# A Triple-head Network with Loss-aware Label Assignment for Object Detection

Wenjie Lin[1], Jun Chu[1*], Lu Leng[1*], Xingbo Dong[2]

[1]Key Laboratory of Jiangxi Province for Image Processing and Pattern Recognition, Nanchang Hangkong University,Nanchang, 330063, China.
[2]School of Electrical and Electronic Engineering, Yonsei University, Seoul, 03722, Republic of Korea.

*Corresponding author(s). E-mail(s): chuj@nchu.edu.cn; leng@nchu.edu.cn;

**Abstract**

Label assignment strategy plays a vital role in supervised object detection systems. However, the handcrafted label assignment used in most previous studies is usually suboptimal because the assignment strategy is not adaptive to the training objective. Considering the limitations of handcrafted label assignments, we propose a novel object detection framework consisting of a triple-head network and a joint-loss-aware label assignment strategy. The foregrounds and backgrounds are first determined based on a ground truth (GT) box. A label assignment strategy using classification and regression joint loss as sampling criteria was designed and used for positive and negative sampling. In addition, ambiguous samples from overlapping areas of multiple GT boxes were sampled based on a loss-weighting mechanism. In addition to the conventional classification and regression branches in FCOS, a pre-regression branch, which serves as a location prior to the other two branches, was introduced to the system to form a triple-head detection network. Extensive experiments showed that the proposed method can effectively improve the detection performance for small and medium objects and achieve comparable performance in terms of average precision on the MS COCO test-Dev2017 datasets compared with existing state-of-the-art methods.

**Keywords:** Objection detection, label assignment, triple-head network

# 1 Introduction

Object detection is a longstanding, fundamental, and challenging task in computer vision. Object detection seeks to locate object instances in natural images and identify objects from specific categories, such as dogs or cats. Object detection supports a broad range of applications including remote sensing [1], video object tracking [2, 3], action recognition [4, 5], etc. In recent years, deep neural networks (DNN) [6–8] have been widely used in object detection tasks. These emerging learning-based object detectors have become the most prevalent algorithms in many real-world applications [9–12].

Many optimization directions have been investigated to boost DNN object detection performance, including network architecture, loss functions, and label assignment strategies. Although many loss functions and optimization strategies have been proposed [3, 13, 14], label assignment, which aims to assign positive or negative anchors from feature maps during training, remains a bottleneck for current object detectors [15].

Current object detectors can be categorized as using anchor-based or anchor-free approaches. Anchor-based object approaches typically assign anchor boxes as positive or negative based on the intersection-over-union (IoU) between anchor boxes and ground-truth (GT) boxes [11, 12, 16, 17]. However, the IoU only considers the regression results, and the IoU threshold requires careful tuning and is sensitive to performance; thus, use of the IoU metric is suboptimal for object detection tasks. An ideal strategy is to consider classification and regression tasks simultaneously.

In contrast, label assignment strategies based on grid cells [18], point sets [19], and spatial scale constraints [20] have been proposed for anchor-free object detectors. These strategies usually use metrics that are independent of the training goal, such as the center distance between the samples and the GT object. In addition, in dense scenes, the spatial-scale-constrained label assignment strategy directly assigns samples (ambiguous samples) from the overlapping areas of multiple GT boxes to objects with the smallest areas, resulting in intractable ambiguity.

Research on dynamic label assignment strategies [21–25] has recently indicated better performance than existing handcrafted label assignment. Zhu et al. [22, 23] proposed a feature-layer selection label assignment strategy to label samples with the lowest loss values as positive samples. By structuring detector training as a maximum likelihood estimation (MLE) procedure, Zhang et al. [24] substituted handcrafted label assignments with "free" anchor matching. The top K candidate samples were selected for each GT object using the IoU metric; samples with the highest detection likelihood were labeled as positive samples. Accordingly, Ke et al. [25] proposed constructing anchor bags and selecting anchors from each bag. This task was solved by repeatedly depressing the confidence of the selected anchors by perturbing their corresponding features. In an adversarial selection–depression condition, the proposed scheme pursues optimal solutions and fully leverages multiple anchors/features to learn a detection model. [26, 27] proposed a dynamic assignment strategy that resampled top K samples as positives. Such a dynamic label assignment strategy is more in line with the actual network training state than handcrafted label assignment strategy; thus, it can significantly enhance detection performance.

However, the methods proposed in [22, 23] determine only positive samples on a single scale. [26] chooses the fixed $K$ positive samples and fails to fully utilize all the samples. [24] and [25, 27] use only IoU as the metric for sampling; therefore, they are usually suboptimal and suffer from objects with significant shape variations.

Motivated by this previous research, we propose a novel object detection framework consisting of a triple-head network and a joint loss-aware (JLA) label assignment strategy. The JLA assignment strategy consists of foreground and background segmentation and positive and negative sampling using the joint-loss value as the sampling criteria. In addition, ambiguous samples from overlapping GT boxes are weighted by the current joint loss value and assigned to GT instances. Inspired by recent detection network research [19, 28] , a triple-head network consisting of pre-regression, classification, and regression branches was designed to further boost performance.

The contributions of this work are as follows:

1. A novel triple-head network was designed for object detection. The triple-head network has a pre-regression branch that serves as a location prior to the other two branches, further boosting detection performance.
2. A joint loss-aware label assignment strategy was designed based on [26, 27]. The label assignment strategy was based on both classification and regression losses; thus, it was consistent with the final training goal.
3. Extensive experiments were conducted using publicly available benchmark datasets. The results show that the proposed method can effectively improve the detection performance of MS COCO test-Dev2017.

The remainder of this article is organized as follows: Section 2 briefly introduces objection detection and label assignment. Section 3 presents the proposed framework. The experiment and result are shown in Section 4. Finally, the conclusion is drawn in Section 5.

# 2 Related Work

## 2.1 Label assignment strategy

Label assignment strategies can be classified as hard-label or soft-label assignment strategies. The output value of a hard-label assignment strategy is generally $\{0, 1\}$ or $\{-1, 1\}$, either positive or negative. The output value of a soft-label assignment strategy is a continuous value in the interval $[0, 1]$.

