Offset Bin Classification Network for Accurate Object Detection

Heqian Qiu, Hongliang Li*, Qingbo Wu, Hengcan Shi
University of Electronic Science and Technology of China
Chengdu, China
hqqiu@std.uestc.edu.cn, hlli@uestc.edu.cn, qbwu@uestc.edu.cn, shihc@std.uestc.edu.cn

Abstract

Object detection combines object classification and object localization problems. Most existing object detection methods usually locate objects by leveraging regression networks trained with $Smooth\,L_1$ loss function to predict offsets between candidate boxes and objects. However, this loss function applies the same penalties on different samples with large errors, which results in suboptimal regression networks and inaccurate offsets. In this paper, we propose an offset bin classification network optimized with cross entropy loss to predict more accurate offsets. It not only provides different penalties for different samples but also avoids the gradient explosion problem caused by the samples with large errors. Specifically, we discretize the continuous offset into a number of bins, and predict the probability of each offset bin. Furthermore, we propose an expectation-based offset prediction and a hierarchical focusing method to improve the prediction precision. Extensive experiments on the PASCAL VOC and MS-COCO datasets demonstrate the effectiveness of our proposed method. Our method outperforms the baseline methods by a large margin.

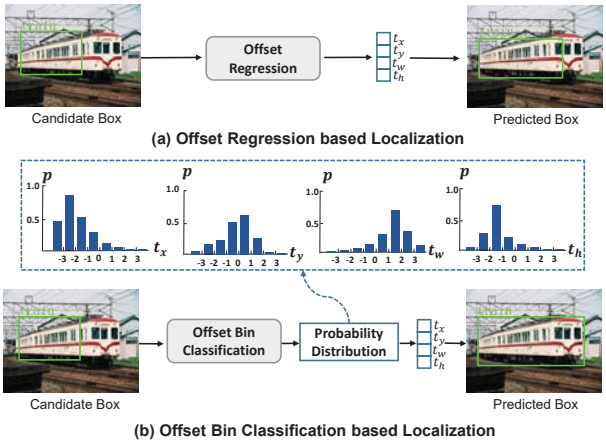

Figure 1. A comparison of typical offset regression based object detection method and our proposed offset bin classification method. (a) The typical object detection method locates objects based on offset regression. (b) The proposed method locates objects based on the output probability distribution over different offset bins. The typical offset regression method make limited offsets from the candidate box towards the object, whereas this problem is avoided by the offset bin classification method.

1. Introduction

Object detection is a fundamental yet challenging computer vision task, which includes object classification and object localization problems. A broad set of computer vision applications, such as autonomous driving [7, 17, 39–41], video surveillance [6, 24] and robotics [38, 42, 45] will benefit from accurate object localization.

Most of state-of-the-art object detection methods [1, 8, 11, 12, 20, 21, 26, 30, 31, 35, 44] firstly generate a series of candidate boxes and then predict offsets for these boxes to locate objects, as shown in Figure 1 (a). Since offsets are continuous values, these methods predict them by leveraging regression networks that are optimized using the L_2 or $Smooth\,L_1$ losses. However, as investigated by [9], the L_2 loss [10] may cause gradient explosions when there are large offset errors. To avoid this problem, the $Smooth\,L_1$ loss [9] weakens the effects of the samples with large errors by clipping their gradients. Although the $Smooth\,L_1$ loss solves the gradient explosion problem, it cannot penalize enough the samples with large errors, which results in suboptimal regression networks and inaccurate offsets between candidate boxes and objects. For example, in Figure 1 (a), the train object can not be tightly surrounded by a bounding box.

To address this problem, we propose an offset bin classification network to predict more accurate offsets, as shown in Figure 1 (b). The proposed method adopts a classification network trained with a cross entropy loss rather than a $Smooth\,L_1$ or L_2 loss. On the one hand, it gives samples with different offset errors adequate penalties. On the

*Corresponding author.

other hand, it avoids the gradient explosion problem. Nevertheless, the classification network can only predict discrete offset values. Therefore, we propose an expectation-based offset prediction and a hierarchical focusing offset prediction to further improve the prediction precision.

Specifically, we quantize the continuous offset into a number of bins using the uniform discretization and then train an offset bin classification network with a cross entropy loss to predict the probability distribution of offset bins. Inspired by [37], we turn the classification results into the object location by calculating the softmax expected value of discretized offset bins. Meanwhile, we propose a hierarchical focusing offset prediction network to gradually refine offset bins for more precise object localization. We validate the effectiveness of our method on two common object detection datasets, including the PASCAL VOC and MS-COCO datasets. The results show that our proposed method is beneficial to accurate object detection.

Our contributions can be summarized as follows:

• We propose an offset bin classification network to predict more accurate offsets instead of regression networks optimized by $Smooth\,L_1$ or L_2 loss.

• To further produce more precise object localization, we propose an expectation-based offset prediction and a hierarchical focusing offset prediction.

• Extensive experiments on two common datasets demonstrate the effectiveness of the proposed methods.

