Objects365: A Large-scale, High-quality Dataset for Object Detection

Shuai Shao*, Zeming Li*, Tianyuan Zhang*, Chao Peng*
Gang Yu†, Xiangyu Zhang, Jing Li, Jian Sun
Megvii Technology
{shaoshuai, lizeming, zhangtianyuan, pengchao, yugang, zhangxiangyu, lijing, sunjian}@megvii.com

Abstract

In this paper, we introduce a new large-scale object detection dataset, Objects365, which has 365 object categories over 600K training images. More than 10 million, high-quality bounding boxes are manually labeled through a three-step, carefully designed annotation pipeline. It is the largest object detection dataset (with full annotation) so far and establishes a more challenging benchmark for the community. Objects365 can serve as a better feature learning dataset for localization-sensitive tasks like object detection and semantic segmentation. The Objects365 pre-trained models significantly outperform ImageNet pre-trained models with 5.6 points gain (42 vs 36.4) based on the standard setting of 90K iterations on COCO benchmark. Even compared with much long training time like 540K iterations, our Objects365 pretrained model with 90K iterations still have 2.7 points gain (42 vs 39.3). Meanwhile, the finetuning time can be greatly reduced (up to 10 times) when reaching the same accuracy. Better generalization ability of Object365 has also been verified on CityPersons, VOC segmentation, and ADE tasks. The dataset as well as the pretrained-models have been released at `www.objects365.org`.

1. Introduction

Object detection is a fundamental task in computer vision. PASCAL VOC [8] and COCO [24], have contributed greatly to rapid advances of object detection. From traditional approaches like DPM [9] to deep-learning based approaches like R-CNN [13] and FPN [22], the above two datasets serve as “golden” benchmarks to evaluate algorithms and boost research progresses. In this paper, we move a step further to introduce a new large-scale, high-quality object detection dataset, Objects365, which focuses on three aspects: *scale*, *quality*, and *generalization*.

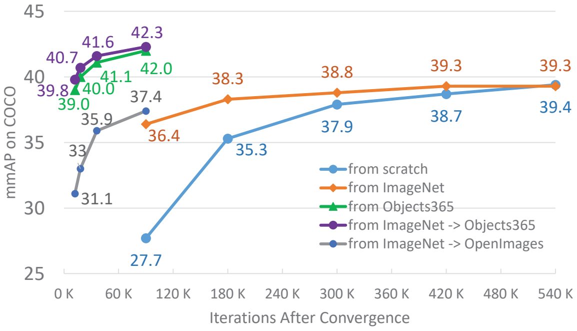

Figure 1. Results of finetuning COCO from ImageNet vs Objects365. With a small number of iterations, 90K, for training, our Objects365 pre-trained model (green curve) can significantly outperform the algorithm finetuned from ImageNet (orange curve), even with much longer training time, e.g., 540K iterations. All results use a FPN based Resnet50 backbone.

Scale. Objects365 is significantly larger than the existing object detection benchmarks like PASCAL and COCO. It contains 365 categories, $638K$ images, and $10,101K$ bounding boxes. We compare our dataset with existing object detection benchmarks with full annotations in Table 1. Our dataset contains 5× more images, 4× more categories, and 10× more boxes than COCO [24]. It can serve as a more challenging benchmark for the detection community.

Quality. In addition to the size, annotation quality is of great importance when building a dataset. To ensure quality, we divide the annotation pipeline into three steps, which can significantly reduce the job requirement for the annotators. Besides the annotators, we also include inspectors and examiners to review the quality of the annotations. To reduce ambiguities during the annotation process, we apply two consistency rules. This annotation pipeline ensures that we obtain high-quality annotation with high efficiency.

Generalization. The feature learned from Objects365 is superior for many localization-sensitive tasks like object detection and semantic segmentation. Conventionally, ImageNet [5] pre-trained basenets like Resnet [17] are widely employed as a backbone for the object de-

[*] indicates equal contribution.
[†] Corresponding Author.

tection/segmentation algorithms. However, there are two issues related to ImageNet pretraining. On one hand, the feature learned on ImageNet classification task is less localization-sensitive. On the other hand, only the backbone part are pre-trained but the head part is initialized with the random weights.

Our Objects365 dataset directly addresses the above two issues and provides a better alternative for feature learning. As shown in Figure 1, the Objects365 pre-trained features can significantly outperform the counterparts based on ImageNet, even the one with sufficient longer training time (540K iterations) as discussed in [15]. Moreover, using Objects365 feature, we can obtain comparable results with one order-of-magnitude less training time.