Anchor-based detectors usually adopt IoU, and a certain threshold as the assigning criterion [9], this is the most widely used hard-label assignment strategy. When the IoU between the anchor boxes and the GT boxes is greater than a given threshold, the sample is classified as positive; otherwise, the sample is classified as negative. To guarantee that each object has a training sample, Ren et al. introduced a minimum positive sample threshold [11]. this is the most widely used hard-label assignment strategy. When the IoU between the anchor boxes and the GT boxes is greater than a given threshold, the sample is classified as positive; otherwise, the sample is classified as negative. To guarantee that each object has a training sample, Ren et al. introduced a minimum positive sample threshold.

Zhang et al. [15] proposed an adaptive label assignment strategy based on feature statistics. The top $N$ samples with the smallest distance to the center of the corresponding GT box were selected. Their distances to the center of the GT box were used as the dynamic threshold to determine positive and negative labels. However, in reality, the center of the GT box does not necessarily fall on the object, leading to inaccurate label assignment.

Zhang [24] et al. proposed a soft-label assignment strategy to determine the positive labels by maximizing the detection likelihood. However, only the top K samples were selected as positive samples in this strategy; thus, these samples were not fully utilized. Zhu et al. [29] proposed a differentiable soft-label assignment strategy that expanded the selection of samples in both spatial and scale dimensions. A category-wise center-weighting module was designed to learn the distribution of each category and adapt to the data distribution of different categories. A confidence weighting module was also designed to update the positive and negative confidence of the locations in both the spatial and scale dimensions to adjust to the appearance and scale of each instance. [26, 27] proposed a dynamic label assignment strategy that resampled fixed $K$ samples as positives, however, such strategy of choosing the fixed $K$ positives is hard to fit varying sizes of instance on different scenes.

## 2.2 Feature encoding for classification and regression

In the early stages, the parameter-sharing detection network [9, 10] was commonly used for objection detection. However, Wu et al. [28] found that the feature distributions of classification and regression tasks were different, and a detection network with parameter sharing could not encode classification and regression features simultaneously.

Detectors that use parameter-independent parallel networks have been proposed [30, 31]. However, the locally receptive fields of the feature extractor are fixed; thus, they cannot dynamically adjust the receptive fields according to different spatial feature distributions.

With the development of spatial transformer networks (STN) [32], attempts have been made to introduce deformable feature extraction structures in parallel networks. Yang et al. [19] introduced deformable convolutions in the classification and regression branches to enhance the parallel detection network's ability to learn the contextual information and geometric information of the GT box, thus the two branches can more accurately encode categorical and regression features of different spatial distributions based on learnable convolution kernel shift. Song et al. [33] introduced deformable pooling and constructed a task-specific decoupling detection network, which allowed the network to encode classification and regression features with different spatial distributions accurately and combined the parallel detection network and task-specific decoupling network. The progressive constraint training mechanism further improves the encoding ability of the classification and regression branches to their respective features.

# 3 Methodology

## 3.1 Overview

Current label assignments generally focus on the bounding box regression quality; the consistency between sampling and the training goal of classification and regression are usually overlooked. We propose a triple-head network with an additional pre-regression branch designed to learn localization prior to guiding contextual information extraction for classification and regression tasks. A joint loss-aware label assignment strategy based on both classification and regression losses was designed to train the proposed triple-head network.

The FCOS network [20] is used as the baseline architecture; the parallel heads of the FCOS are substituted with a three-branch network to form a triple-head detection network (TDN) consisting of pre-regression, classification, and regression branches. During training, the feature output of the backbone network is first input into the pre-regression branch in the TDN (Fig. 1 (a)) to obtain the coarse location prediction of the object instance. Subsequently, as a localization prior, coarse prediction is passed on to the classification and regression branches to provide additional contextual information. The joint loss of the prediction results is used in the joint loss-aware label assignment strategy to generate positive and negative samples during training.

During inference, the joint loss-aware label assignment is no longer required. The regression branch is fine-tuned based on the prediction results of the pre-regression branch, and the classification branch is fine-tuned based on the location confidence predicted by the pre-regression branch. The outputs of these two branches are considered for the final classification and regression prediction. The following subsections detail the proposed label assignment strategy, the weighted assignment strategy for ambiguous samples, and the triple-head network.

## 3.2 Joint-loss-aware dynamic label assignment

The proposed joint loss-aware dynamic label assignment (JLA) is illustrated in Figure 2. The label assignment strategy includes two stages: foreground and background segmentation and positive and negative sampling. The proposed label assignment strategy can be formalized as $g\left(x_i\right)$:

$$g\left(x_i\right) = f_{\mathrm{g}}\left(L_u\left(x_i\right), f_m\left(L_u\left(x_i \cdot \mathbb{I}_{in}\left(x_i\right)\right)\right)\right) \tag{1}$$

where $x_i$ denotes the $i$-th sample. $\mathbb{I}_{in}$ is an indicator function for the foreground and background segmentation. When the sample is within the object bounding box, it is labeled as foreground, with a value of 1; otherwise, it is labeled as background, with a value of 0. $L_u\left(x_i\right)$ is a joint loss function defined as:

$$L_u\left(x_i\right) = \begin{cases} L_{cls}\left(x_i\right) + \alpha L_{reg}\left(x_i\right), & \mathbb{I}_{\mathrm{in}}\left(x_i\right) = 1 \\ L_{cls}\left(x_i\right) + \infty, & \mathbb{I}_{\mathrm{in}}\left(x_i\right) = 0, \end{cases} \tag{2}$$

where $L_{cls}$ and $L_{reg}$ represent the classification and regression loss functions of the $i$-th sample, respectively. $\alpha$ is a coefficient used to balance the impact of classification

