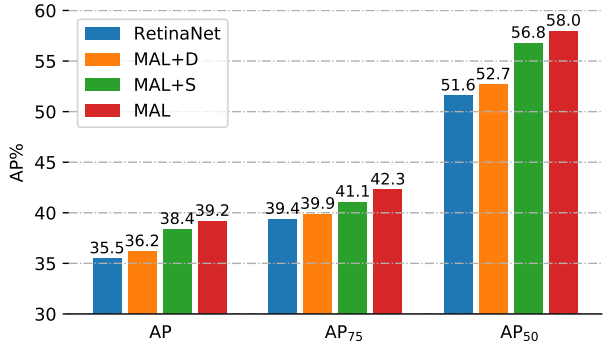
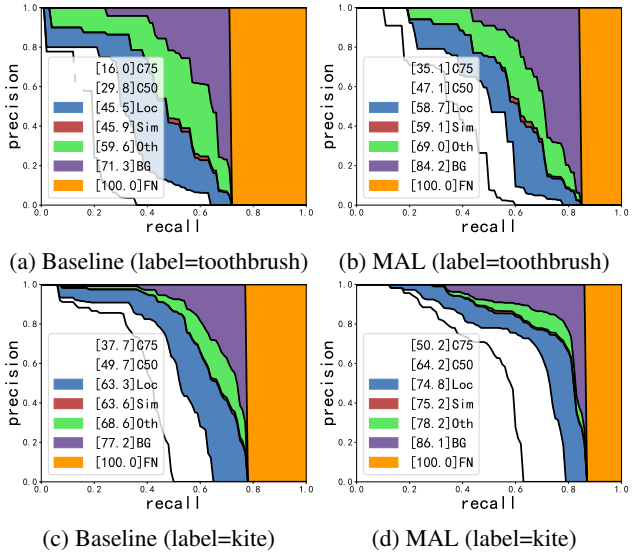


Figure 6: Ablation studies of the anchor selection and depression modules on the COCO-minval dataset. On the metrics AP, AP_{75} and AP_{50}, MAL outperforms the baseline detector (RetinaNet) with significant margins. "S" and "D" respectively denote "Selection" and "Detection".



(a) Baseline (label=toothbrush) (b) MAL (label=toothbrush)

(c) Baseline (label=kite) (d) MAL (label=kite)

Figure 7: Quantitative evaluation of detection performance. Top row: performance comparison on toothbrush detection. Bottom row: performance summary for the kit category.

AP, AP_{75}, and AP_{50}. The AP is increased to 36.2% with the depression component and to 38.4% with the selection component. When the adversarial manner is taken between selection and the depression, the AP is further improved to 39.2%, which is 3.7% (35.5% vs. 39.2%) performance gain compared with the original RetinaNet. The AP_{75} and AP_{50} have the same trend of growth as the AP.

Localization Improvement: In Fig. 7, we show an error factor analysis [2] of the localization results. It can be seen that poor localization (Loc) hinders the improvement of detection performance for objects of irregular shapes, *i.e.*, tilted and slender objects. Compared with the baseline method, MAL significantly reduces the localization error (blue part in Fig. 7) of these objects. For instance, the area under curve (AUC) decreases from 15.7% (45.5%−29.8%) to 11.6% (58.7%−47.1%) for the toothbrush category and from 13.6% (63.3%−49.7%) to 10.6% (74.8%−64.2%) for the kite category.

4.3. Comparison with State-of-the-Art Detectors

Keeping the best setting in the ablation study, we compare the propsoed MAL with the baseline, *i.e.*, RetinaNet, in Table 2. For ResNet-50, MAL improves the baseline from 35.5% to 39.2% with 3.7% improvement. For ResNet-101 and ResNeXt-101, the improvements are 4.5% and 4.1%, respectively. It illustrates that MAL achieves reliable gains with various of backbones.

Method	Backbone	AP	AP_{50}	AP_{75}
RetinaNet [18]	ResNet-50	35.5	51.6	39.4
MAL (ours)	ResNet-50	**39.2**	**58.0**	**42.3**
RetinaNet [18]	ResNet-101	39.1	59.1	42.3
MAL (ours)	ResNet-101	**43.6**	**62.8**	**47.1**
RetinaNet [18]	ResNeXt-101	40.8	61.1	44.1
MAL (ours)	ResNeXt-101	**45.9**	**65.4**	**49.7**

Table 2: Performance comparison with the baseline method (single-scale results) on the MS-COCO test-dev dataset. MAL improves the baseline with significant margins.