2. Related Work

General Object Detection Object detection task has been a fundamental research topic for a while. DPM [9] is one of the most famous object detection algorithms to be used before the introduction of deep learning techniques. R-CNN [13] is one of the first work to integrate the convolutional neural network for object detection. Later, with the rapid development of convolutional neural network, most object detection algorithms started to utilize deep learning techniques. Roughly, we can divide the existing object detection algorithms into two categories: single-stage detector and two-stage detector. The main difference lies on whether to pool the feature maps for a second stage. SSD [25], DSSD [10] and YOLO series [27, 28, 29] are some widely used single-stage detector with efficient speed. RetinaNet [23] is introduced with strong performance even compared with the two-stage detector. For the two-stage detector, the early work like Fast R-CNN [12], Faster R-CNN [30], R-FCN [3], try to speed up the algorithms. In FPN [22] and Mask R-CNN [16], feature pyramid structure and ROI-Align are proposed to boost the performance. Deformable ConvNet [4, 39], Soft Sampling [34], SNIP [32], SNIPER [33], and Cascade R-CNN [1], DetNet [21] are introduced to further improve the performance.

Large-scale Detection Dataset The large-scale dataset is an important reason for the continuous improvement of the object detection algorithms, especially for deep learning based techniques. From early datasets like ImageNet [5], VOC [8], to the recent benchmarks like COCO [24], they all play an important role in the image classification and object detection community. In Table 1, we give statistics of the existing object detection benchmarks together with our Objects365 benchmark. Our Objects365 dataset has around 60 times images larger than PASCAL VOC and 5 times larger than COCO. Compared with the ImageNet DET dataset [5], our dataset has a larger number of boxes per image, with 15.8 vs 1.1 (2.3 for the Dense set).

Besides the general object detection datasets, there are also a lot of other detection benchmarks like face detection [19, 35], pedestrian detection [7, 36, 31], and human/vehicle detection for the autonomous driving [11, 2], all of which play an important role in the detection community.

3. Objects365 Dataset

In this section, we present the details on the collection, annotation, statistics, and quality of the dataset respectively.

3.1. Data Collection

3.1.1 Data Source

To make the image sources more diverse, we collect images mainly from Flicker [1]. All the images conform to licensing for research purposes. Sample images can be found on our website[2].

3.1.2 Object Categories

Based on the collected images, we first select eleven super-categories which are common and diverse to cover most object instances. They are: *human and related accessories*, *living room*, *clothes*, *kitchen*, *instrument*, *transportation*, *bathroom*, *electronics*, *food (vegetables)*, *office supplies*, and *animal*. Based on the super-categories, we further propose 442 categories which widely exists in our daily lives. As some of the object categories are rarely found, we first annotate all 442 categories in the first 100K images and then select the most frequent 365 object categories as our target objects. Also, to be compatible with the existing object detection benchmarks, the 365 categories include the categories defined in PASCAL VOC [8] and COCO [24] benchmarks.

3.1.3 Non-Iconic Images

As our Objects365 dataset focuses on object detection, we eliminate those images which are only suitable for image classification. For example, the image only contains one object instance around the image center. This filtering process was first adopted in COCO [24].

3.2. Annotation

As there are a large number of images and object categories, a good annotation process is of great importance to ensure high quality and efficiency.

Dataset	Images	Boxes	Categories	Boxes/img	Fully Annotated
Pascal VOC	11.5k	27k	20	2.4	Yes
ImageNet All	477k	534k	200	1.1	Yes
ImageNet Dense	80k	186k	200	2.3	Yes
COCO	123k	896k	80	7.3	Yes
OpenImages	1,515k	14,815k	600	9.8	Partial
Objects365	**638k**	**10,101k**	**365**	**15.8**	Yes

Table 1. Comparison of the dataset statistics with existing object detection benchmarks. The table includes statistics for training and validation sets.

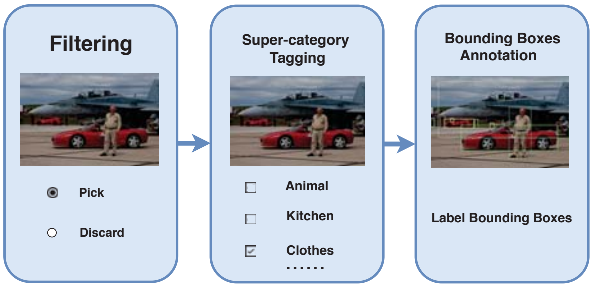

Figure 2. Our annotation pipeline for Objects365.