2. Related Work

Object Detectors: Modern object detection frameworks usually can be classified as two-stage and single-stage detectors. In two-stage detectors [1, 8, 11, 12, 20, 21, 26, 30, 31, 35, 44], a sparse set of region proposals that may contain objects are first generated, and then their features are extracted for the following classification and localization. The representative methods, including Faster R-CNN [35], FPN [20] and Mask R-CNN [12], have achieved dominated performance on various benchmarks. Compared with two-stage detectors, single-stage detectors [18, 19, 21, 23, 32–34] reach high inference speed, such as YOLO [32–34], SSD [23], RetinaNet [21]. They usually skip the region proposal generation step and directly predict bounding boxes following the anchor box scheme. Although these methods have detected objects successfully, it is still a challenging problem to achieve accurate object localization.

Bounding Box Regression: In order to solve the problem of object localization, most of object detection methods [1, 8–11, 15, 26, 28, 44] leverage bounding box regression networks to predict offsets of four coordinates that transform candidate boxes to objects. R-CNN [10] predicted these offsets by training a linear regression model with L_2 loss. However, it is easy to cause gradient explosion when there are some samples with large errors. Replacing L_2 loss, Fast R-CNN [9] proposed $Smooth\,L_1$ loss to reduce the effects of the samples with large errors, which has been widely accepted for regression in object detection. Balanced L1 loss [28] further increased the gradient contribution of the samples with small errors to rebalance the the involved classification and localization tasks as well as samples with different attributes. A different approach KL loss [14] took the ambiguities of ground truth bounding boxes into account and learned bounding box regression and localization variance for more accurate object localization. In addition, UnitBox [46] and GIoU [36] directly used the evaluation metric as object functions to address the gap between optimizing the commonly used distance loss and maximizing metric values. However, it is hard to optimize different bounding boxes with the same IoU.

In addition, a series of object detectors [1, 8, 11, 26, 44] attempt to improve the object localization by iteratively regressing bounding boxes. They both cascaded multiple regressors and fed the detection results after each iteration into the next bounding box regressor. Cascade R-CNN [1] considered the distribution of detection outputs and resampled bounding boxes at each iteration to guarantee the matching between the quality of detector and that of testing. However, it is non-monotonic to improve the location accuracy as the number of iterations increases. IoU-Net [15] proposed to predict the IoU with matched ground-truth as the localization confidence to guide the regression of bounding box. Instead of regression network, we propose an offset bin classification network with a cross entropy loss to achieve more accurate object localization, which is also effectively turned in other computer vision areas. For example, [27] predicted the detection heatmaps and the associative embedding tags for human pose estimation. [5] trained a depth estimation network by using an ordinal regression loss instead of a L_2 loss.

Recently, some anchor-free methods [16, 43, 47] directly predict the heatmaps of keypoints of bounding boxes, and introduce different kinds of loss functions to refine and group these keypoints for the final detected bounding boxes. CornerNet [16] used a SmoothL1 loss to regress the local offsets, and pull loss and push loss to constrain the distances between keypoints. CenterNet [47] regressed localization offset and object size using two L1 loss functions. FCOS [43] employed an IoU loss to regress the area of bounding box. Unlike the proposed method, they usually require carefully group keypoints for final objects.

3. Approach

In this section, we first review and analyze the problem of the conventional bounding box regressors. Then, we introduce our proposed offset bin classification network to address this problem, which is implemented based on popular FPN [20].

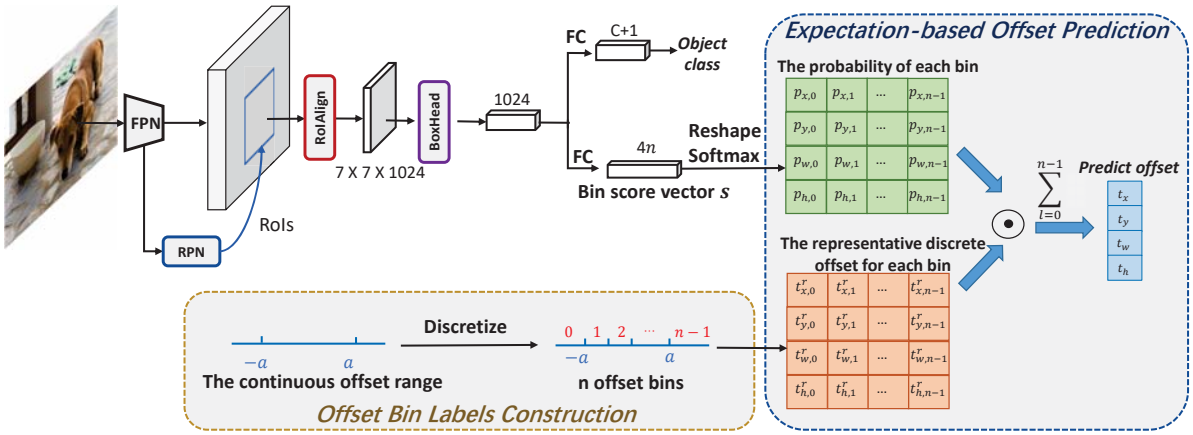

Figure 2. The overall architecture of our proposed offset bin classification method for object detection. It consists of three main parts: RoI features extraction, offset bin labels construction and expectation-based offset prediction. The RoI features are extracted by the backbone network FPN [20]. The offset bin labels construction is to discretize the continuous offset range to several offset bins. The expectation-based offset predict is used to turn the classification results to offset estimation by calculating a expected value.