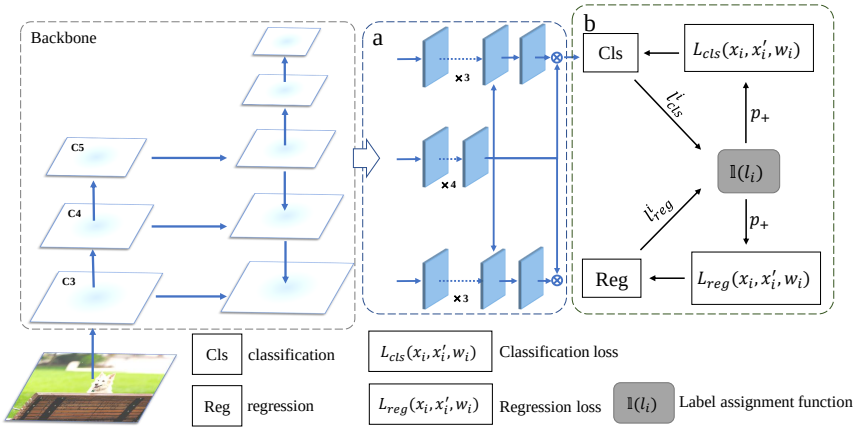


**Fig. 1**: Network architecture. (a) indicates the triple-head. (b) indicates the proposed label assignment strategy.

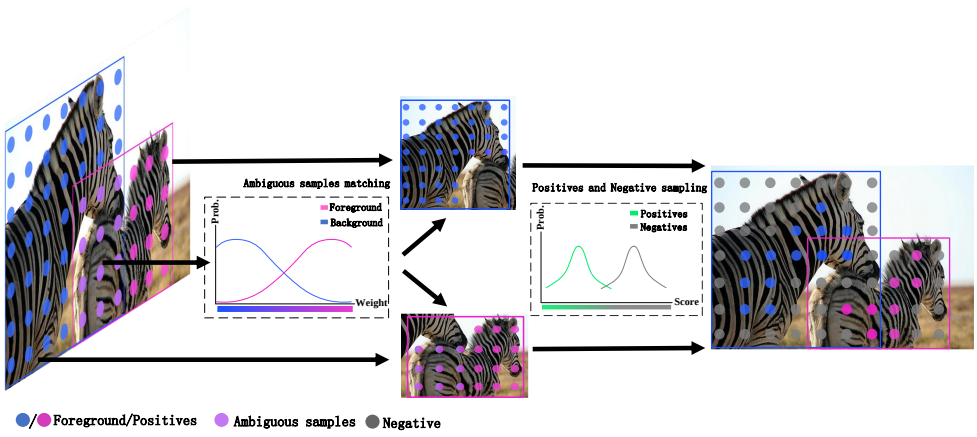


**Fig. 2**: Procedure of label assignment strategy. Left coordinate diagram represents that the matching weight between ambiguous sample and matched ground truth gradually increase, while others decrease. Right coordinate diagram represents the division of positives and negatives by the mixture models.

and regression on sample measurement. To force the model to consider both aspects equally, so it is set to 1.

$f_m(\cdot)$ is the decision boundary function of positive samples. The sample cluster with the lowest loss value is first obtained based on the Gaussian mixture model

(GMM); the boundary value of the cluster is used as the positive sample boundary, formulated as:

$$\begin{aligned} f_m\left(L_u\right) &= \arg\min_{L_{min}} \min\left(G\left(L_u \mid \Theta\right)\right) \\ G\left(L_u \mid \Theta\right) &= \sum_{i=1}^{N} \gamma_i \mathcal{N}\left(L_u \mid u_i, \Sigma_i\right) \end{aligned} \tag{3}$$

where $G(\cdot)$ denotes a Gaussian mixture model. $\gamma, \mu, \Sigma$ represent the weight, mean and variance of the $i$-th Gaussian model, respectively. $N$ denotes the number of mixture models (we set it to 2 in our experiments).

$f_g$ is the sampling function for positive and negative samples based on the decision boundary and the joint loss value; It labels samples whose joint loss value is less than the minimum positive sample boundary as positive, otherwise negative. It can be formulated as:

$$f_g\left(L_u, f_m\right) = \begin{cases} 1, & L_u \leq f_m \\ 0, & L_u > f_m \end{cases} \tag{4}$$

The proposed sampling method considered the overall goal of training loss minimization. Furthermore, a decision boundary through the GMM was used instead of a fixed value to determine the sampling decision boundary. This has two advantages: 1) the joint loss can accurately reflect the network's current learning state; 2) the decision boundary of positive and negative samples can be dynamically adjusted according to the network's learning state.

The proposed scheme chooses the top-$k$ ($k = 35$ in our setting) samples with the smallest joint loss rather than all samples for clustering in the GMM. An ablation study of different values is discussed in Section 4.3.1

During the foreground and background segmentation stages, multiple object-bounding boxes often overlap in dense scenes, leading to intractable ambiguities. Existing methods usually assign ambiguous samples to instances with the smallest areas [20] or lowest center distances [19], which may lead to suboptimal performance.

To solve the problem, we propose a weighted ambiguity samples label assignment strategy, and it is determined by the joint loss of the sample. The weight $w_{i,j}$ when the $j$-th overlapped sample matches to the $i$-th object is determined as:

$$\begin{aligned} w_{i,j} &= \frac{l^*_{i,j}}{\sum_i^n l^*_{i,j}} \\ l^*_{i,j} &= \frac{\sum_i^n l_{i,j}}{l_{i,j}} \end{aligned} \tag{5}$$

where $l_{i,j}$ represents the joint loss function of classification and regression when the $j$-th ambiguous sample matches to the $i$-th ground truth object, and $n$ is the number of ground truth objects corresponding to the overlapped samples. Compared with existing assignment approaches, weighted assignment is adaptive to the network training process.

To improve performance, a weight threshold was introduced into the ambiguity sample label assignment strategy. The ambiguous sample was assigned to the GT box when the weight of the ambiguous sample to that GT box was greater than the

threshold. Given ambiguous samples corresponding to two ground-truth objects, the weight threshold for multiple objects can be calculated as:

$$\theta^* = \theta + \frac{1}{n} - 0.5 \tag{6}$$

where $\theta$ is the weight threshold when the ambiguous samples correspond to two ground truth objects, $\theta^*$ is the matching weight when the ambiguous samples correspond to all ground truth objects. The experimental results from Section 4.3.2 show that the weighted assignment strategy is not sensitive to $\theta$; thus, heavy parameter tuning is not required.

In the early stage of training, the proposed weighted label assignment strategy is used; in the later stage of training, ambiguous samples are directly assigned to the instances with the smallest joint loss value, as the joint loss is sufficiently stable and reliable.