3.2.1 Annotation Pipeline

It is almost impossible for an annotator to remember and annotate all 365 categories. Also, a small number of images should be rejected due to the iconic images or the images without 365 object categories. Motivated by existing datasets like ImageNet [5] and COCO [24] as well as the discussion of scalable multi-class annotation in [6], we design our annotation pipeline as the following three steps. The first step performs a two-class classification. If the image is non-iconic or contains at least one object instance in the eleven super-categories, it will be passed to the next step. In the second step, the image-level tags with the eleven super-categories will be labeled. An image may be labeled with more than one tag. In the third step, one annotator will be assigned to label the object instances in one specific super-category. All object instances belonging to the super-category should be labeled with a bounding box together with an object name. An illustration of our annotation pipeline is in Figure 2.

Based on the proposed annotation pipeline, each annotator only needs to be familiar with the object categories in one super-category rather than all 365 object categories. This will not only improve annotation efficiency but also boost annotation quality.

3.2.2 Annotation Team

We divide all the team members (hired as vendors) into three groups: annotators, inspectors, examiners. All the images are first annotated by the annotators, and then checked by the inspectors. By accumulating around 100 labeled (and checked) images as a job, we involve the examiner to further review the quality of the annotation. The job will be rejected if any of the steps exists annotation error.

Annotator The annotators need to perform the annotation which involves all three steps in the annotation pipeline shown in Figure 2. Each annotator will be assigned one and only one annotation task. Before starting the annotation, they should take a course and pass an examination to be qualified for the annotation.

Inspector The work of inspector is to examine all the annotated images labeled by the annotators. If an annotation error is found in one image, the image will be rejected and annotated again by the same annotator. This refining step can greatly prevent the annotator from making the same mistakes in the following annotation process.

Examiner Examiners are usually geographically isolated from the annotators and inspectors to make a fair judgment. The job, which usually contains around 100 images verified by the inspectors, will be reviewed again by the examiner. If an annotation error is found in one image, the whole job which includes the mislabeled image will be rejected. The rejection rate of the jobs is one of the key factors to determine the income for the annotators and inspectors.

3.2.3 Annotation Process Consistency

Due to the large scale of the dataset, there will be a lot of annotators involved in this project. Without consistent definition and rules for the annotation, the labeling for the same image will provide different annotation results. To reduce the ambiguities during the annotation process, we define a number of rules. Two important rules are: classification rule and bounding-box rule.

Classification Rule It defines a clear priority order with **function**-first principle for the ambiguity case in labeling. For instance, in the left Figure 3, the object can be considered as either “tap” or “teapot”. Based on our classification rule, we use the function-first principle and the object will be labeled as “tap” in this case.

[1] https://www.flickr.com/
[2] https://www.objects365.org

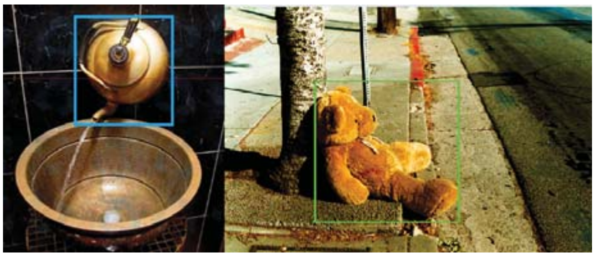

Figure 3. An example showing our function-first consistency rule. In the left figure, the object in the blue bounding-box will be annotated as “tap” rather than “teapot” while the green bounding box will be annotated as “toy” instead of “bear” in the right figure.

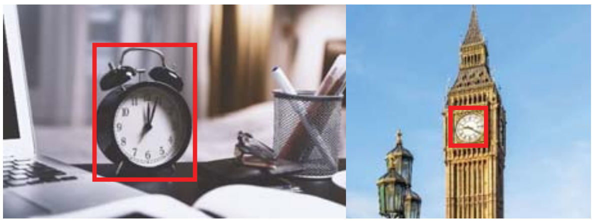

Figure 4. An example of our bounding-box consistency rule.

	Train	Validation	Test
Images	600k	38k	100k
Boxes	9623k	479k	1700k

Table 2. The split of our Objects365 benchmark.