3.1. Revisiting Bounding Box Regression

Let (x, y, w, h) be the center coordinates of bounding box and its width and height. Following R-CNN [10], the common methods leverage regression networks to learn offsets that transform candidate boxes to ground-truth boxes. They parameterize the offsets of four coordinates as follows:

$$\begin{aligned} t_x &= (x - x_a)/w_a, t_y = (y - y_a)/h_a \\ t_w &= log(w/w_a), t_h = log(h/h_a) \\ t_x^* &= (x^* - x_a)/w_a, t_y^* = (y^* - y_a)/h_a \\ t_w^* &= log(w^*/w_a), t_h^* = log(h^*/h_a) \end{aligned} \tag{1}$$

where t_x, t_y, t_w, t_h are the predicted offsets, $t_x^*, t_y^*, t_w^*, t_h^*$ are the target offsets. x, x^* and x_a (likewise for y, w and h) are from the predicted box, ground-truth box and the candidate box (anchor or proposal box) respectively. The goal is to minimize the errors between the predicted and target offsets:

$$L_{loc} = \sum_{i \in \{x,y,w,h\}} L_{reg}(t_i - t_i^*) \tag{2}$$

where L_{reg} is squared-error L_2 loss function in R-CNN [10]. However, it is sensitive to some samples when there is a large offset errors.

Replacing L_2 loss, Fast R-CNN [9] adopts $Smooth\,L_1$ loss function to evade the above problem:

$$Smooth\,L_1(x) = \begin{cases} \frac{x^2}{2\beta}, & |x| \le \beta \\ |x| - \frac{\beta}{2}, & otherwise \end{cases} \tag{3}$$

$$\frac{\partial Smooth\,L_1}{\partial t_i} = \frac{\partial Smooth\,L_1}{\partial x} \begin{cases} \frac{x}{\beta}, & |x| \le \beta \\ sgn(x), & otherwise \end{cases} \tag{4}$$

where the deviation $x = t_i - t_i^*$, β is usually set to 1 in two-stage detectors. sgn represents symbolic function. Note that the samples with the offset error larger than β are forced to clip the gradients to 1 or -1 for reducing their effects, causing insufficient penalty for these samples. So, the regression networks optimized by the $Smooth\,L_1$ loss function predict inaccurate offsets between candidate boxes and objects.

3.2. Offset Bin Classification Network

To address this problem, we propose an offset bin classification network to achieve more accurate object localization. The overall architecture of the proposed method is illustrated in Figure 2. Given an image, we first generate a sparse set of candidate boxes using Region Proposal Network (RPN) [20] and then extract these RoIs features from the image feature maps obtained by feature pyramid networks (FPN) [20]. Based on the extracted RoI features, we predict their corresponding object categories and offset bin confidence scores instead of concrete offset values. Moreover, we use the expectation-based offset prediction and the hierarchical focusing offset prediction in Figure 3 to further improve the precision of predicted offsets.

3.2.1 Offset Bin Labels Construction

As shown in Figure 4, we quantize the continuous offset in Section 3.1 into a set of representative discrete offsets. Divide the offset range $(-a, a)$ uniformly into m non-overlapping bins. The width w of each bin in the range $(-a, a)$ is $\frac{2a}{m}$. In addition, we also separately divide the range $(-\infty, -a]$ and $[a, +\infty)$ into two bins. Thus, the total number of bins is denoted as $n = m + 2$. The discrete bin labels are denoted as $L \in \{0, 1, ..., n-1\}$. The representa-

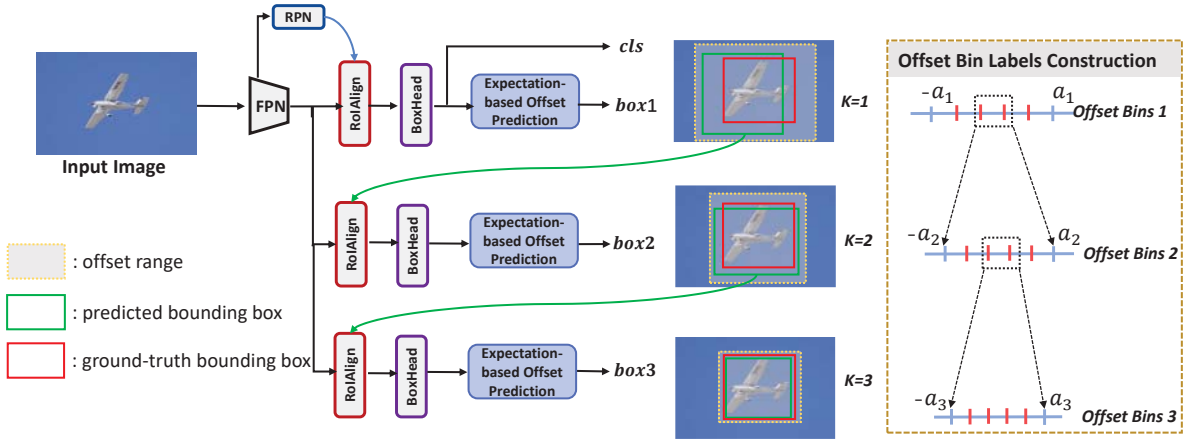

Figure 3. The architecture of the proposed hierarchical focusing offset prediction. Here, we show three stages in the hierarchical focusing offset prediction. Yellow dashed boxes filled with gray denote the offset range in each stage. Green boxes and red boxes represent predicted boxes and ground-truth boxes in each stage. The offset range in each stage is defined within the offset bins of previous stage.