Although the designed JLA label assignment was inspired by [26, 27], the following differences distinguish this study:

1. In [26], ambiguous samples were directly assigned to GT instances with the lowest loss value. However, assigning ambiguous samples to GT instances with the lowest loss value may lead to an inference bias in the early stage of network training, deteriorating performance. We address this issue using a weighted label assignment strategy.
2. [26, 27] directly takes the fixed $K$ ($K \in [1, 16]$) samples as the positives of a GT instance. However, using a fixed $K$ does not conform to the actual training state of the network because GT instances may be of different scales and categories. Thus, we propose using a Gaussian mixture model to generate a decision boundary and dynamically sample the positive and negative samples.

### 3.3 Triple-head network

Fig. 3. shows the proposed triple-head network. The triple-head network consists of a pre-regression branch used to predict the coarse bounding box of the object instance and confidence and two task-specific branches for classification and regression. The standard convolution block, which consists of a $3 \times 3$ convolution, group normalization layer [34], and the ReLU layer, was used to construct the feature extraction module of the triple-head network. The numbers of convolutional blocks for the classification, regression, and pre-regression branches were three, three, and four, respectively. In addition to the output layer, the classification and regression branches had a deformable convolution.

In the training stage, similar to the Region Proposal Network (RPN), a coarse bounding box predicted by the pre-regression branch is passed into the task-specific branches as priors to guide the task-specific branches in extracting the contextual information of the instance. In the inference stage, the coarse bounding box and its confidence predicted from the pre-regression branch are used to fine-tune the classification and regression branches; the fine-tuned result is used as the final classification and regression output.

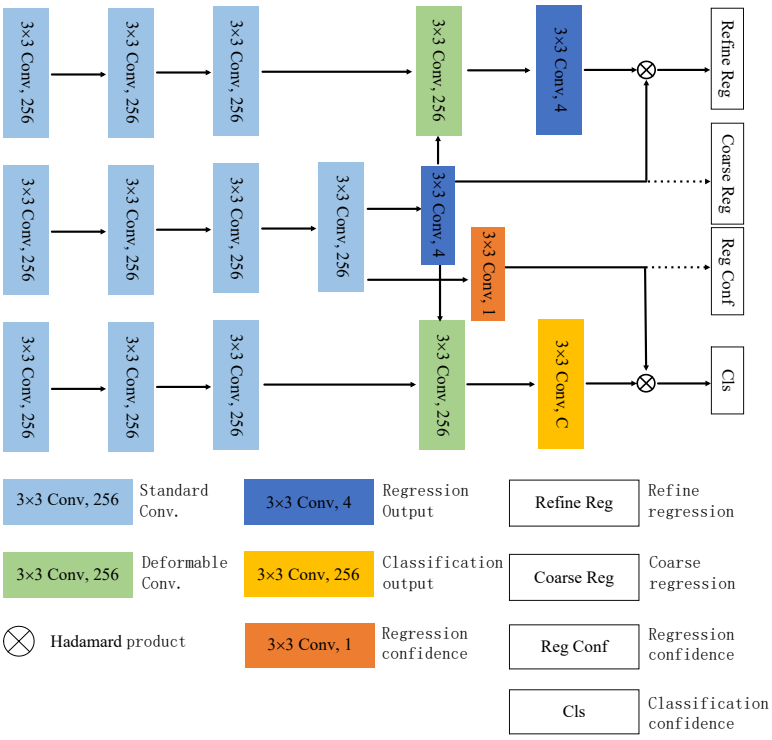


**Fig. 3**: Triple-head network.

In our implementation, the regression branch of the baseline network (FCOS) was directly used as the pre-regression branch, and a confidence prediction head was added to the output layer of the pre-regression branch. The two task-specific branches were similar to the pre-regression branch, except that the third convolution module was replaced by deformable convolution modules [35] to fuse the predicted coarse bounding box to the task-specific branches. By fusing the bounding box prior, the offset of the deformable convolution kernel can be formalized as:

$$d = d^* + \Delta d \tag{7}$$

where $d^*$ represents the base offset vector of the deformable convolution kernel, and $\Delta d$ represents the predicted coarse bounding box vector.

In the inference stage, the final bounding box is obtained by fusing the output of the regression branch and coarse prediction from the pre-regression branch. The fusion method is expressed as:

$$B = B^* \otimes \Delta d \tag{8}$$

where $B^*$ denotes the prediction of the regression branch and $\otimes$ denotes the Hadamard product.

In the inference stage, the classification branch is fine-tuned using the confidence predicted by the pre-regression branches (box refinement). Unlike a regression branch that uses the center distance, the IoU between the predicted bounding box and the GT bounding box is used as the training label in the fine-tuning stage.

# 4 Experiments

## 4.1 Datasets and evaluation metrics

MS COCO 2017 dataset is adopted in the experiments. MS COCO 2017 [36] has three subsets: 118K training set (train-2017), 5K validation set (val-2017), and 20K test set (test-dev), and it has a total of 500K labeled instances and 80 object categories. The labels of the COCO test set are not public, and the test results can be obtained by uploading them to the COCO server for verification.

The PASCAL VOC [37] dataset (VOC2007 and VOC2012) is adopted in ablation studies. VOC2007 has 5011 images for training (trainval) and 4952 images for testing (test). VOC2012 has 17125 images for training (trainval) and 5138 images for testing (test).

Average Precision (AP) is adopted as the evaluation metric, including AP (IoU threshold is 0.05), AP50 (IoU threshold is 0.5), AP75 (IoU threshold is 0.75), APS (average accuracy of small objects), APM (average accuracy of medium objects), APL (average accuracy of large objects).

## 4.2 Implementation details

The proposed system is implemented using the Detectron2, a PyTorch library. The batch size of the training data is 16 (4 times 4 graphics cards). Stochastic gradient descent with momentum (0.9) is adopted as the optimizer [38]. A constant warming-up mechanism is also adopted in training, and the number of prediction steps is 500 with a prediction factor of 0.33 and the gamma 0.1. The base learning rate is set to 0.01, and the weight decay is 0.0001. The proposed network is trained and tested on a workstation PC equipped with E5 CPU, 1080ti quad GPU, 128GB memory, on Ubuntu 18.04 system and CUDA 10.2.

On the COCO 2017 dataset, the maximum training steps using the ResNet50 backbone network is 90,000, and the input size is 500x1333. On the PASCAL VOC dataset, the network is jointly trained using the training and validation sets of VOC2007 and VOC2012 and tested on the VOC2007 test set. The maximum number of training steps is 45,000s; the input size is 600x1000.