Bounding Box Rules Due to the diversity of annotators, the annotation for the bounding-boxes might be inconsistent sometimes. We define the following rule when ambiguities exist for the bounding-box. The annotator is required to cover the largest bounding box which would not lead to the ambiguities of defining the object category. For example, we need to include the decoration part of the clock in the left figure of Figure 4 as the decoration part belongs to the clock and would not lead to misunderstanding of the object category. For the right figure of Figure 4, the annotators are required to label the small bounding box because the external area of the clock will lead to another category as “tower”.

3.3. Statistics

Based on our proposed annotation pipeline, around 740K images are annotated in our Objects365 dataset. A split of train, validation, test set is in Table 2. There are 600K images for training, 38K images for validation, and another 100K images for testing.

To delve into the details of our Objects365, we first provide the statistics of the number of object categories per image. According to Figure 5, Objects365 is more dense and diverse than VOC [8], COCO [24], and OpenImages [20]. Quantitatively, our Objects365 has 5 categories (on average) per image, which is more dense compared with the other object detection benchmarks like VOC and COCO. Due to the partial annotation used in OpenImages, the category number/image of OpenImages is lower than COCO and Objects365 even though there are 600 categories defined in OpenImages. Moreover, based on the curve in Figure 5, the variance of the number of object categories per image from our Objects365 is significantly larger than the existing benchmarks, which shows the diverse nature of our Objects365 dataset.

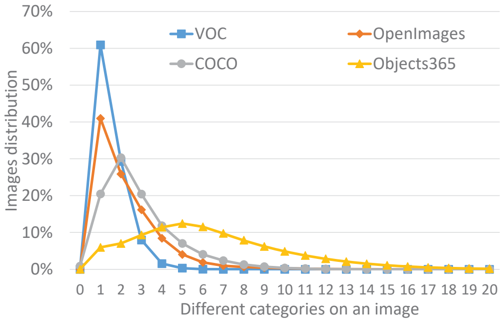

Figure 5. A comparison with the statistics of the number of object categories per image. Objects365 is more dense, with a modal on 5 categories per image, and more diverse, with a more flat curve.

For the image resolution, Figure 6 shows that Objects365 has larger and more diverse image resolutions.

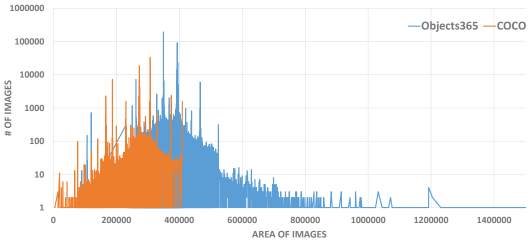

Figure 6. Distribution of the image resolution.

Figure 7 provides a comparison of “effective annotations” with the PASCAL VOC and COCO benchmark. By selecting the COCO 80 categories from Objects365, i.e., ignoring the other object categories, our dataset has more boxes per image, i.e., 9.04 vs 7.34, as well as more object categories per image, 3.16 vs 2.92. A similar conclusion can be obtained by mapping the object categories into the VOC 20 categories. We also compare the effective annotation areas, i.e., the ratio of annotated object areas to the total area of the images. Our Objects365 provides obvious more annotations 63% compared with 57% in COCO and 53% in VOC.

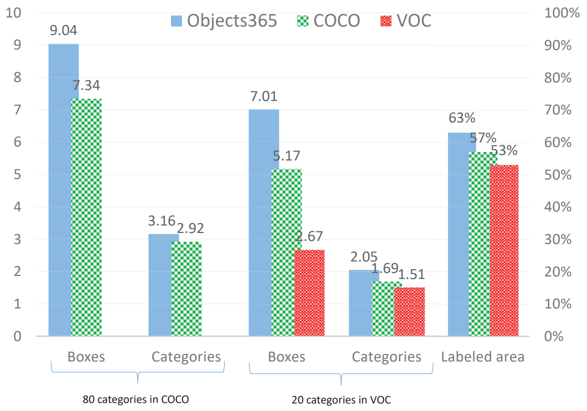

Figure 7. A comparison of the statistics of effective annotations with VOC and COCO datasets.

3.4. Quality

To validate the quality of our Objects365 dataset, three well-trained annotators are asked to label 200 randomly selected images. In total, there are 3250 bounding-boxes based on the refinement from the annotators. 92% of instances are annotated in the original annotations. A comparison of annotation recall with COCO and OpenImages can be found in Table 3.