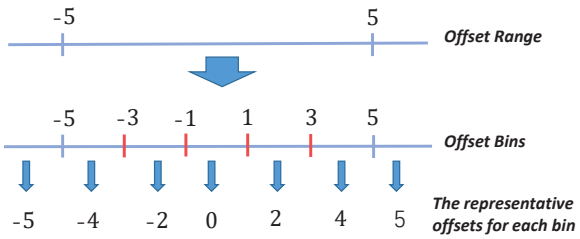

Figure 4. Illustration of offset bin construction. The offset range $(-5, 5)$ is uniformly discretized into five bins, and the median values of each bin stand for their representative offsets. In addition, the range $(-\infty, -5]$ and $[5, +\infty)$ uses the endpoint -5 and 5 as their representative offset, respectively.

tive offset for each bin can be indicated as follows:

$$t^r_{i,l} = \begin{cases} -a + (l + \frac{1}{2}) * w & l \in [0, m] \\ -a & l = m + 1 \\ a & l = m + 2 \end{cases} \quad (5)$$

where $t^r_{i,l}$ is the representative offset corresponding to the bin label l for the coordinate i of the bounding box. The representative offsets for the labels from 0 to m are expressed as the median value of each bin, and the other labels are expressed as the offset of the endpoint.

3.2.2 Network Learning

Based on the discretized offset bin labels, it is straightforward to cast the object localization as the multi-class classification problem instead of directly regression. As shown in Figure 2, the candidate box is fed into the BoxHead of backbone network FPN [20] to generate its offset bin score vector $s \in R^{4n}$, where 4 is the four coordinates of the bounding box, n is the number of offset bins. Then we reshape the score vector to $R^{4\times n}$ and normalize respectively the score vector of each coordinate into the form of probability by a softmax function as follows:

$$p_{i,l} = \frac{exp(s_{i,l})}{\sum_{l=0}^{n-1} exp(s_{i,l})} \quad (6)$$

where $p_{i,l}$ indicates the probability of the i-th coordinate offset belongs to the l-th bin.

The loss function L_{bin} for the offset bin classifier is formulated as a cross entropy loss:

$$L_{bin}(p_{i,l}, l) = -\sum_{i\in\{x,y,w,h\}} \sum_{l=0}^{n-1} y_l * log p_{i,l} \quad (7)$$

in which the loss is calculated when the ground-truth class is labeled l, where $y_l \in \{0, 1\}$. The gradient with regard to the output score s^b_{il} of the classifier layer can be derived as follows:

$$\frac{\partial L_{bin}}{\partial s_i} = \begin{cases} -\sum_{i\in\{x,y,w,h\}}(p_{i,l} - 1), & y_l = 1 \\ -\sum_{i\in\{x,y,w,h\}}(p_{i,l}), & y_l = 0 \end{cases} \quad (8)$$

Based on the above formula, the gradient is bounded and its norm is limited to $[0, 1]$, which is more stable for all samples compared with L_2 loss function. Meanwhile, it effectively takes into account the samples using different gradient contributions based on the predicted probabilities $p_{i,l}$ compared to $Smooth\ L1$ loss.

To end up, we use the loss function L to end-to-end train our network for accurate object detection:

$$L = L_{cls} + \lambda_{bin} L_{bin} \quad (9)$$

where L_{cls} denotes the loss for classification of objects, The offset bin classification loss L_{bin} is used for localization of objects. λ_{bin} is the weight that control the balance among these losses. In this paper, we set λ_{bin} to 1.

Method	Expectation	Hierarchical	AP	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{90}
Bounding Box Regression [20]			45.0	**74.5**	**69.5**	57.6	36.0	6.6
Bin Classification			45.8	73.3	67.9	57.2	39.6	9.8
Bin Classification	✓		47.5	74.0	69.0	58.8	41.5	13.6
Bin Classification		✓	47.5	72.8	67.9	58.1	42.0	16.0
Bin Classification	✓	✓	**49.0**	73.2	68.4	**59.0**	**44.3**	**19.6**

Table 1. The effects of each component in the proposed method. Results are reported on the VOC2007 *test* set [4]. The baseline method with ResNet-50-FPN [20] locates object by bounding box regression method. Expectation and Hierarchical represent the expectation-based offset prediction and hierarchical focusing offset prediction.

3.2.3 Expectation-based Offset Prediction

Since offsets are continuous values with high precision, the classification network only predicts discrete offset values. Thus, we propose two different methods to improve the precision of detection results: the expectation-based offset prediction and the hierarchical focusing offset prediction.

For the expectation-based offset prediction method in Figure 2, we utilize the probability distribution over different offset bins to estimate the predicted offset t_i, which is calculated by a softmax expected value instead of a max value, as follows:

$$\begin{aligned} t_i &= \mathbb{E}(T_i^r) \\ &= \sum_{l=0}^{n-1}(p_{i,l} * t_{i,l}^r) \end{aligned} \tag{10}$$

where $T_i^r = \{t_{i,0}^r, t_{i,1}^r, ..., t_{i,n-1}^r\}$ denotes the set of representative discrete offsets for n bins. The symbol $\mathbb{E}$ indicates the expectation of discrete offsets.