In Table 3, MAL is compared with the state-of-the-art detectors of two-stage methods and one-stage methods on the MS COCO test dataset, which are arranged in the increasing order of AP. For fair comparison, we re-scale the images such that their shorter sides are 800 pixels and the

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [11]	ResNet-101	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [17]	ResNet-101	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN w TDM [28]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	52.1
Deformable R-FCN [4]	Inception-ResNet-v2	37.5	58.0	40.8	19.4	40.1	52.5
Mask R-CNN [10]	ResNeXt-101	39.8	62.3	43.4	22.1	43.2	51.2
IoU-Net [13]	ResNet-101	40.6	59.0	-	-	-	-
Cascade RCNN [1]	ResNet-101	42.8	62.1	46.3	23.7	45.5	55.2
Grid R-CNN w/ FPN [22]	ResNeXt-101	43.2	63.0	46.6	25.1	46.5	55.2
One-stage methods							
YOLOv2 [25]	DarkNet-19	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [21]	ResNet-101	31.2	50.4	33.3	10.2	34.5	49.8
YOLOv3 [26]	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9
DSSD513 [21]	ResNet-101	33.2	53.3	35.2	13.0	35.4	51.1
GA-RetinaNet [31]	ResNet-50	37.1	56.9	40.0	20.1	40.1	48.0
MetaAnchor [32]	ResNet-50	37.9	-	-	-	-	-
RetinaNet [18]	ResNet101	39.1	59.1	42.3	21.8	42.7	50.2
CornerNet [15]	Hourglass-104	40.6	56.4	43.2	19.1	42.8	54.3
RetinaNet [18]	ResNeXt-101	40.8	61.1	44.1	24.1	44.2	51.2
FCOS [29]	ResNet-101	41.5	60.7	45.0	24.4	44.8	51.6
FoveaBox [14]	ResNeXt-101	42.1	61.9	45.2	24.9	46.8	55.6
AB+FSAF [36]	ResNeXt-101	42.9	63.8	46.3	26.6	46.2	52.7
FreeAnchor [34]	ResNeXt-101	44.8	64.3	48.4	27.0	47.9	56.0
CenterNet [5]	Hourglass-104	44.9	62.4	48.1	25.6	47.4	57.4
ours							
MAL	ResNet-101	43.6	62.8	47.1	25.0	46.9	55.8
MAL	ResNeXt-101	45.9	65.4	49.7	27.8	49.1	57.8
MAL (multi-scale)	ResNet-101	45.0	63.7	48.9	28.0	48.0	57.0
MAL (multi-scale)	ResNeXt-101	**47.0**	**66.1**	**51.2**	**30.2**	**50.1**	**58.9**

Table 3: Performance comparison with the state-of-the-art methods on the MS-COCO test-dev dataset (single-scale results unless explicitly stated). MAL achieves new state-of-the-art performance. As a one-stage detector, MAL also outperforms most two-stage detectors.

longer sides not more than 1333 pixels.

For one-stage methods, we compare the state-of-the-art including YOLO [25, 26], SSD [21], FCOS [29], FreeAnchor [34] and CenterNet [5]. With the ResNet-101 backbone, MAL achieves 43.6% AP of single-scale, which outperforms the anchor-free approach FCOS [29] by 2.1% (43.6% vs. 41.5%). With the ResNeXt-101 backbone, MAL achieves 45.9% AP of single scale, which achieves 1.1% (45.9% vs. 44.8%) gain compared with the recent FreeAnchor [34]. It also outperforms state-of-the-art CenterNet [5] by 1.0% AP (45.9% vs. 44.9%). Note that CenterNet uses the Hourglass-104 backbone which has much more network parameters than ResNeXt-101. These are significant margins for the challenging object detection task. The multi-scale testing APs of MAL are further improved to 45.0% and 47.0% with ResNet-101 and ResNeXt101, respectively.

Table 3 also compares MAL with representative two-stage detectors including Faster-RCNN with FPN [17], Mask R-CNN [10], IoU-Net [13], and Grid R-CNN [22]. MAL outperforms most two-stage detectors. Particularly, it outperforms the recent Grid R-CNN detector by 2.7% (45.9 vs. 43.2%) with the same backbone. As a one-stage detector with simpler implementation, MAL shows great potential to surpass two-stage detectors.

