



Boundary Information Aggregation and Adaptive Keypoint Combination Enhanced Object Detection

Ping Zhao¹, Dongsheng Yao²(✉), Lijun Sun¹(✉), Jiaqi Fan¹, Panyue Chen¹,
and Zihua Wei¹

¹ Tongji University, Shanghai, China

{zhaoping, sunlijun, 1930795, chenpanyue, zihua.wei}@tongji.edu.cn

² Shanghai Police College, Shanghai, China

Abstract. Keypoint-based methods achieve increasing attention and competitive performance in the field of object detection. In this paper, we propose a new keypoint-based object detection method in order to better locate center keypoints of objects and adaptively combine keypoints to obtain more accurate bounding boxes. Specifically, to better locate center keypoints of objects, we aggregate boundary information by adding the center pooling operation to the original center keypoints prediction branch. The boundary information is the location of object boundary which is more easier to predict than object center. Furthermore, to obtain more accurate bounding boxes, we propose an adaptive keypoint combination algorithm to map all keypoints back to the original image so that the keypoints are combined with less localization errors. Experiments have demonstrated the effectiveness of the our proposed methods.

Keywords: Object detection · Keypoint prediction

1 Introduction

Object detection is a challenging task in computer vision. Keypoint-based object detection methods have achieved increasing attention in the last few years. They regard object detection as a keypoint combination problem. I.e. Detecting different keypoints firstly, then combining keypoints to obtain the final bounding box. In keypoint-based methods, predicting the center keypoint accurately is an important factor to improve the detection performance. However, the widely used center keypoint prediction method focuses on geometric centers of objects, which may fail to contain discriminative information of objects. Take the pedestrian as an example, the geometric center of a pedestrian is usually the middle of the human body. However, the face contains more discriminative information of a pedestrian. Besides, keypoint-based object detection methods predict keypoints instead of bounding boxes, so they need a well-designed combination

algorithm to obtain bounding boxes from keypoints. However, existing combination algorithms are carried out on the feature maps which are usually smaller than the original image, so the location of keypoints is not accurate enough.

Therefore, in this paper, we focus on the following two problems in keypoint-based methods: (1) how to better locate center keypoints and (2) how to adaptively combine keypoints. To better locate center keypoints, we use object boundary information which is the location of object boundary. Specifically, we add an extra center pooling [2] operation to the original center keypoints prediction algorithm. The center pooling operation extracts boundary information of the object, which ensures that the predicted center keypoint contains more discriminative information of the object. Besides, to adaptively combine keypoints, we map all keypoints back to the original image and carry out all operations on the original image so that the keypoint locations are more accurate.

Based on the above two mechanisms, we design a new object detection network, namely BANet (i.e. Network with Boundary information aggregation and Adaptive keypoint combination). Given an input image, our BANet predicts four extreme keypoints (i.e. the extremely top, bottom, left, and right points of an object) and a center keypoint of the object, an adaptive keypoints combination algorithm is then used to combine the predicted keypoints and obtain the final bounding box of the object.

The previous study, ExtremeNet has similar procedure of object detection with our BANet. However, ExtremeNet directly takes the geometric center of an object as the center keypoint, which may not contain the discriminative information of the object. Besides, ExtremeNet combines keypoints based on small-size feature maps, which ignores the combination error caused by the keypoints offsets from feature maps to original images. In comparison, our BANet improves the above two problems at the same time.

Contributions of the proposed BANet can be summarized as follows. First, we make use of object boundary information by adding the center pooling operation to the original center keypoints prediction algorithm for better center keypoint prediction. Second, we propose a new keypoints combination algorithm to adaptively combine different keypoints, so as to improve the accuracy of bounding box prediction. Experiments have demonstrated the effectiveness of our BANet.

2 Related Works

Deep learning based object detection can be roughly divided into two categories according to their network structure: two-stage approaches and one-stage approaches.

Two-stage approaches decompose object detection task into two stages: extracting Region of Interests (RoIs), classifying and regressing RoIs.