3.2.4 Hierarchical Focusing Offset Prediction

Furthermore, we propose a hierarchical focusing offset prediction with a coarse-to-fine strategy to gradually refine the bin interval as shown in Figure 3. The discretized value will be closer to the target value when the bin interval is very small. Assume that there are K stages and n_k bins in the k-th stage. In each stage, the offset range $(-a_k, a_k)$ is defined within the offset bins of previous stage. So, the width w_k of bins can be denoted as $\frac{w_{k-1}}{n_k}$. Then, we predict the offset t_i^k of each stage similar to Section 3.2.3. The final predicted offset can be calculated as:

$$t_i = \sum_{k=1}^{K} t_i^k \tag{11}$$

As shown in Figure 3, in the first stage, we predict offsets between candidate boxes generated by RPN and objects within the offset range $(-a_1, a_1)$. Subsequently, at each stage, we predict finer offsets within the previous offset bin. By progressively classifying offsets, we can obtain more precise bounding boxes.

4. Experiments

To evaluate the effectiveness of the proposed offset bin classification network, we conduct extensive experiments on two standard object detection datasets, including the PASCAL VOC dataset [4] and the MS-COCO dataset [22].

Datasets. The PASCAL VOC dataset [4] contains 20 object categories, which consists of the PASCAL VOC2007 dataset and the PASCAL VOC2012 dataset. Following [35], we train our network on the union of VOC 2007 *trainval* and VOC2012 *trainval* sets, including 5011 and 11540 images, respectively, and evaluate on the VOC2007 *test* set containing 4952 images. The MS-COCO dataset [22] involves 80 object categories, which has larger scale than the PASCAL VOC dataset. Following the common practice [20, 28], we use the *train-2017* set with 115K images for training and report the final results on the *test-dev* set with 20k images.

Evaluation Metrics. We adopt the standard COCO-style Average Precision (AP) to measure the detection performance of various qualities, which averages mAP across different IoU thresholds from 0.5 to 0.95 with an interval of 0.05. It also includes AP across small scale AP_S, medium scale AP_M and large scale AP_L.

Implementation Details. For fair comparison, we implement all experiments based on PyTorch [29] and MMDetection [2]. We employ FPN [20] based on ResNet-50 and ResNet-101 [13] as the baseline networks. Following the typical convention, we adopt the input image scale of 1000×600 on the PASCAL VOC dataset [4] and a scale of 1333×800 on the MS-COCO dataset [22]. We train detectors end-to-end with 2 GPUs (2 images per GPU) for 12 epoch. The initial learning rate is set to 0.005 and decreased by a factor 0.1 after 8 epochs and 11 epochs. Unless otherwise specified, all other hyper-parameters follow the default settings in MMDetection [2]. The loss weights λ_{bin} are set to 1. The offset range a and the number of bins n are set to 3 and 20, respectively. In the hierarchical focusing offset prediction, the number of stages K is set to 2.

4.1. Ablation Study

In this section, we validate the effectiveness on the baseline ResNet-50-FPN [20]. Without loss generality, we per-

Method	AP	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{90}
L_2 Loss [10]	44.7	72.6	67.6	56.8	37.4	7.8
$Smooth\,L_1$ Loss [20]						
$\beta = 1.0$	45.0	**74.5**	**69.5**	57.6	36.0	6.6
$\beta = 1.5$	44.3	73.9	68.6	56.5	34.9	6.4
$\beta = 2.0$	44.2	74.3	68.9	56.1	33.9	6.2
Bin Classification	**47.5**	74.0	69.0	**58.8**	**41.5**	**13.6**

Table 2. The effectiveness of different loss functions. β denotes the division point in the $Smooth\,L_1$ loss function. Results are reported on the VOC2007 *test* set [4].

Stage	AP	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{90}
$K = 1$	47.5	**74.0**	**69.0**	58.8	41.5	13.6
$K = 2$	**49.0**	73.2	68.4	**59.0**	**44.3**	**19.6**
$K = 3$	48.8	73.3	68.3	58.5	43.6	19.1

Table 3. The effectiveness of number of stages in the proposed hierarchical focusing offset prediction method. Results are reported on the VOC2007 *test* set [4].

form ablation studies to reveal the effect of each component in our proposed method on the PASCAL VOC dataset [4].

Main Component Analysis. We analyze the effect of each proposed component in Table 1. Simply estimating object localization by the proposed offset bin classification method improves the AP by 0.8% compared with the baseline bounding box regression method [20]. Introducing expectation-based offset prediction and hierarchical focusing offset prediction both achieve gain of 2.5% compared with the baseline, which further boost the prediction precise. The expectation-based offset prediction takes into account the probability of samples in other offset bins to estimate offsets, and consistently improves AP with different IoU metrics. The hierarchical focusing offset prediction performs better in the high IoU metrics. The reason is that it predicts more precise offsets within finer offset bin. Ultimately, our full method outperforms the baseline bounding box regression method by 4.0%. The result demonstrates that the effectiveness of the proposed method in terms of more accurate object detection, especially performing better in the high IoU metrics.