FocalLoss [12] is adopted for all experiments. The coefficients $\alpha$ and $\gamma$ are set to 0.25 and 2.0, respectively. IoULoss [39] is adopted as the regression loss function.

**Table 1**: Parameter tuning of $K$ (ResNet50).

| K | AP | AP50 | AP75 |
|---|---|---|---|
| 25 | 37.9 | 57.9 | 40.3 |
| 30 | 37.9 | 58.2 | 40.3 |
| 35 | 38.0 | 58.2 | 40.5 |
| 40 | 37.8 | 57.6 | 40.1 |
| 45 | **37.9** | **57.8** | **40.7** |

## 4.3 Ablation study

### 4.3.1 Parameter tuning of $K$

In our method, top-$k$ samples with the smallest loss value are selected as the input of the GMM. In order to evaluate its influence on the sampling of the positive and negative samples, the detection performance on different values of $K$ is evaluated and shown in Table 1. It can be seen from the table that comparable performance can be achieved under different $K$, and $K$=35 can achieve slightly better performance. Thus, unless otherwise specified, $K$ is set to 35 in all subsequent experiments.

### 4.3.2 Parameter tuning of $\theta$

A weighted assignment strategy was used to assign ambiguous samples to the ground truth labels. In this strategy, when the weight between the ambiguous sample and ground truth object is greater than the set threshold, the ambiguous sample is labeled as the corresponding ground truth; otherwise, it is labeled as the ground truth object with the largest weight.

The detection performances with different thresholds $\theta$ are presented in Table 2. The ambiguous sample assignment strategy was not used for $\theta < 0.5$; we directly assigned the ambiguous samples to the ground truth object with the lowest loss value. From the table, poor performance is observed when $\theta = 0$, as the network parameters are randomly initialized, and the labeling of ambiguous samples in the early training stage is inaccurate.

However, when the ambiguous sample assignment strategy was applied ($\theta = \{0.6, 0.7, 0.8\}$), the model achieved better performance. However, no significant performance difference was observed with different values of $\theta$; this is possible because the ambiguous sample assignment strategy mainly works in the early stages of network training. After the early stages of training, the network can achieve accurate labeling of ambiguous samples. For the remaining experiments, we set $\theta = 0.6$ by default unless otherwise stated.

### 4.3.3 Inference Speed

Inference speed is an important metric indicating the computational effectiveness of models. We choose FCOS, Faster R-CNN and the proposed method to test the inference speed on MSCOCO val2017 dataset on single GTX1080ti gpu. As shown in Table 3, the proposed method is slower than FCOS, but Faster than Faster R-CNN. The

**Table 2**: Parameter tuning of $\theta$.

| $\theta$ | $K$ | JLA | AP | AP50 | AP75 |
|---|---|---|---|---|---|
| 0 | 35 | | 37.5 | 57.6 | 40.2 |
| 0.6 | 35 | ✓ | 38.0 | 57.8 | 40.6 |
| 0.7 | 35 | ✓ | 38.0 | 57.7 | 40.6 |
| 0.8 | 35 | ✓ | 38.0 | 57.8 | 40.6 |

main reason is that our method leverages the fusion of the bounding box prior, and this procedure includes offset calculation and deformable convolution.

**Table 3**: Inference speed

| Model | FCOS | Faster R-CNN | Ours |
|---|---|---|---|
| FPS | 12.6 | 6.4 | 10.1 |

#### 4.3.4 Parameter of model

The model parameters are presented in Table 4. The parameter of the proposed model is slightly higher than FCOS, but far less than the Faster R-CNN. Meanwhile, under the same training configuration (including the same training steps, learning rate, weight decay .etc), the proposed method obtain 3.2AP improvement.

**Table 4**: The number of parameters

| Model | FCOS | Faster R-CNN | Ours |
|---|---|---|---|
| Parameters | 32,244,250 | 41,704,036 | 34,614,307 |

### 4.4 Performance evaluation

In this section, the performance with different combinations of modules is evaluated, including the FCOS baseline, JLA label assignment, triple-head, and box refinement.

The baseline system is the original FCOS [20] excluding the center branch; only the classification and regression branches were used for training with the original label assignment strategy. The JLA replaced the original label assignment strategy of the FCOS with the proposed strategy based on the baseline.

As shown in Table 5, using our proposed label assignment strategy, the detection performance in terms of AP has improved by about 4.5%, reaching 38.0%. After introducing the pre-regression branch, the performance of the network in terms of the AP was further improved to 39.6% and 40.3%, respectively, demonstrating the effectiveness of the proposed method.

The visualization results of the positive and negative sampling are shown in Fig. 4. As the number of training steps increased, the proposed label assignment strategy achieved more accurate label assignments. The positive-sample prediction bounding

**Table 5**: Individual component ablation studies (Bold indicates the best performance).

| Baseline | JLA | Triplet Head | Box Refinement | IoU Pred | AP | AP50 | AP75 |
|---|---|---|---|---|---|---|---|
| FCOS | | | | | 37.0 | | |
| ✓ | | | | | 33.5 | 52.8 | 35.5 |
| ✓ | ✓ | | | | 38.0 | 58.2 | 40.5 |
| ✓ | ✓ | ✓ | | | 38.9 | 57.8 | 41.7 |
| ✓ | ✓ | ✓ | ✓ | | 39.6 | 58.0 | 42.3 |
| ✓ | ✓ | ✓ | | ✓ | 39.6 | 57.7 | 42.8 |
| ✓ | ✓ | ✓ | ✓ | ✓ | **40.3** | **57.8** | **43.3** |