R-CNN [5] uses a selective search method [20] to locate the ROIs in the input image. Then, each ROI is adjusted to a fixed size image and input into a CNN model trained on ImageNet to extract features. Finally, the linear SVM classifier is used to predict the target category. Later, SPP [6] and Fast RCNN [4] improve

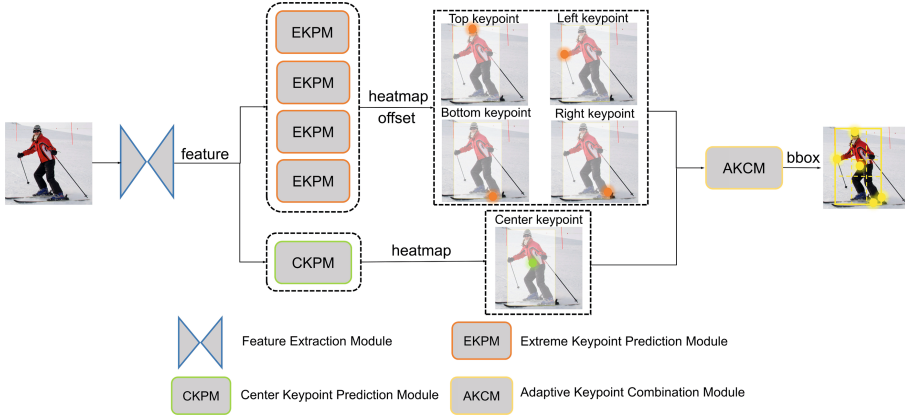


Fig. 1. Our network architecture. An image is first sent to the feature extraction module, then the output features are sent to the detection head composed of four extreme keypoint prediction modules to obtain extreme keypoint heatmaps and offsets, one center keypoint prediction module in parallel to obtain center keypoint heatmap. By combining these heatmaps and offsets, we can obtain the candidate bounding boxes.

R-CNN by designing a special pooling layer that pools each region from feature maps instead. Faster RCNN [17] proposes Region Proposal Network (RPN) to replace selective search method, so that the object detection network can be trained in an end-to-end manner. RPN generate RoIs by regressing anchor boxes. Later, anchor boxes are widely used in many object detection tasks. R-FCN [1] further improves the efficiency of Faster-RCNN by replacing the fully connected prediction head with a fully convolutional prediction head. Many following methods are mainly improved on the network details.

One-stage approaches remove the RoI extracting step and directly get bounding box in a single network.

SSD [11] classifies and regresses by densely placing anchor boxes on multi-scale feature maps. YOLO [15] divides grid on image and makes coordinate prediction directly. DSSD [3] proposes a structure similar to hourglass network to fuse features of different scales. At this time, there is a big performance gap between one-stage approaches and two-stage approaches. The emergence of RetinaNet [9] solves this problem. It proposes FocalLoss to deal unbalanced positive and negative samples. This loss can be well applied to other networks. CornerNet [8] is a completely different one-stage approach. It predicts the positions of different keypoints and combines them instead of getting the bounding box directly. Therefore, CornerNet can be regarded as a bottom-up method. ExtremeNet [23] is similar to CornerNet, but it detects four extreme points instead of two corners.

3 Methods

Preliminaries: Extreme-Points-Based Bounding Box Representation

In the field of object detection, an object is detected using a bounding box, which is usually represent using two points, i.e. the top left point of the bounding box, $(x^{(tl)}, y^{(tl)})$, and the bottom right point of the bounding box, $(x^{(br)}, y^{(br)})$. Instead of using such traditional bounding box representation, in this study, we use four extreme points [14] to represent an object, i.e. the extremely top point of the object, $(x^{(t)}, y^{(t)})$, the extremely left point of the object, $(x^{(l)}, y^{(l)})$, the extremely bottom point of the object, $(x^{(b)}, y^{(b)})$, and the extremely right point of the object, $(x^{(r)}, y^{(r)})$. Obviously, these four extreme points can completely represent the object (by using $(x^{(l)}, y^{(t)}, x^{(r)}, y^{(b)})$) that represented by the traditional bounding box. Compared with the traditional bounding box representation, extreme-points-based bounding box representation has four more values, which bring in more information. In addition, extreme-points-based bounding box representation is easier to obtain than the traditional bounding box representation. When annotating the traditional bounding box, we need to accurately locate up-left corner point $(x^{(tl)}, y^{(tl)})$ and bottom-right corner point $(x^{(br)}, y^{(br)})$ of the box. This process usually requires multiple adjustments.