Effectiveness of Different Loss Function for Predicting Offsets. The effectiveness of different loss function for predicting offsets is shown in Table 2. Based on the same backbone network ResNet-50-FPN [20], we adjust the division point β of regression loss $Smooth\,L1$ to make more samples be treated based on enough gradient contributions. However, the detection performance AP is decreased when we set β to a larger value. One possible reason is that the network learning is dominated by some samples with large distance error. Compared with the $Smooth\,L_1$ loss and the L_2 loss, our method performs better performance as shown in Table 2, which alleviates the problem by the offset bin classification.

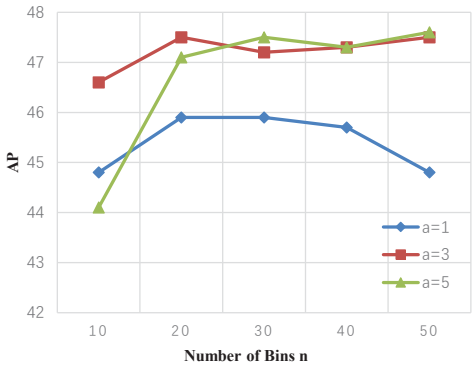

Figure 5. The effectiveness of bin classification for offset bin labels with different hyper-parameters.The horizontal axis represents the number of bins n, the vertical axis stands for detection performance AP. The blue line, the red line and the green line indicate the offset range $a = 1, 3, 5$, respectively.

Setting of Offset Bin Labels. Figure 5 shows the effectiveness of bin classification for offset bin labels with different hyper-parameters. a and n respectively denote the endpoint of the divided offset range and the number of bins. When the number of bins n is fixed, it can be seen that the detection performance is decreased for $a = 1$, while the performance is similar for $a = 3$ and $a = 5$. This is because many samples with offset greater than 1 are ignored during training if $a = 1$. When the endpoint $a = 3$ or 5, it can be observed that the detection performance are very close to each other when the number of bins n is set from 20 to 50, thereby is robust to a long range of offset bin numbers. In addition, the detection performance is relatively poor when n is small (i.e. $n = 10$). To balance the performance with the bin numbers, we choose $a = 3$ and $n = 20$ in our experiments.

Number of Stages in Hierarchical Focusing Offset Prediction. The effectiveness of number of stages in

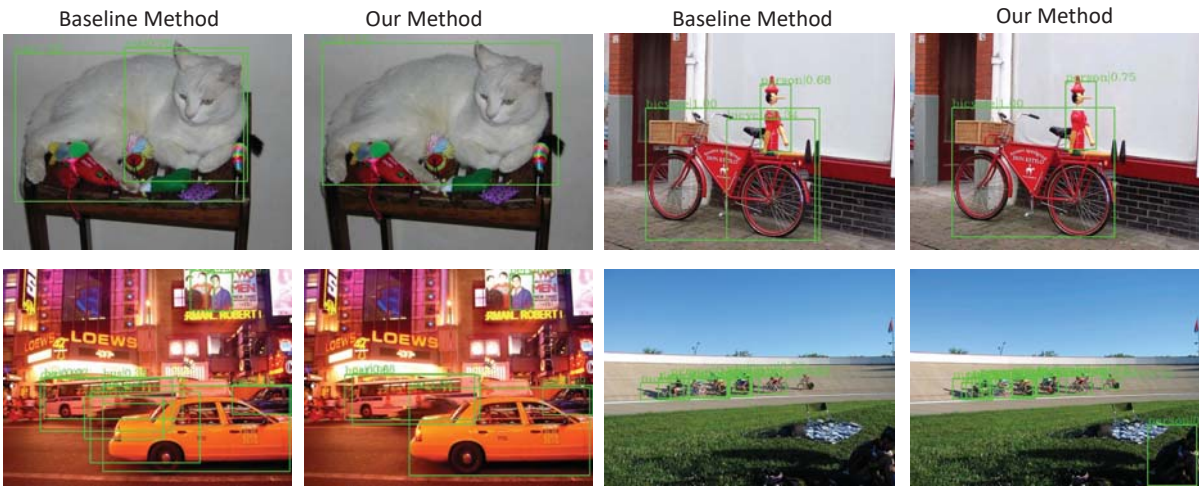

Figure 6. Visualization comparison between the baseline method and the proposed offset bin classification method on the VOC2007 *test* set [4]. The first and third columns show the detection results of the baseline method. The second and fourth columns show that the detection results of our method.