Table 3. Comparison of annotation recall.

Dataset	OpenImage	COCO	Objects365
Recall(%)	43	83	92

For the annotation precision, we consider a false positive if the object category is wrong or the annotation bounding-box is not accurate. The precision of our Objects365 is obviously higher than COCO with 91.7% vs 71.9%.

4. Experiment

As Objects365 is a new benchmark, we first provide the baseline results based on two widely used detectors, FPN [22] and RetinaNet [23]. Then, we study the generalization ability.

4.1. Experiment Setup

The COCO style mmAP is adopted to evaluate the performance of our Objects365 benchmark. More specifically, we average the IoU=.50:.05:.95 for all the object categories.

For the implementation details, we follow the setting defined in Detectron [14, 15] for COCO. We train our detector on 8 1080-Ti GPUs, with a batch-size of 16. The input image size is 800×1333, which is also the same in the training process and testing process. A similar setting is utilized for training Objects365 except for the learning rate schedule, where we adopt a learning rate of 0.02, decreased by a factor of 10 at 900K and 1200K, and stopped at 1350K iterations. To speed up the convergence, we apply Syn the Batch Normalization technique in [26]. For the other datasets, we follow the standard setting.

4.2. Results on Object365

In Table 4, we compare the results of FPN and RetinaNet. The algorithms are trained on the Objects365 training set and the results are reported on the Objects365 validation set as described in Table 2. The mmAP for FPN is 22.5, which is significantly lower than the result on the COCO benchmark (38.3). This shows that our benchmark is more challenging. To analyze the results of our Objects365 dataset, we select the 80 categories defined in COCO from the 365 categories and obtain the mmAP of 38.5 for FPN, 34.5 for RetinaNet, which is comparable to the results on the COCO benchmark. We can see that the low mmAP score on Objects365 is due to the large number of categories in Objects365 dataset, especially those categories which are not defined in COCO dataset.

In addition, we perform diagnosis based on [18] and the results can be found in Figure 8. By comparing the result on Objects365 (left) and the result on COCO (right), we can see that the main gap for the low performance for Objects365 lies on the recall (the gap between BG and FN). A large number of object instances have been missed in our Objects365 dataset. Also, there exist a few object categories which do not have any positive matches, like *radish* and *saw*. In Figure 9, we show three examples for the results on the Objects365. We can see that the false negative may exist in small objects like glove/skis as well as the rare objects like antelope/swing.

4.3. Generalization Ability of Object365

In this sub-section, we study the generalization ability of Object365 as the pretraining dataset for object detection and semantic segmentation. For the detection problem, we select COCO and Pascal VOC to evaluate the general object detection, CityPersons to evaluate the ability on the pedestrian detection. For semantic segmentation, two standard benchmarks like PASCAL VOC and ADE are adopted.

4.3.1 Learning Rate Strategy

We will first discuss how to design the learning rate strategy for finetuning. Let us take the COCO dataset as an example. First, we simply adopt the standard learning rate strategy in [14] for a finetuning experiment. As shown in Table 5, this standard fine-tune strategy based on Objects365 (the third column in Table 5) only provides a small improvement (40.4 vs 39.3) compared with the ImageNet pretraining (shown in the second column of Table 5 with 540K iterations training for the COCO dataset).

To further exploit the finetuning capability of Objects365, we analyze the difference between Objects365

Method	$mmAP$	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}	AR^{1}	AR^{10}	AR^{100}	AR^{S}	AR^{M}	AR^{L}
FPN	22.5	35.5	24.3	12.8	24.4	32.8	23.0	36.7	38.1	25.2	40.7	49.8
RetinaNet	18.7	27.3	20.4	9.0	21.1	28.8	21.3	33.3	34.4	19.0	38.4	50.1

Table 4. Results of the baseline algorithms on the Objects365 dataset.