5. Conclusion

We have proposed an elegant and effective training approach, referred to as Multiple Anchor Learning (MAL), for visual object detection. By selecting anchors to jointly optimize bounding box classification and localization, MAL upgrades the standard hand-crafted anchor assignment mechanism to a learnable object-anchor matching mechanism. We proposed a simple selection-depression strategy to alleviate the sub-optimization issue of MAL. MAL improved object detection with significant margins compared with the

baseline detector RetinaNet, achieves the best result on MS-COCO among single-stage methods, and outperforms many recent two-stage approaches. Such improvements root in not only the optimal selection of anchors but also implicit feature assembling based on a bag of anchors. Our work presents a promising direction to relax anchor design in learning a visual object detection.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *IEEE CVPR*, pages 6154–6162, 2018. 8

[2] 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. *arXiv:1906.07155*, 2019. 7

[3] Jiwoong Choi, Dayoung Chun, Hyun Kim, and Hyuk-Jae Lee. Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving. In *IEEE ICCV*, pages 502–511, 2019. 3

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *IEEE ICCV*, page 764773, 2017. 8

[5] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Object detection with keypoint triplets. In *IEEE CVPR*, 2019. 3, 8

[6] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C. Berg. DSSD: Deconvolutional single shot detector. In *arXiv preprint arXiv:1701.06659*. 2

[7] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V. Le. NAS-FPN: learning scalable feature pyramid architecture for object detection. In *IEEE CVPR*, pages 7036–7045, 2019. 2

[8] Ross B. Girshick. Fast R-CNN. In *IEEE ICCV*, pages 1440–1448, 2015. 1

[9] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE CVPR*, pages 580–587, 2014. 1

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *IEEE ICCV*, pages 2980–2988, 2017. 8

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE CVPR*, pages 770–778, 2016. 8

[12] Zeyi Huang, Wei Ke, and Dong Huang. Improving object detection with inverted attention. *arXiv:1903.12255*, 2019. 5

[13] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *ECCV*, pages 784–799, 2018. 2, 8

[14] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, and Jianbo Shi. Foveabox: Beyond anchor-based object detector. *arXiv:1904.03797*, 2019. 8

[15] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 765–781, 2018. 3, 8

[16] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *IEEE ICCV*, pages 502–511, 2019. 2

[17] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *IEEE CVPR*, pages 936–944, 2017. 1, 2, 5, 8

[18] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE ICCV*, pages 2999–3007, 2017. 1, 2, 3, 4, 7, 8

[19] 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*, 2014. 6

[20] Li Liu, Wanli Ouyang, XiaogangWang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikainen. Deep learning for generic object detection: A survey. *Int. J. Comp. Vis.*, 2019. 2

[21] 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. 1, 2, 8

[22] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid R-CNN. In *IEEE CVPR*, 2019. 8

[23] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In *NeurIPS*, pages 570–576, 1997. 2

[24] Junran Peng, Ming Sun, Zhaoxiang Zhang, Tieniu Tan, and Junjie Yan. Pod: Practical object detection with scale-sensitive network. In *IEEE ICCV*, pages 6054–6063, 2019. 2

[25] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE CVPR*, pages 779–788, 2016. 1, 8

[26] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *IEEE CVPR*, pages 6517–6525, 2017. 2, 8

[27] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015. 1, 2

[28] Abhinav Shrivastava, Rahul Sukthankar, Jitendra Malik, and Abhinav Gupta. Beyond skip connections: Top-down modulation for object detection. *arXiv:1612.06851*, 2016. 8

[29] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. *arXiv:1904.01355*, 2019. 3, 8

[30] Fang Wan, Chang Liu, Wei Ke, Xiangyang Ji, Jianbin Jiao, and Qixiang Ye. C-MIL: continuation multiple instance learning for weakly supervised object detection. In *IEEE CVPR*, pages 2199–2208, 2019. 5

[31] Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. Region proposal by guided anchoring. In *IEEE CVPR*, pages 2965–2974, 2019. 2, 8

[32] Tong Yang, Xiangyu Zhang, Zeming Li, Wenqiang Zhang, and Jian Sun. Metaanchor: Learning to detect objects with customized anchors. In *NeurIPS*, pages 320–330, 2018. 2, 8

[33] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *IEEE ICCV*, pages 502–511, 2019. 3

[34] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. FreeAnchor: Learning to match anchors for visual object detection. In *NeurIPS*, 2019. 2, 6, 8

[35] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krähenbühl. Bottom-up object detection by grouping extreme and center points. In *IEEE CVPR*, pages 850–859, 2019. 3

[36] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In *IEEE CVPR*, pages 840–849, 2019. 8