ObjectBox: From Centers to Boxes for Anchor-Free Object Detection (Supplementary Material)

Mohsen Zand, Ali Etemad, and Michael Greenspan

Dept. of Electrical and Computer Engineering, Ingenuity Labs Research Institute
Queen's University, Kingston, Ontario, Canada

S.1 Overview

We provide additional experiments to further explore the robustness of our method. Experimental results on the PASCAL VOC 2012 [S.1] are represented in Sec. S.2. In Sec. S.3, we investigate correlations between the object size and the scale at which it is detected. We also utilize our IoU loss in other detectors and report the results in Sec. S.4. Inference details are also discussed in Sec. S.5. Finally, we qualitatively show some ObjectBox results in Sec. S.6.

S.2 PASCAL VOC 2012

To demonstrate the effectiveness of our method on different object categories in a subtle way, we perform another experiment on the PASCAL VOC 2012 dataset. We trained the network under the same settings as we performed on MS-COCO dataset. Notably, our method does not need to set dataset-dependent hyperparameters like anchor boxes. The results are shown in Table S.1. ObjectBox outperforms the other methods, achieving a higher AP score on 13 of the 20 object classes. Overall, ObjectBox achieves an mAP of 83.7%, which is +2.4% higher than the next best performing method. It can be observed that ObjectBox works relatively well in both small object and large object classes. For example, it achieves 92.1% in the class *'plane'* and 93.3% in the class *'car'*. It can be observed that ObjectBox is either the best or the second-best method in terms of AP score in all categories except 'cat', 'dog', and 'bike'. This is probably due to the use of YOLO's Darknet backbone, as YOLOv2 similarly does not work well in these categories.

S.3 Multiscale Prediction

We investigate correlations between the object size and the scale at which it is detected. As shown in Table S.2, we consider predictions per individual scales, observing that larger objects are better detected at coarser scales, and smaller objects are better detected at finer scales, despite being trained without the bias in defining the positive samples. Each scale level still contributes to the

Table S.1. Detection results on the PASCAL VOC 2012 dataset. F-RCNN denotes Faster R-CNN. The bold and underlined numbers respectively indicate the best and second best results in each column

Method	**mAP**	plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	bike	person	plant	sheep	sofa	train	tv
F-RCNN[S.8]	73.8	86.5	81.6	77.2	58.0	51.0	78.6	76.6	93.2	48.6	80.4	59.0	92.1	85.3	84.8	80.7	48.1	77.3	66.5	84.7	65.6
SSD513[S.4]	79.4	90.7	87.3	78.3	66.3	56.5	84.1	83.7	**94.2**	62.9	84.5	66.3	**92.9**	88.6	87.9	85.7	55.1	83.6	74.3	88.2	76.8
YOLOv2[S.6]	73.4	86.3	82.0	74.8	59.2	51.8	79.8	76.5	90.6	52.1	78.2	58.5	89.3	82.5	83.4	81.3	49.1	77.2	62.4	83.8	68.7
Retina[S.3]	67.7	80.4	74.0	73.4	53.5	49.7	73.0	71.2	88.2	45.8	69.7	50.6	87.1	74.0	76.8	78.9	45.6	69.1	51.3	77.2	65.0
ASSD513[S.10]	81.3	**92.1**	89.2	**82.5**	71.5	60.4	85.5	84.8	93.9	63.7	**88.6**	67.4	92.6	90.2	**89.0**	86.5	**60.4**	**88.2**	73.4	88.6	77.0
ObjectBox	**83.7**	**92.1**	**92.2**	80.5	**74.0**	**77.4**	**92.1**	**93.3**	89.5	**68.2**	85.3	**75.7**	87.3	**90.6**	86.6	**87.9**	60.0	84.7	**77.6**	**91.3**	**85.4**

prediction of objects at other scales. This shows that the learning can be better because all objects are being learned at all possible scales.

Table S.2. Predictions per scale level on the MS-COCO dataset

	Avg. Precision, IoU			Avg. Precision, Area			Avg. Recall, Area		
Scale level	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_S	AR_M	AR_L
0, s={8}	24.9	44.9	32.1	25.5	11.7	20.5	38.1	18.4	37.3
1, s={16}	35.7	56.4	37.7	24.3	49.2	33.6	37.6	64.8	46.3
2, s={32}	42.7	61.6	46.0	22.0	48.0	56.1	30.9	63.2	75.6
all, s={8,16,32}	46.8	65.9	49.5	26.8	49.5	57.6	39.4	65.2	77.0

S.4 IoU Loss

In this work, we propose an IoU-based loss tailored for our object detection method. Recall that our loss is applied to our regression targets $\{L, T, R, B\}$, which are distance values from the corners of the object center cells to the four sides of the bounding box. There are other methods that use similar targets for box regression. For example, FCOS [S.9] defines $\{l, t, r, b\}$ as regression targets as distances from each positive location to the bounding box boundaries. The positive locations are selected based on the scale ranges defined for each pyramid level. ATSS [S.11] uses the same targets but with a different strategy for positive sample selection. It specifically uses statistical characteristics of objects as the IoU threshold to adaptively select enough positives for each object from appropriate pyramid levels.