Method	Backbone	AP	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{90}
Faster R-CNN* [35]	ResNet-50-FPN	45.0	**74.5**	**69.5**	57.6	36.0	6.6
Our +Faster R-CNN [35]	ResNet-50-FPN	**49.0**	73.2	68.4	**59.0**	**44.3**	**19.6**
Faster R-CNN* [35]	ResNet-101-FPN	47.8	**75.5**	**70.6**	60.3	41.3	10.5
Our+Faster R-CNN [35]	ResNet-101-FPN	**50.8**	74.0	69.5	**60.8**	**47.2**	**22.5**
Cascade R-CNN* [1]	ResNet-50-FPN	49.5	73.1	**69.0**	**61.0**	45.9	18.1
Our+Cascade R-CNN [1]	ResNet-50-FPN	**50.4**	**73.3**	68.9	60.4	**46.5**	**22.2**
Cascade R-CNN* [1]	ResNet-101-FPN	51.0	73.6	69.6	61.9	48.3	21.1
Our+Cascade R-CNN [1]	ResNet-101-FPN	**51.9**	**73.9**	**69.8**	**62.1**	**48.7**	**25.0**

Table 4. Comparison with state-of-the-art methods on VOC2007 *test* set [4]. The symbol * represents our re-implement results based on MMDetection [2].

hierarchical focusing offset prediction is shown in Table 3. According to the analysis in Figure 5, we set the number of bins n_k in each stage to be same ($n_k = 20$, $k = 1, 2, 3$) and the endpoint $a_1 = 3$ in the first stage. Thus, the end point of offset range a_2 in second stage and a_3 in third stage are set to 0.15 and 0.015, respectively. It can be seen that the detection results AP is improved by 1.6% compared with only one stage when the number of stages $K = 2$. In the second stage, the width of bin is already within a very small range. Adding the third stage, the detection performance is close to the second stage. It can be seen that the bin classification with two stages can achieve the better detection performance.

Visualization Comparison. Figure 6 shows the visualization comparison between the baseline method [20] and the proposed offset bin classification method. It can be observed that the baseline method [20] assigns some bounding boxes that do not tightly surround objects in the first row images of Figure 6, while our method can detect objects more accurately. The second row images of Figure 6 show that the car object and the person object are missed detection in the baseline method [20] due to the low quality bounding boxes.

4.2. Comparison With State-of-the-art Methods

Results on Pascal VOC Dataset. We compare our method with two baselines [1, 20] on VOC2007 *test* set [4] in Table 4. For fair comparison, we adopt the same parameter setting for our method and the corresponding baselines. We replace the bounding box regression network by the proposed method to validate their effectiveness. Because Cascade R-CNN [1] is a multi-stage object detector, we replace the regression branch of each stage in Cascade R-CNN with our offset bin class branch in Figure 2. To reduce the number of parameters, the offset bin classification branch here does not include the hierarchical focusing in Figure 3. We set the

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv2 [33]	DarkNet-19	21.6	44.0	19.2	5.0	22.4	35.5
SSD512 [23]	ResNet-101	31.2	50.4	33.3	10.2	34.5	49.8
RetinaNet [21]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
Faster R-CNN [20]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Deformable R-FCN [3]	Inception-ResNet-v2	37.5	58.0	40.8	19.4	40.1	52.5
Mask R-CNN [12]	ResNet-101-FPN	38.2	60.3	41.7	20.1	41.1	50.2
Libra R-CNN [28]	ResNet-101-FPN	40.3	61.3	43.9	22.9	43.1	51.0
KL Loss [14]	ResNet-50-FPN	39.2	57.6	42.5	21.2	41.8	52.5
Grid R-CNN [25]	ResNet-101-FPN	41.5	60.9	44.5	23.3	44.9	53.1
IoU-Net [15]	ResNet-101-FPN	40.6	59.0	-	-	-	-
Cascade R-CNN [1]	ResNet-101-FPN	**42.8**	**62.1**	**46.3**	**23.7**	**45.5**	**55.2**
Faster R-CNN* [20]	ResNet-50-FPN	36.6	58.8	39.6	21.6	39.8	45.0
Our+Faster R-CNN	ResNet-50-FPN	**40.5**	**59.6**	**43.1**	**22.6**	**43.1**	**51.0**
Faster R-CNN* [20]	ResNet-101-FPN	38.8	60.9	42.1	22.6	42.4	48.5
Our+Faster R-CNN	ResNet-101-FPN	**42.5**	**61.7**	**45.4**	**23.9**	**45.6**	**53.8**
Faster R-CNN* [20]	ResNeXt-101-FPN	41.9	**63.9**	45.9	**25.0**	45.3	52.3
Our+Faster R-CNN	ResNeXt-101-FPN	**43.2**	62.7	**46.3**	24.7	**46.4**	**54.8**
Cascade R-CNN* [1]	ResNet-50-FPN	40.7	59.3	44.1	23.1	43.6	51.4
Our+Cascade R-CNN	ResNet-50-FPN	**42.3**	**60.4**	**45.8**	**23.9**	**44.8**	**53.6**
Cascade R-CNN* [1]	ResNet-101-FPN	42.4	61.1	46.1	23.6	45.0	54.4
Our+Cascade R-CNN	ResNet-101-FPN	**44.4**	**62.6**	**48.3**	**24.7**	**47.5**	**56.7**
Cascade R-CNN* [1]	ResNeXt-101-FPN	43.7	62.6	47.5	25.3	46.7	55.5
Our+Cascade R-CNN	ResNeXt-101-FPN	**44.7**	**63.1**	**48.5**	**25.3**	**47.8**	**57.1**

Table 5. Comparison with state-of-the-art methods on MS-COCO *test-dev* set [22]. The symbol * represents our re-implement results based on MMDetection [2].

number of stages of Cascade R-CNN to 2. The IoU thresholds are set to 0.5 and 0.7 in the first and second stages, respectively. These baselines are consistently improved by our methods, which demonstrates the advantage and generality of the proposed methods.