3.1 Overview of BANet

As shown in Fig. 1, our network consists of four modules, including (a) the feature extraction module, (b) the extreme keypoint prediction module, (c) the center keypoint prediction module, and (d) the adaptive keypoint combination module.

Given an input image I , the feature extraction module extract features $X \in \mathbb{R}^{C \times H \times W}$, where C denotes the number of channels; $H \times W$ denotes the size of the feature maps. Then, based on such features X , four extreme keypoint prediction modules are parallely used to predict four different extreme keypoint heatmaps, $Y_t, Y_b, Y_l, Y_r \in \mathbb{R}^{M \times H \times W}$, where M denotes object categories, and offsets, $O_t, O_b, O_l, O_r \in \mathbb{R}^{2 \times H \times W}$. The center keypoint prediction module uses the extracted features X to predict the center keypoint heatmaps $Y_c \in \mathbb{R}^{M \times H \times W}$. Then, by combining these heatmaps and offsets, the adaptive keypoint combination module obtains the candidate bounding boxes. After getting all candidate bounding boxes, Non-maximum suppression (NMS) is used to determine the final bounding box of the object.

Feature Extraction Module: The feature extraction module is implemented based on the hourglass network [8, 13].

Center Keypoint Prediction Module: As shown in Fig. 2 (a), the center keypoint prediction module uses a boundary information aggregation module followed by a single convolution to predict heatmaps. The boundary information aggregation module consists of a center pooling branch and a single convolution residual connection. Details will be discussed in Sect. 3.3.

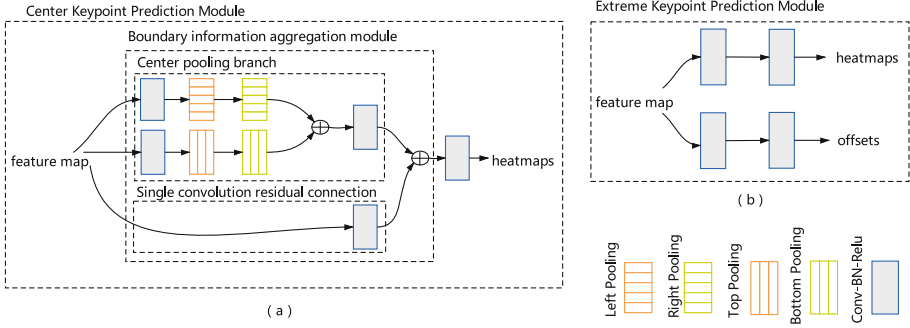


Fig. 2. Center keypoint prediction module (a) uses a boundary information aggregation module followed by a single convolution to predict heatmaps, and two convolutions to predict offsets, while extreme keypoint prediction module (b) uses two parallel double convolutions to predict heatmaps and offsets respectively. Boundary information aggregation module consists of a center pooling branch and a single convolution residual connection.

Extreme Keypoint Prediction Module: As shown in Fig. 2 (b), the extreme keypoint prediction module uses two parallel double convolution branches to predict heatmaps and offsets of extreme keypoints. Our network uses four extreme keypoint prediction modules to predict four extreme keypoints respectively.

Adaptive Keypoint Combination Module: The adaptive keypoint combination module is used to combine different keypoints and get candidate bounding boxes. By using keypoint heatmaps and offsets, the process can be finished accurately on original image. The detailed algorithm will be discussed in Sect. 3.4.

3.2 Boundary Information Aggregation

To better locate center keypoints, we aggregate object boundary information inside the center keypoint prediction module. As mentioned above, directly using the geometric center of the object as the predicted center keypoint may hamper the performance of object detection. Therefore, to predict the center with more discriminative information of the object, we design a module named Boundary Information Aggregation (BIA) to aggregate boundary information of objects. As Fig. 2 shows, the BIA module contains two parallel branches: a center pooling branch and a simple convolution residual connection.