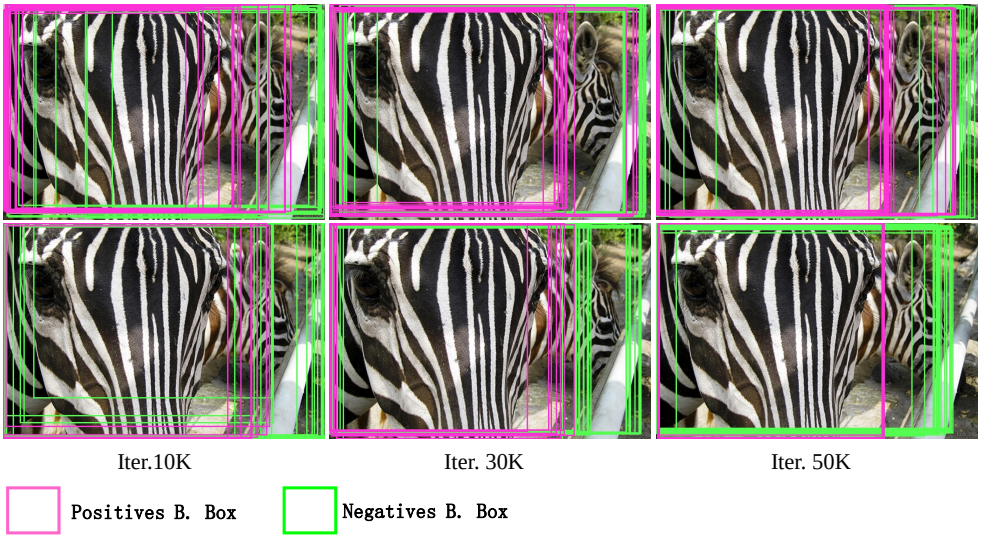


**Fig. 4**: visualization of positives and negatives in different training stages. The purple boxes are the positive instance predicted bounding boxes. The green boxes are the negative instance predicted bounding boxes. The first row is the label assignment of the baseline and the second row is our approach.

box was close to the object; the negative-sample bounding box was far from the object. Furthermore, the number of positive and negative samples dynamically improved and was better than that of the baseline model as the training steps increased.

We also provide a localization performance evaluation under different labeling strategies, including maximizing IoU label assignment [12], spatial-scale-constrained label assignment [20], statistics-based adaptive label assignment [15], and our proposed label assignment.

The first three label assignment strategies are trained with RetinaNet (Retina Network), FCOS (Fully Convolution One-Stage), and ATSS [15] (Adaptive Training Sample Selection), respectively. MS COCO val2017 data set is adopted in evaluation. The performance of each label assignment strategy under various IoU threshold are

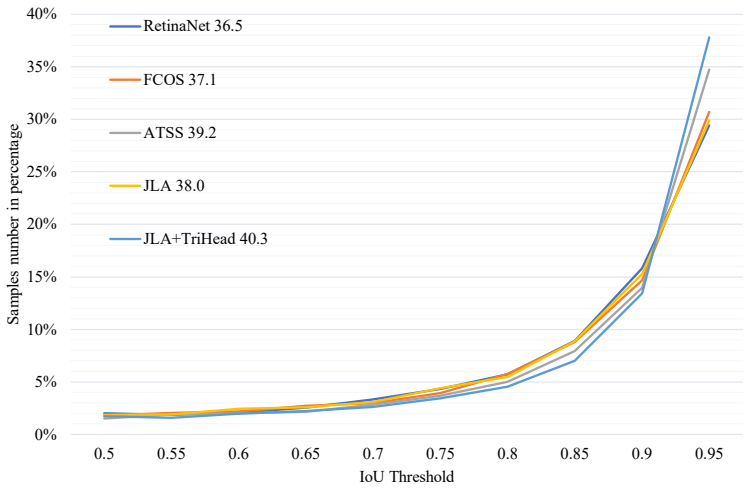


**Fig. 5**: Quantification of different label assignment strategies.

shown in Fig. 5. Note that the higher percentage of samples above the threshold indicates better performance.

The figure shows that the dynamic label assignment strategy ATSS and the JLA+Triple-Head network can achieve better prediction performance with an IoU of 0.95 thresholds. This indicates that the dynamic label assignment strategy can improve the detection accuracy for high-quality samples.

For JLA, the number of high-quality samples with an IoU value of 0.95 thresholds is lower than FCOS; this is because the JLA baseline network lacks an additional fine-tuning branch with confidence. Compared to RetinaNet, which also has no fine-tuning branch, JLA performs significantly better at the threshold of 0.95.

Performance at the threshold of 0.95 has been significantly improved for JLA+Triple-Head network. This indicates that the triple-head network can minimize the proposed loss value better, as the positive and negative sampling decision boundary is more stable in the label assignment strategy. The proportion of samples at the lower threshold of JLA+Triple-Head is lower than JLA, which proves that the joint loss value is indeed used as the quality indicator of the samples and can improve the detection performance of the network.

The detection performance of our method on the PASCAL VOC dataset is shown in Table 6. Compared to the hand-crafted label assignment strategy RetinaNet and FCOS, the proposed method can achieve a performance gain of 2.4% and 3.7% on AP, respectively. Compared with the statistical-based adaptive label assignment strategy ATSS, the proposed method can achieve a performance gain of 2.6%. This suggested the advantages of the proposed label assignment strategy and the weighted assignment strategy for ambiguous samples.

The detection performance of ATSS and FCOS is slightly worse than RetinaNet, as positive and negative sampling based on the distance between the center of the

**Table 6**: Detection performance on Pascal VOC dataset (Bold indicates the best performance).

| Model | AP | AP50 | AP75 |
|---|---|---|---|
| RetinaNet | 54.3 | 80.4 | 59.2 |
| FCOS | 53.0 | 79.0 | 58.2 |
| ATSS | 54.1 | 78.6 | 59.6 |
| Ours | **56.7** | **81.6** | **61.6** |

**Table 7**: Detection performance on Visdrone2019 (Bold indicates the best performance).

| Model | AP | AP50 | AP75 |
|---|---|---|---|
| RetinaNet | 11.8 | 21.4 | 11.6 |
| FCOS | 16.7 | 29.3 | 16.7 |
| Ours | **19.0** | **32.7** | **18.7** |

sample and the GT box is inaccurate. This shows that compared with the adaptive label assignment strategy that uses distance as a metric, the proposed joint loss aware label assignment strategy has a better generalization capability.

The detection performance of the dense object dataset Visdrone2019 is presented in Table 7. For a fair comparison, in addition to the dataset, the other configurations were not adjusted. Compared with the RetinaNet and FCOS, the proposed method can achieve a performance gain of 7.2% and 2.3% on AP, respectively, further demonstrating the effectiveness of the innovation.