Results on MS-COCO Dataset. Furthermore, we also compare the proposed method with some state-of-the-art object detection methods on the large-scale MS-COCO *test-dev* set [22] in Table 5. It can be observed that the proposed method significantly outperforms these state-of-the-art methods. The proposed offset bin classification method can improve the AP of Faster R-CNN [20, 35] with ResNet-50-FPN, ResNet-101-FPN and ResNeXt-101-FPN by 3.9%, 3.7% and 1.3%, respectively. The results AP can achieve a considerable accuracy 42.3%, 44.4% and 44.7% when we introduce Cascade R-CNN [1] to our method. The superior performance demonstrates the effectiveness of the proposed offset bin classification method.

5. Conclusion

In this paper, we have proposed an offset bin classification network to achieve more accurate object detection. The offset bin labels construction is first used to discretize the continuous offset into several bins. Then the offset bin classification network predicts the probability distribution of offset bins. Furthermore, the expectation-based offset prediction and the hierarchical focusing offset prediction methods are introduced to turn the discretized classification results into more precise offsets. Our method both achieve superior performance on the PASCAL VOC and MS-COCO object detection datasets. The results demonstrate the effectiveness of our proposed method.

Acknowledgement. This work was supported in part by National Natural Science Foundation of China (No. 61525102, 61831005, 61971095 and 61871078).

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[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, Tian-

heng 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 preprint arXiv:1906.07155*, 2019.

[3] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[4] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[5] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

[6] Zhihang Fu, Yaowu Chen, Hongwei Yong, Rongxin Jiang, Lei Zhang, and Xian-Sheng Hua. Foreground gating and background refining network for surveillance object detection. *IEEE Transactions on Image Processing*, 28(12):6077–6090, 2019.

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012.

[8] Spyros Gidaris and Nikos Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *Proceedings of the IEEE international conference on computer vision*, pages 1134–1142, 2015.

[9] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[11] Jicheng Gong, Zhao Zhao, and Nic Li. Improving multi-stage object detection via iterative proposal refinement.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2888–2897, 2019.

[15] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–799, 2018.

[16] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018.

[17] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. Gs3d: An efficient 3d object detection framework for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1019–1028, 2019.

[18] Wei Li, Hongliang Li, Qingbo Wu, Xiaoyu Chen, and King Ngi Ngan. Simultaneously detecting and counting dense vehicles from drone images. *IEEE Transactions on Industrial Electronics*, 66(12):9651–9662, 2019.

[19] Wei Li, Hongliang Li, Qingbo Wu, Fanman Meng, Linfeng Xu, and King Ngi Ngan. Headnet: An end-to-end adaptive relational network for head detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[23] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[24] Wei Liu, Shengcai Liao, and Weidong Hu. Perceiving motion from dynamic memory for vehicle detection in surveillance videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[25] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid r-cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7363–7372, 2019.

[26] Mahyar Najibi, Mohammad Rastegari, and Larry S Davis. G-cnn: an iterative grid based object detector. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2369–2377, 2016.

[27] Alejandro Newell and Jia Deng. Pixels to graphs by associative embedding. In *Advances in neural information processing systems*, pages 2171–2180, 2017.

[28] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 821–830, 2019.

[29] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[30] Heqian Qiu, Hongliang Li, Qingbo Wu, Fanman Meng, King Ngi Ngan, and Hengcan Shi. A2rmnet: Adaptively aspect ratio multi-scale network for object detection in remote sensing images. *Remote Sensing*, 11(13):1594, 2019.

[31] Heqian Qiu, Hongliang Li, Qingbo Wu, Fanman Meng, Linfeng Xu, King N Ngan, and Hengcan Shi. Hierarchical context features embedding for object detection. *IEEE Transactions on Multimedia*, 2020.

[32] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[33] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[34] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[36] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019.

[37] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Dex: Deep expectation of apparent age from a single image. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 10–15, 2015.

[38] Hengcan Shi, Hongliang Li, Fanman Meng, and Qingbo Wu. Key-word-aware network for referring expression image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 38–54, 2018.

[39] Hengcan Shi, Hongliang Li, Fanman Meng, Qingbo Wu, Linfeng Xu, and King Ngi Ngan. Hierarchical parsing net: Semantic scene parsing from global scene to objects. *IEEE Transactions on Multimedia*, 20(10):2670–2682, 2018.

[40] Hengcan Shi, Hongliang Li, Qingbo Wu, Fanman Meng, and King N Ngan. Boosting scene parsing performance via reliable scale prediction. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 492–500, 2018.

[41] Hengcan Shi, Hongliang Li, Qingbo Wu, and Zichen Song. Scene parsing via integrated classification model and variance-based regularization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5307–5316, 2019.

[42] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 292–301, 2018.

[43] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9627–9636, 2019.

[44] Bin Yang, Junjie Yan, Zhen Lei, and Stan Z Li. Craft objects from images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6043–6051, 2016.

[45] Tsun-Yi Yang, Yi-Ting Chen, Yen-Yu Lin, and Yung-Yu Chuang. Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1087–1096, 2019.

[46] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 516–520. ACM, 2016.

[47] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.