The center pooling [2] branch is composed of pooling operations in four directions, *i.e.* the left pooling, the right pooling, the bottom pooling, and the top pooling. Given an input feature map $x \in \mathbb{R}^{H \times W}$, let $x_{i,:} \in \mathbb{R}^{1 \times W}$ denote the vector of activation scores of the i -th row of the feature map. Then, after the left pooling, we can obtain the following vector of activation scores.

$$Pool_{\text{left}}(x_{i,:}) = [\max_{j=1,\dots,W} x_{ij}, \max_{j=2,\dots,W} x_{ij}, \dots, \max_{j=W} x_{ij}] \quad (1)$$

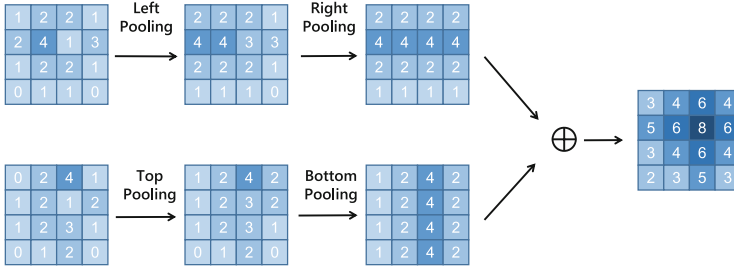


Fig. 3. Example of the center pooling operation. The input feature maps are trained to be sensitive to the left and right boundary (top) and the upper and lower boundary (bottom) respectively, after center pooling, we can aggregate boundary information and get the feature map which is sensitive to the object center.

Similarly, we can obtain the following vector of activation scores after the right pooling.

$$Pool_{\text{right}}(x_{i,:}) = [\max_{j=1} x_{ij}, \max_{j=1,2} x_{ij}, \dots, \max_{j=1,\dots,W} x_{ij}] \quad (2)$$

Let $x_{:,j} \in \mathbb{R}^{H \times 1}$ denote the vector of activation scores of the j -th column of the feature map. Then, after the top pooling, we can obtain the following vector of activation scores.

$$Pool_{\text{top}}(x_{:,j}) = [\max_{i=1,\dots,H} x_{ij}, \max_{i=2,\dots,H} x_{ij}, \dots, \max_{i=H} x_{ij}] \quad (3)$$

Similarly, we can obtain the following vector of activation scores after the bottom pooling.

$$Pool_{\text{bottom}}(x_{:,j}) = [\max_{i=1} x_{ij}, \max_{i=1,2} x_{ij}, \dots, \max_{i=1,\dots,H} x_{ij}] \quad (4)$$

In order to determine whether a location in the feature map is a center keypoint, we need to find the maximum value in its both horizontal and vertical directions and add them together, as shown in Fig. 3. These maximum values correspond to the boundaries of the object in four directions.

However, there are usually many objects in an image, one possible case is that the maximum value of one object in the horizontal direction and the maximum value of another object in the vertical direction are added in a meaningless position after center pooling, that means not even the values of center keypoint are enhanced. Simply using the output of center pooling to predict center keypoint may be harmful, so we add an additional branch which contains a single convolution to represent the feature map before center pooling. We add these two outputs to obtain the final output.

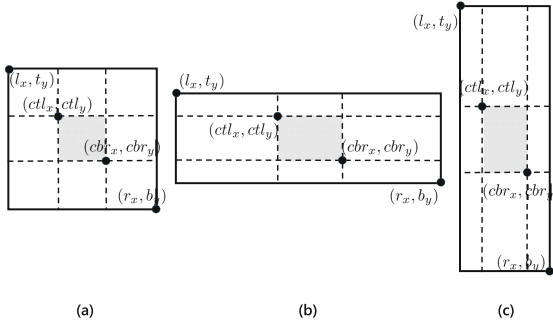


Fig. 4. Some typical central regions. The central region lies in the center of the candidate box (grey part), its length and width are adaptive according to the length and width of the candidate bounding box respectively.

3.3 Adaptive Keypoint Combination

Directly combines keypoints based on the feature map (here we can also call it heatmap) may bring in error, because the feature map is smaller (usually with the size of 128×128) than the size of the original image (usually with the size of 511×511). *I.e.* a pixel of a feature map corresponds to 4×4 pixels of the original image. Therefore, the location of keypoints on feature maps are not accurate. Based on such observation, we propose an adaptive keypoint combination algorithm worked on original images by using keypoint offset information.