Regardless of their sample selection strategies, both FCOS and ATSS use IoU-based losses, such as the GIoU loss function, for bounding box regression. In our experiments, we replaced their regression losses with our tailored IoU loss, and trained their models with the same settings as the original ones. The results are reported in Table S.3. It can be seen that our loss consistently improves detection performance. Our loss improves FCOS by +0.2% on AP, +0.6% on AP_{50}, +0.1 on AP_S, +0.9 on AP_M, and +1.2 on AP_L. Similarly, it achieves a higher performance in ATSS by +0.4% on AP, +1.1% on AP_{50}, +0.1% on AP_{75},

Table S.3. Relative performance of our loss function and GIoU loss, when applied to ObjectBox, FCOS and ATSS

Method	Backbone	Loss	Avg. Precision, IoU			Avg. Precision, Area		
			AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
FCOS [S.9]	ResNeXt-101	GIoU	42.1	62.1	45.2	25.6	44.9	52.0
		ours	42.3	62.7	45.2	25.7	45.8	53.2
ATSS [S.11]	ResNet-101	GIoU	43.6	62.1	47.4	26.1	47.0	53.6
		ours	44.0	63.2	47.5	26.2	48.4	54.2
ObjectBox	ResNet-101	GIoU	44.9	63.6	47.3	25.6	48.5	55.9
		ours	46.1	65.0	48.3	26.0	48.7	57.3
	CSPDarknet	GIoU	45.7	64.2	48.0	26.1	48.9	57.0
		ours	46.8	65.9	49.5	26.8	49.5	57.6

+0.1 on AP_S, +1.4 on AP_M, and +0.6 on AP_L. Note that our loss function is directly applied to the network outputs. It therefore keeps the box integrity based on the model regression targets, and scores the overlapping areas in all four directions.

In Sec. 4.3, we showed the effectiveness of our loss function for box regression, where replacing it with other losses drastically decreased performance. It was however coupled with the impact of regression location from only one center location. To further investigate the necessity of this loss in our method, we keep the best settings from our ablation study in Sec. 4.3, and only replace our loss with the GIoU loss as used in FCOS and ATSS. Particularly, we use our proposed regression targets ($\{L, T, R, B\}$), augmented centers (the best results in Table 3 part A), one prediction per scale level, and no scale range constraints. As shown in Table S.3, ObjectBox with a ResNet-101 backbone clearly benefits from the new IoU loss since it obtains a higher performance by +1.2% on AP, +1.4% on AP_{50}, +1.0% on AP_{75}, +0.4 on AP_S, +0.2 on AP_M, and +1.4 on AP_L, when compared with GIoU loss. The performance boost on ObjectBox with a CSPDarknet backbone is also evident as our loss improves the performance by +1.1% on AP, +1.7% on AP_{50}, +1.5 on AP_{75}, +0.7 on AP_S, +0.6 on AP_M, and +0.6 on AP_L. These relative improvements indicate that the box IoU loss in our method practically helps to align the bounding boxes more precisely.

S.5 Inference

Our method does not impose any additional costs to the inference stage. Given an input image, ObjetcBox predicts an objectness (confidence) score, m classification scores, and four regression values ($\{L, T, R, B\}$) for each feature map location, where m denotes the number of class labels. Therefore, the network output is of size $3 \times \frac{W}{s_i} \times \frac{H}{s_i} \times (m + 5)$, where $s_i \in \{8, 16, 32\}$. The predictions at all scale levels are sorted based on their confidence scores. They are then refined sequentially based on a threshold value (we set it as 0.001) until all candidates are investigated. To reduce redundancy in the box prediction, we use

Table S.4. Inference speed comparison

Method	Backbone	# params	FPS	AP
SSD513 [S.4]	ResNet-101	57 M	43	31.2
Faster R-CNN w/ FPN [S.2]	ResNet-101	42 M	26	36.2
YOLOv3 [S.7]	DarkNet-53	65 M	20	33.0
FCOS [S.9]	ResNeXt-101	32 M	50	42.1
ATSS [S.11]	ResNet-101	32 M	50	43.6
ObjectBox	ResNet-101	30 M	70	46.1
ObjectBox	CSPDarknet	86 M	**120**	**46.8**

non-maximum suppression (NMS) [S.5] on the predicted boxes based on their classification error. We use an NMS threshold 0.6 and obtain the final results by keeping the highest quality bounding boxes and eliminating the others.

We used the same sizes of input images as in training, and evaluated the inference speed on a single Titan RTX GPU by measuring the end-to-end inference time. We selected different anchor-based and anchor-free methods as comparisons, including SSD513 [S.4], Faster R-CNN with FPN [S.2], YOLOv3 [S.7], FCOS [S.9], and ATSS [S.11]. As shown in Table S.4, the average inference speed of ObjectBox with the ResNet-101 backbone is 70 FPS. Meanwhile, using the CSPDarknet backbone can improve the inference speed by 50 FPS, achieving 120 FPS. ObjectBox is significantly faster than other detectors, while having a larger number of parameters (86 M). For instance, the detection speed of ObjectBox is more than two times higher than that of FCOS (50 FPS). Even with the same ResNet-101 backbone, ObjectBox outperforms the ATSS frame rate by 40%, i.e. from 50 FPS for ATSS to 70 FPS for ObjectBox.

