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.

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
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 $L_1$ loss [28] further increased the gradient contribution of the samples with small errors to rebalance 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].
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:

\[
\begin{align*}
t_x &= (x - x_a)/w_a, t_y = (y - y_a)/h_a \\
t_w &= \log(w/w_a), t_h = \log(h/h_a) \\
t^*_x &= (x^* - x_a)/w_a, t^*_y = (y^* - y_a)/h_a \\
t^*_w &= \log(w^*/w_a), t^*_h &= \log(h^*/h_a)
\end{align*}
\]  

(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 \(y, y^*\) 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:

\[
L_{\text{loc}} = \sum_{i \in \{x,y,w,h\}} L_{\text{reg}}(t_i - t^*_i) 
\]

(2)

where \(L_{\text{reg}}\) 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 \(\text{Smooth } L_1\) loss function to evade the above problem:

\[
\begin{align*}
\text{Smooth } L_1(x) &= \begin{cases} 
\frac{x^2}{2}, & |x| \leq \beta \\
\frac{\beta x}{2}, & \text{otherwise}
\end{cases} \\
\partial \text{Smooth } L_1 / \partial t_i &= \partial \text{Smooth } L_1 / \partial x \begin{cases} 
\frac{x}{\beta}, & |x| \leq \beta \\
\text{sgn}(x), & \text{otherwise}
\end{cases}
\end{align*}
\]

(3)

(4)

where the deviation \(x = t_i - t^*_i, \beta\) is usually set to 1 in two-stage detectors. \(\text{sgn}\) represents symbolic function. Note that the samples with the offset error larger than \(\beta\) 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 RoI 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 \(L \in \{0, 1, \ldots, n - 1\}\). The representa-
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 $s \in \mathbb{R}^{4n}$, 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:

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

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

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

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

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

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

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:

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

where $L_{\text{cls}}$ denotes the loss for classification of objects, The offset bin classification loss $L_{\text{bin}}$ is used for localization of objects. $\lambda_{\text{bin}}$ is the weight that control the balance among these losses. In this paper, we set $\lambda_{\text{bin}}$ to 1.
predicted offset can be calculated as:

\[ t_i = \mathbb{E}(T^r_i) = \sum_{l=0}^{n-1} (p_{i,l} * t^r_{i,l}) \]  \( (10) \)

where \( T^r_i = \{t^r_{i,0}, t^r_{i,1}, ..., t^r_{i,n-1} \} \) 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 \( \frac{w_{k-1}}{w_k} \). Then, we predict the offset \( t^b_k \) of each stage similar to Section 3.2.3. The final predicted offset can be calculated as:

\[ t_i = \sum_{k=1}^{K} t^b_k \]  \( (11) \)

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 APs, medium scale APs, and large scale APs.

#### 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 \( \lambda_{bin} \) are set to 1. The offset range \( a \) and the number of bins \( n \) are set to 5 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-
offsets. Effectiveness of Different Loss Function for Predicting Offsets. In the high IoU metrics, more accurate object detection, especially performing better in the high IoU metrics. The reason is that the effectiveness of the proposed method in terms of network learning is dominated by some samples with large offsets, and consistently improves AP with different IoU metrics. 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 $\beta$ 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 $\beta$ 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 L1 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.

Setting of Offset Bin Labels. Figure 5 shows 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.

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

<table>
<thead>
<tr>
<th>Stage</th>
<th>AP</th>
<th>$AP_{50}$</th>
<th>$AP_{60}$</th>
<th>$AP_{70}$</th>
<th>$AP_{80}$</th>
<th>$AP_{90}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K = 1$</td>
<td>47.5</td>
<td>74.0</td>
<td>69.0</td>
<td>58.8</td>
<td>44.3</td>
<td>19.6</td>
</tr>
<tr>
<td>$K = 2$</td>
<td>49.0</td>
<td>73.2</td>
<td>68.4</td>
<td>59.0</td>
<td>44.3</td>
<td>19.6</td>
</tr>
<tr>
<td>$K = 3$</td>
<td>48.8</td>
<td>73.3</td>
<td>68.3</td>
<td>58.5</td>
<td>43.6</td>
<td>19.1</td>
</tr>
</tbody>
</table>

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].

![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.](image-url)

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

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

We analyze 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.
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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>AP</th>
<th>AP50</th>
<th>AP60</th>
<th>AP70</th>
<th>AP80</th>
<th>AP90</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster R-CNN* [35]</td>
<td>ResNet-50-FPN</td>
<td>45.0</td>
<td>74.5</td>
<td>69.5</td>
<td>57.6</td>
<td>36.0</td>
<td>6.6</td>
</tr>
<tr>
<td>Our+Faster R-CNN [35]</td>
<td>ResNet-50-FPN</td>
<td>49.0</td>
<td>73.2</td>
<td>68.4</td>
<td>59.0</td>
<td>44.3</td>
<td>19.6</td>
</tr>
<tr>
<td>Faster R-CNN* [35]</td>
<td>ResNet-101-FPN</td>
<td>47.8</td>
<td>75.5</td>
<td>70.6</td>
<td>60.3</td>
<td>41.3</td>
<td>10.5</td>
</tr>
<tr>
<td>Our+Faster R-CNN [35]</td>
<td>ResNet-101-FPN</td>
<td>50.8</td>
<td>74.0</td>
<td>69.5</td>
<td>60.8</td>
<td>47.2</td>
<td>22.5</td>
</tr>
<tr>
<td>Cascade R-CNN* [1]</td>
<td>ResNet-50-FPN</td>
<td>49.5</td>
<td>73.1</td>
<td>69.0</td>
<td>61.0</td>
<td>45.9</td>
<td>18.1</td>
</tr>
<tr>
<td>Our+Cascade R-CNN [1]</td>
<td>ResNet-50-FPN</td>
<td>50.4</td>
<td>73.3</td>
<td>68.9</td>
<td>60.4</td>
<td>46.5</td>
<td>22.2</td>
</tr>
<tr>
<td>Cascade R-CNN* [1]</td>
<td>ResNet-101-FPN</td>
<td>51.0</td>
<td>73.6</td>
<td>69.6</td>
<td>61.9</td>
<td>48.3</td>
<td>21.1</td>
</tr>
<tr>
<td>Our+Cascade R-CNN [1]</td>
<td>ResNet-101-FPN</td>
<td>51.9</td>
<td>73.9</td>
<td>69.8</td>
<td>62.1</td>
<td>48.7</td>
<td>25.0</td>
</tr>
</tbody>
</table>

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 \( \alpha_1 = 3 \) in the first stage. Thus, the end point of offset range \( \alpha_2 \) in second stage and \( \alpha_3 \) 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
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 3.0%, 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.

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

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].

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**


[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-