The process of our algorithm is as follows: (1) Five keypoint sets, C (center keypoint set), T (top keypoint set), L (left keypoint set), B (bottom keypoint set), and R (right keypoint set), are extracted from the predicted five keypoint heatmaps. These keypoints correspond to the positions whose values are greater than the pre-defined threshold τ_p ; (2) Keypoints in C , T , L , B and R are mapped back to the original image according to their offsets. For example, if the size of feature map and original image are 128×128 and 511×511 respectively, and one keypoint locate at (k_x, k_y) on heatmap has offset (o_x, o_y) , after mapping, its coordinates on original image are $(4(k_x + o_x), 4(k_y + o_y))$. Note that in our model, we don't predict offsets for center keypoints, their offsets are always 0.5. After this step, we can get five new keypoint sets on original image, C_m , T_m , L_m , B_m and R_m ; (3) Let (l, t, r, b) represents a combination by selecting four points from T_m , L_m , B_m and R_m respectively. Candidate boxes $bbox(l_x, t_y, r_x, b_y)$ are obtained by enumerating all combinations of (l, t, r, b) ; (4) For each candidate box $bbox(l_x, t_y, r_x, b_y)$, defines a central region. As long as a center keypoint c from C_m falls in this region, the candidate bounding box $bbox(l_x, t_y, r_x, b_y)$ is accepted. If there are many center keypoints fall in this region, we will choose the center keypoint with the largest value on heatmap. The score of $bbox(l_x, t_y, r_x, b_y)$ is the sum of the values of the four extreme keypoints and the chosen center keypoint.

We can find that how to define central region can significantly affect the detection results. A small central region means the prediction of center keypoint

must be more accurate, which leads to low recall rate and high accuracy, Similarly, a large central region leads to high recall rate and low accuracy. To deal with this trade-off, our definition is: (1) The central region is a rectangle lying in the center of the candidate box; (2) Its height and width are adaptive according to the height and width of the candidate bounding box respectively. Let (ctl_x, ctl_y) and (cbr_x, cbr_y) donate the top-left and bottom-right corners respectively, to obey the first definition, we have:

$$\begin{cases} ctl_x + cbr_x = l_x + r_x \\ ctl_y + cbr_y = t_y + b_y \end{cases} \quad (5)$$

To obey the second definition, we have:

$$\begin{cases} cbr_x - ctl_x = (r_x - l_x) * \left(\frac{6}{r_x - l_x + 18} + \frac{1}{6} \right) \\ cbr_y - ctl_y = (b_y - t_y) * \left(\frac{6}{b_y - t_y + 18} + \frac{1}{6} \right) \end{cases} \quad (6)$$

Figure 4 shows some typical central regions. As we can see, when $r_x - l_x \rightarrow \infty$, we have:

$$cbr_x - ctl_x = \frac{1}{6} (r_x - l_x) \quad (7)$$

When $r_x - l_x \rightarrow 0$, we have:

$$cbr_x - ctl_x = \frac{1}{2} (r_x - l_x) \quad (8)$$

4 Experiments

We conducted experiments on the popular MSCOCO datasets [10]. In order to prove the competitiveness of our model, we compared it with many different one-stage methods. Furthermore, to prove the effectiveness of the proposed methods, we conducted ablation study and error analysis.

Implementation Details: In real implementation, the feature extraction module was implemented based on the hourglass network [8, 13]. During training, the resolution of input image was adjusted to 511×511 , and the resolution of output feature map was 128×128 . The whole network was optimized by adam optimizer with an initial learning rate of $2.5e-4$, learning rate decreased with the number of iterations. In addition, parameters of the hourglass network (*i.e.* the feature extraction module) were preloaded from the official pre-trained CornerNet [8].

Dataset and Evaluation Metrics: We evaluated the proposed model based on the MS COCO dataset [10]. There are 80 kinds of object bounding box annotations in MS COCO dataset. The training process of the model was based on the train2017 dataset, the test and ablation experiments were based on the val2017 dataset. The main evaluation criteria were average precision of fixed recall threshold (AP) and average recall of fixed box number (AR).