The superior time performance of ObjectBox is mainly due to its smaller detection head, and that it considesrs only object central locations for box regression. More specifically, the number of predictions per scale level is just one in ObjectBox, while it is equal to the number of anchors (usually > 1) in anchor-based detectors. It also filters out all non-center locations by using the confidence score, which undoubtedly reduces the NMS computational load. By relaxing the scale range constraints, ObjectBox redefines positive and negative training samples without incurring any additional overheads. It is therefore quite efficient, while achieving the state-of-the-art performance.

S.6 Qualitative results

In Figure S.1, we show some detection examples on the MS-COCO test-dev dataset. It can be seen that our method is able to successfully detect objects with different sizes and different scene types, with severely overlapping boxes.

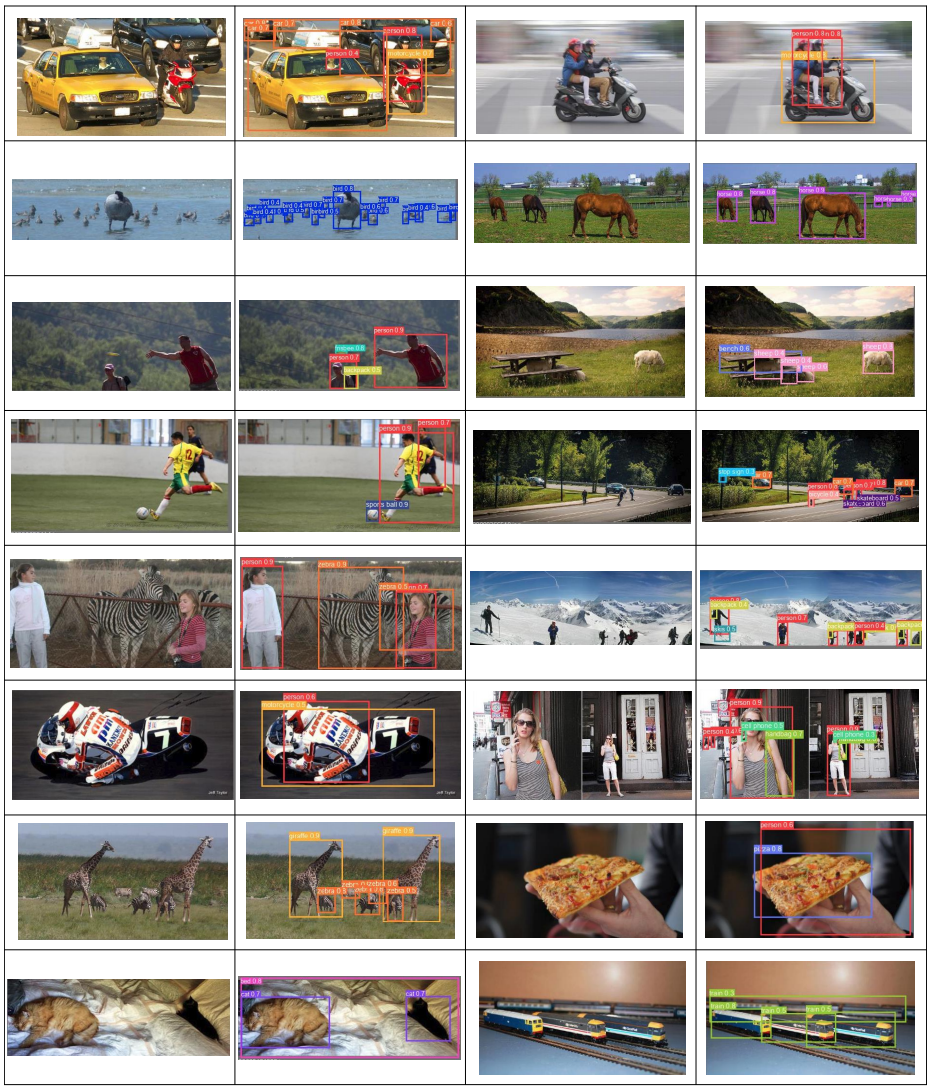

Fig. S.1. Detection examples of applying ObjectBox on COCO test-dev

References

[S.1] Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International journal of computer vision **111**(1), 98–136 (2015)

[S.2] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)

[S.3] 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)

[S.4] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)

[S.5] Neubeck, A., Van Gool, L.: Efficient non-maximum suppression. In: 18th International Conference on Pattern Recognition (ICPR'06). vol. 3, pp. 850–855. IEEE (2006)

[S.6] Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7263–7271 (2017)

[S.7] Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)

[S.8] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)

[S.9] Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9627–9636 (2019)

[S.10] Yi, J., Wu, P., Metaxas, D.N.: Assd: Attentive single shot multibox detector. Computer Vision and Image Understanding **189**, 102827 (2019)

[S.11] Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9759–9768 (2020)