The detection performance of difference advanced head is shown in Table 8. We trained all heads in FCOS detection framework on MSCOCO 2017 dataset using the same configuration. As shown in the table, the proposed method exhibited the best performance on the AP. The proposed method exhibited better performance on the AP75. This suggests that the triple-head network is more accurate than the other networks.

**Table 8**: Compare with difference advanced head (Bold indicates the best performance).

| Model | AP | AP50 | AP75 |
|---|---|---|---|
| BorderDet | 39.2 | 56.6 | 42.5 |
| DyHead | 39.4 | **58.4** | 42.2 |
| Ours | **39.5** | 57.6 | **42.8** |

## 4.5 Detection visualization

The detection results of FCOS, ATSS, and ours are visually shown in Fig. 6. Compared with FCOS and ATSS, our method achieved better performance. Taking the third

**Fig. 6**: Quantification of location prediction. The first, second, and third rows show the predictions of FCOS, ATSS, and ours, respectively.

column of the figure as an example, the object size is large, and the texture is similar to the background. FCOS and ATSS show false-negative detection results; our method can achieve a high recall detection rate as the 'keyboard' can be detected. Taking the fourth column as a dense scene example, FCOS shows false-positive detection results, and ATSS shows false-negative detection results; our method achieved the best detection results.

## 4.6 Comparison with state-of-the-art methods

The proposed framework is evaluated using ResNet50 and ResNet101 as the backbone on COCO 2017 since most existing label assignment strategy work adopts the ResNet [7] as the backbone network, COCO 2017 as the test set. The comparison is shown in Table 9. Our proposed framework can outperform traditional hand-crafted label assignment methods, including Faster R-CNN [11] (Faster Region-based Convolution Neural Network), Cas-RetinaNet [40] (Cascade Retina Network) and Cascade R-CNN [41] (Cascade Region- The maximum IoU label assignment strategy), the spatial-scale-constrained label assignment adopted in FCOS. Compared with FCOS, the current state-of-the-art method, our method improves the performance by about 3.5% in terms of AP on ResNet50. Compared with Cascade R-CNN on ResNet101, our method can achieve a performance gain of 2.2% in terms of AP.

Our proposed framework on ResNet50 can also outperform dynamic label assignment strategies, including GA-RPN [42] (Guided Anchoring Region Proposal Network), Libra R-CNN [43] (Libra Region-based Convolution Neural Network) Network) and Dynamic R-CNN [44] (Dynamic Region-based Convolution Neural Network). As shown in Table 9, our method can achieve a detection performance of 40.9% in terms of AP, which is 1.1% higher than the existing state-of-the-art method GA-RPN [42].

Our proposed framework on ResNet100 can outperform ATSS [15], SAPD [23] (Soft Anchor-Point Detection), FreeAnchor [24], Libra R-CNN [43], and PAA [45], while achieve comparable performance compared to IQDet [27]. In specific, our method can achieve a detection performance of 45.0% on Resnet101 in terms of AP, which is 1.4% higher than ATSS, 1.5% higher than SAPD, and 1.5% higher than FreeAnchor.

**Table 9**: Comparison with existing methods (Bold indicates the best performance).

| Method | backbone | AP | AP50 | AP75 | APS | APM | APL |
|---|---|---|---|---|---|---|---|
| Conventional label assignment | | | | | | | |
| Faster R-CNN [11] | ResNet50 | 36.2 | 58.5 | 38.9 | 21.0 | 38.9 | 45.3 |
| Faster R-CNN[11] | ResNet101 | 38.8 | 60.9 | 42.1 | 22.6 | 42.4 | 48.5 |
| RetinaNet[12] | ResNet101 | 39.1 | 59.1 | 42.3 | 21.8 | 42.7 | 50.2 |
| FCOS[20] | ResNet50 | 37.4 | 56.5 | 40.3 | 21.2 | 40.3 | 47.1 |
| FCOS[20] | ResNet101 | 41.5 | 60.7 | 45.0 | 24.4 | 44.8 | 51.6 |
| Cas-RetinaNet[40] | ResNet101 | 39.3 | 59.0 | 42.8 | 22.4 | 42.6 | 50.0 |
| Cascade R-CNN[41] | ResNet101 | 42.8 | 62.1 | 46.3 | 23.7 | 45.5 | 55.2 |
| Dynamic label assignment | | | | | | | |
| FreeAnchor[24] | ResNet101 | 43.1 | 62.2 | 46.4 | 24.5 | 46.1 | 54.8 |
| GA-RPN[42] | ResNet50 | 39.8 | 59.2 | 43.5 | 21.8 | 42.6 | 50.7 |
| Dynamic R-CNN[44] | ResNet50 | 39.1 | 58.0 | 42.8 | 21.3 | 40.9 | 50.3 |
| Dynamic R-CNN[44] | ResNet101 | 41.2 | 60.1 | 45.1 | 22.5 | 43.6 | 53.2 |
| Lib R-CNN[43] | ResNet50 | 38.7 | 59.9 | 42.0 | 22.5 | 41.1 | 48.7 |
| Lib R-CNN[43] | ResNet101 | 40.3 | 61.3 | 43.9 | 22.9 | 43.1 | 51.0 |
| SAPD[23] | ResNet101 | 43.5 | 63.5 | 46.5 | 24.9 | 46.8 | 54.6 |
| ATSS[15] | ResNet101 | 43.6 | 62.1 | 47.4 | 26.1 | 47.0 | 53.6 |
| PAA[45] | ResNet101 | 44.8 | 63.3 | 47.4 | 26.1 | 47.0 | 53.6 |
| IQDet[27] | ResNet-101-FPN | **45.1** | 63.4 | 49.3 | 26.7 | 48.5 | **56.6** |
| Ours | | | | | | | |
| Ours | ResNet50 | 40.9 | 58.5 | 44.3 | 23.1 | 44.1 | 51.4 |
| Ours | ResNet101 | 45.0 | **63.6** | **49.3** | **30.3** | **49.1** | 53.9 |

Compared to IQDet, a slightly 0.1% drop of AP is observed, while better performance in terms of AP50, AP75, APS, and APM has been achieved.