Table 1. Comparisons with other one-stage object detectors.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
YOLOv2 [16]	DarkNet-19	21.6	44.0	19.2	5.0	22.4	35.5	20.7	31.6	33.3	9.8	36.5	54.4
DSOD300 [18]	DS/64-192-48-1	29.3	47.3	30.6	9.4	31.5	47.0	27.3	40.7	43.0	16.7	47.1	65.0
GRP-DSOD320 [19]	DS/64-192-48-1	30.0	47.9	31.8	10.9	33.6	46.3	28.0	42.1	44.5	18.8	49.1	65.0
SSD513 [11]	ResNet-101	31.2	50.4	33.3	10.2	34.5	49.8	28.3	42.1	44.4	17.6	49.2	65.8
DSSD513 [3]	ResNet-101	33.2	53.3	35.2	13.0	35.4	51.1	28.9	43.5	46.2	21.8	49.1	66.4
RefineDet512 [22]	ResNet-101	36.4	57.5	39.5	16.6	39.9	51.4	-	-	-	-	-	-
RetinaNet800 [9]	ResNet-101	39.1	59.1	42.3	21.8	42.7	50.2	-	-	-	-	-	-
ExtremeNet511 [23]	Hourglass-104	40.3	55.1	43.7	21.6	44.0	56.1	32.0	50.2	53.1	30.7	57.8	69.9
CornerNet511 [8]	Hourglass-104	40.6	56.4	43.2	19.1	42.8	54.3	35.3	54.7	59.4	37.4	62.4	77.2
BANet	Hourglass-104	40.7	56.5	43.6	22.4	44.9	55.8	32.5	52.1	55.4	31.5	60.2	74.8

Table 2. Ablation study and error analysis. Here bia means boundary information aggregation, akc means adaptive keypoint combination, gt ex means using ground truth extreme keypoints, gt ct means using ground truth center keypoints.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
BANet	40.7	56.5	43.6	22.4	44.9	55.8	32.5	52.1	55.4	31.5	60.2	74.8
BANet without bia	39.4	55.2	42.0	21.7	42.6	54.1	31.8	51.2	54.9	31.7	59.2	73.8
BANet without akc	41.0	56.2	44.2	21.9	44.8	56.9	32.2	51.1	54.1	31.3	58.6	72.0
BANet without bia & akc	39.9	54.6	42.9	20.8	43.7	54.0	31.6	49.8	52.8	30.5	57.5	69.0
BANet without bia + gt ex	52.9	66.1	56.7	34.7	60.2	64.3	36.4	56.0	58.2	39.3	65.3	68.2
BANet + gt ex	54.0	67.5	57.7	35.6	61.1	67.1	36.9	57.1	59.5	40.7	66.3	71.1
BANet without bia + gt ct	50.7	64.1	56.0	26.9	55.7	70.0	36.7	55.1	57.2	32.1	62.5	77.4
BANet + gt ct	51.3	64.8	56.7	27.3	56.1	71.5	36.9	55.7	57.7	32.3	62.8	78.8

4.1 Comparisons with Other One-Stage Object Detectors

Table 1 showed the comparison with several popular one-stage object detectors on the MS COCO val2017 dataset. Compared with the baseline ExtremeNet [23], the proposed BANet achieved a comparative improvement. BANet reported a testing AP of 40.7%, an improvement of 0.4% over 40.3%, and a test AR_{100} of 55.4%, an improvement of 2.3% over 53.1%, achieved by ExtremeNet under the same setting. Moreover, BANet surpassed CornerNet [8] in AP . CornerNet used associative embeddings to help combine different keypoints, while BANet used a pure geometric method. These results firmly demonstrate the effectiveness of BANet.

4.2 Ablation Study and Error Analyse

Boundary Information Aggregation: After adding boundary information aggregation module, our network reached 41.0% AP and 54.1% AR , which was 1.1% and 1.3% higher than the baseline. This proved the effectiveness of our proposed module. See Table 2 for more details. We can find that boundary information aggregation had the greatest improvement on large objects, it was because keypoint-based methods were very strict in detecting large objects. In their post-processing, no matter large objects, medium objects and small objects, the object



Fig. 5. Visualization of detection results. Images showed detected extreme keypoints heatmaps (left), center keypoint heatmaps (middle), and detection results (right) respectively.

center only occupied a pixel on the feature map, so for large objects, its location should be more accurate.

Adaptive Keypoint Combination: Results were shown in Table 2. We can see that the adaptive keypoint combination algorithm had little change in AP , but because of the use of offset information and the definition of central region, the model greatly exceeded baseline in AR , especially for large objects, which indicated that combination algorithms on feature maps were very strict in detecting the center keypoint of large objects. After changing, the acceptable region of candidate boxes had been expanded, especially large objects, so their recall rates had been greatly improved.

Error Analysis: In order to further analyze the influence of boundary information aggregation on different parts of the whole model, we used ground truth extreme keypoint heatmaps and ground truth center keypoint heatmap to replace

the corresponding predicted heatmaps, and compared the gap of the other part. The specific comparative information was shown in Table 2.

Discussions: Through the experimental results, we can get the following two conclusions: (1) After adding boundary information aggregation, the model had a great improvement in the prediction of center keypoints. This further shows that proposed module can obtain boundary information and locate center keypoints more accurately. (2) Without modifying other parts of the model, boundary information aggregation also slightly improved the prediction performance of extreme keypoints, which showed that the model paid more attention to the training of extreme keypoints (the proportion of loss was larger) due to the reduction of the difficulty of center keypoint prediction.

4.3 Visualization of Detection Results

Images in Fig. 5 visualized detected heatmaps and final results. In order to show the detected keypoints, we added a mask to the original image and highlight different categories of keypoints with different colors on the heatmaps. These qualitative results proved that our network can find keypoints and combine them to obtain the final bounding boxes accurately.

5 Conclusions

In this paper, we have proposed a new keypoint-based object detection network, namely BANet. Specifically, we propose the boundary information aggregation algorithm to achieve better center keypoint location. Besides, we propose the adaptive keypoint combination algorithm to obtain more accurate keypoints combination. Experiments have demonstrated the effectiveness of our BANet.

References

1. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. arXiv preprint [arXiv:1605.06409](https://arxiv.org/abs/1605.06409) (2016)
2. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: CenterNet: keypoint triplets for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6569–6578 (2019)
3. Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: DSSD: deconvolutional single shot detector. arXiv preprint [arXiv:1701.06659](https://arxiv.org/abs/1701.06659) (2017)
4. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
5. 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)
6. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(9), 1904–1916 (2015)

7. Jiao, L., et al.: A survey of deep learning-based object detection. *IEEE Access* **7**, 128837–128868 (2019)
8. Law, H., Deng, J.: CornerNet: detecting objects as paired keypoints. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 734–750 (2018)
9. 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)
10. Lin, T.Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
11. Liu, W., et al.: SSD: single shot MultiBox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
12. Newell, A., Huang, Z., Deng, J.: Associative embedding: end-to-end learning for joint detection and grouping. arXiv preprint [arXiv:1611.05424](https://arxiv.org/abs/1611.05424) (2016)
13. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 483–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_29
14. Papadopoulos, D.P., Uijlings, J.R., Keller, F., Ferrari, V.: Extreme clicking for efficient object annotation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4930–4939 (2017)
15. 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)
16. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271 (2017)
17. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. arXiv preprint [arXiv:1506.01497](https://arxiv.org/abs/1506.01497) (2015)
18. Shen, Z., Liu, Z., Li, J., Jiang, Y.G., Chen, Y., Xue, X.: DSOD: learning deeply supervised object detectors from scratch. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1919–1927 (2017)
19. Shen, Z., et al.: Learning object detectors from scratch with gated recurrent feature pyramids. arXiv preprint [arXiv:1712.00886](https://arxiv.org/abs/1712.00886) (2017)
20. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. *Int. J. Comput. Vis.* **104**(2), 154–171 (2013)
21. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 466–481 (2018)
22. Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z.: Single-shot refinement neural network for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4203–4212 (2018)
23. Zhou, X., Zhuo, J., Krahenbuhl, P.: Bottom-up object detection by grouping extreme and center points. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 850–859 (2019)
24. Zou, Z., Shi, Z., Guo, Y., Ye, J.: Object detection in 20 years: a survey. arXiv preprint [arXiv:1905.05055](https://arxiv.org/abs/1905.05055) (2019)