# 5 Conclusion

This study proposes a novel object detection framework consisting of a triple-head network and a joint loss-aware label assignment strategy. The proposed label assignment strategy uses the joint loss value of classification and regression as a metric to dynamically assign samples as positives and negatives, with an adaptive sampling process for the learning state of the network. In addition, a weighted assignment method was proposed on top of the joint loss function for ambiguous samples in the overlapping area of multiple GT boxes, avoiding inaccurate label assignment caused by handcrafted approaches. To further improve network performance, a triple-head network was proposed. The triple-head network has conventional classification and regression branches and an additional pre-regression branch that serves as a location prior for the other two branches and improves the accuracy of classification and regression by fine-tuning the classification and regression predictions.

Extensive experiments showed that the proposed method can effectively improve detection performance with different backbone networks on the MS COCO test-Dev2017 datasets.

In future research, we will design a detection network with higher efficiency. Optimizing an ambiguous sample assignment strategy is another interesting direction.

# References


[1] Li, K., Wan, G., Cheng, G., Meng, L., Han, J.: Object detection in optical remote sensing images: A survey and a new benchmark. ISPRS Journal of Photogrammetry and Remote Sensing **159**, 296–307 (2020)

[2] Ciaparrone, G., Sánchez, F.L., Tabik, S., Troiano, L., Tagliaferri, R., Herrera, F.: Deep learning in video multi-object tracking: A survey. Neurocomputing **381**, 61–88 (2020)

[3] Pal, S.K., Pramanik, A., Maiti, J., Mitra, P.: Deep learning in multi-object detection and tracking: state of the art. Applied Intelligence **51**(9), 6400–6429 (2021)

[4] Kong, Y., Fu, Y.: Human action recognition and prediction: A survey. International Journal of Computer Vision **130**(5), 1366–1401 (2022)

[5] G Martín, A., Fernández-Isabel, A., Diego, I., Beltrán, M.: A survey for user behavior analysis based on machine learning techniques: current models and applications. Applied Intelligence **51**(8), 6029–6055 (2021)

[6] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[7] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

[8] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)

[9] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)

[10] Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)

[11] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)

[12] Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)

[13] Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., Yang, J.: Generalized

focal loss: Learning qualified and distributed bounding boxes for dense object detection. Advances in Neural Information Processing Systems **33**, 21002–21012 (2020)

[14] Savarese, P.: On the convergence of adabound and its connection to sgd. arXiv preprint arXiv:1908.04457 (2019)

[15] Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9759–9768 (2020)

[16] Singh, B., Davis, L.S.: An analysis of scale invariance in object detection snip. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3578–3587 (2018)

[17] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European Conference on Computer Vision, pp. 21–37 (2016). Springer

[18] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)

[19] Yang, Z., Liu, S., Hu, H., Wang, L., Lin, S.: Reppoints: Point set representation for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9657–9666 (2019)

[20] Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9627–9636 (2019)

[21] Yang, T., Zhang, X., Li, Z., Zhang, W., Sun, J.: Metaanchor: Learning to detect objects with customized anchors. Advances in neural information processing systems **31** (2018)

[22] Zhu, C., He, Y., Savvides, M.: Feature selective anchor-free module for single-shot object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 840–849 (2019)

[23] Zhu, C., Chen, F., Shen, Z., Savvides, M.: Soft anchor-point object detection. In: European Conference on Computer Vision, pp. 91–107 (2020). Springer

[24] Zhang, X., Wan, F., Liu, C., Ji, R., Ye, Q.: Freeanchor: Learning to match anchors for visual object detection. Advances in neural information processing systems **32** (2019)

[25] Ke, W., Zhang, T., Huang, Z., Ye, Q., Liu, J., Huang, D.: Multiple anchor learning for visual object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10206–10215 (2020)

[26] Ge, Z., Wang, J., Huang, X., Liu, S., Yoshie, O.: Lla: Loss-aware label assignment for dense pedestrian detection. Neurocomputing **462**, 272–281 (2021)

[27] Ma, Y., Liu, S., Li, Z., Sun, J.: Iqdet: Instance-wise quality distribution sampling for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1717–1725 (2021)

[28] Wu, Y., Chen, Y., Yuan, L., Liu, Z., Wang, L., Li, H., Fu, Y.: Rethinking classification and localization for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10186–10195 (2020)

[29] Zhu, B., Wang, J., Jiang, Z., Zong, F., Liu, S., Li, Z., Sun, J.: Autoassign: Differentiable label assignment for dense object detection. arXiv preprint arXiv:2007.03496 (2020)

[30] Lin, B.-Y., Chen, C.-S.: Two parallel deep convolutional neural networks for pedestrian detection. In: 2015 International Conference on Image and Vision Computing New Zealand (IVCNZ), pp. 1–6 (2015). IEEE

[31] Li, Y., Chen, Y., Wang, N., Zhang, Z.: Scale-aware trident networks for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6054–6063 (2019)

[32] Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. Advances in neural information processing systems **28** (2015)

[33] Song, G., Liu, Y., Wang, X.: Revisiting the sibling head in object detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11563–11572 (2020)

[34] Wu, Y., He, K.: Group normalization. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19 (2018)

[35] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 764–773 (2017)

[36] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision, pp. 740–755 (2014). Springer

[37] Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The

pascal visual object classes (voc) challenge. International journal of computer vision **88**(2), 303–338 (2010)

[38] Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, pp. 818–833 (2014). Springer

[39] Yu, J., Jiang, Y., Wang, Z., Cao, Z., Huang, T.: Unitbox: An advanced object detection network. In: Proceedings of the 24th ACM International Conference on Multimedia, pp. 516–520 (2016)

[40] Zhang, H., Chang, H., Ma, B., Shan, S., Chen, X.: Cascade retinanet: Maintaining consistency for single-stage object detection. arXiv preprint arXiv:1907.06881 (2019)

[41] Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6154–6162 (2018)

[42] Wang, J., Chen, K., Yang, S., Loy, C.C., Lin, D.: Region proposal by guided anchoring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2965–2974 (2019)

[43] Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., Lin, D.: Libra r-cnn: Towards balanced learning for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 821–830 (2019)

[44] Zhang, H., Chang, H., Ma, B., Wang, N., Chen, X.: Dynamic r-cnn: Towards high quality object detection via dynamic training. In: European Conference on Computer Vision, pp. 260–275 (2020). Springer

[45] Kim, K., Lee, H.S.: Probabilistic anchor assignment with iou prediction for object detection. In: European Conference on Computer Vision, pp. 355–371 (2020). Springer