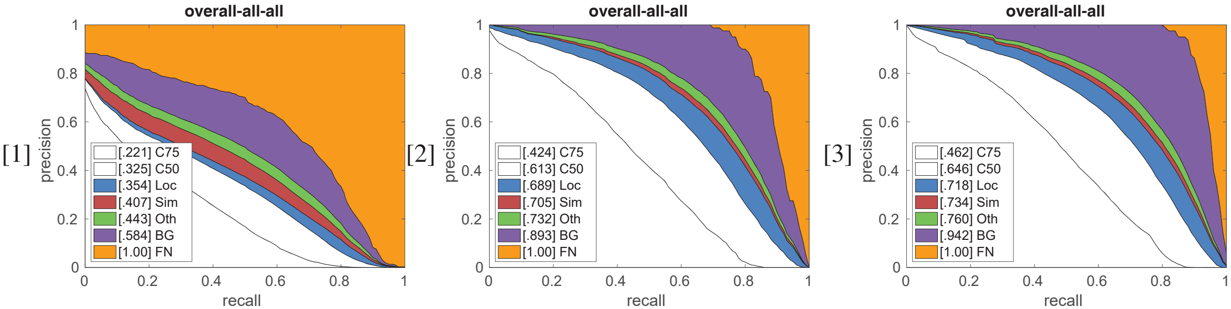

Figure 8. Diagnosis results on Objects365 and COCO. Three figures denote the result on Objects365 (left), the result on COCO with the model pretrained from ImageNet (middle), the result on COCO with the model pretrained from Objects365 (right). All the algorithms are implemented with FPN based on Resnet50 backbone.

Pretraining Dataset	ImageNet	Obj365	Obj365
mmAP	39.3	40.4	42.0
lr@begining	0.02	0.02	0.003
lr@convergence	0.0002	0.0002	0.00012
L2@beginning (1e3)	5.67	0.85	0.85
L2@convergence (1e3)	1.42	1.3	0.85

Table 5. A comparison of different learning rate strategies for finetuning on the COCO benchmark. We list the L2-Norm of the weights from the pretrained model of ImageNet and Objects365 at the beginning and convergence of the training. From left to right, these three fine-tuning experiments take 540K, 180K and 90K iterations, respectively.

pre-trained weights and ImageNet pre-trained weights. As shown in the last row of the Table 5, we find that the pretrained weights from Objects365 have much smaller L2-Norm compared with the ImageNet pre-trained weights. It is about **0.15** times smaller (0.85 vs 5.67) at the beginning of training. Intuitively, the smaller weights usually require a smaller learning rate to train. Therefore, we propose to use a smaller learning rate for Objects365 pre-trained models. Specifically, we design the learning rate according to the ratio of the pre-trained weights' L2-Norm between Objects365 and ImageNet. At the beginning of the training, we multiply the original learning rate with the L2-norm ratio of 0.15 and set the new learning rate as $3e^{-3} = 0.15 \times 0.02$. We decrease the learning rate by multiplying 1/5 twice during the training process. As Table 5 shows, our new learning rate strategy (the fourth column) can further improve the finetuning capability of the Objects365, which yields a large gain (42.0 vs 40.4).

4.3.2 Finetuning Results

COCO Detection According to Table 6, our Objects365 dataset shows strong generalization ability. Compared with FPN and RetinaNet baseline, our finetuned model can significantly boost the performance, with 5.6 points (42 vs 36.4) over the baseline for FPN with the 90K iterations, and 2.5 points (39.5 vs 37) for RetinaNet with the 180K iterations. By pretraining the Objects365 dataset which removes the 80 categories from COCO (Objects365 w/o COCO 80), we can still achieve competitive results with 39.6 on the COCO benchmark. Compared with OpenImages, our Objects365 benchmark has much higher performance gain. By involving ImageNet Pretraining before Objects365, we can slightly improve the FPN with 0.3 (42.3 vs 42). But for the RetinaNet, whose parameters mainly lie on the backbone part, the ImageNet pretraining before Objects365 can further bring in 1.5 points gain (41 vs 39.5).

Method	Pretrain Dataset	Iters	mmAP
FPN	None	540K	39.4
FPN	ImageNet	90K	36.4
FPN	ImageNet	180K	38.3
FPN	ImageNet	540K	39.3
FPN	OpenImages	90k	37.4
FPN	Objects365 w/o COCO 80	90K	39.6
FPN	Objects365	90K	42.0
FPN	ImageNet -> Objects365	90K	**42.3**
RetinaNet	ImageNet	180K	37.0
RetinaNet	Objects365	180K	39.5
RetinaNet	ImageNet -> Objects365	90K	**41.0**

Table 6. Generalization ability of general object detection results on the COCO dataset. "Iters" denotes the number of iterations for finetuning the models on the COCO dataset.

According to Figure 8 (middle and right figures), the Objects365 pretrained model (right figure) can significantly improve the classification ability of the ImageNet pretrained model (middle figure). The performance gain is mainly due to two factors: the large number of object categories and the

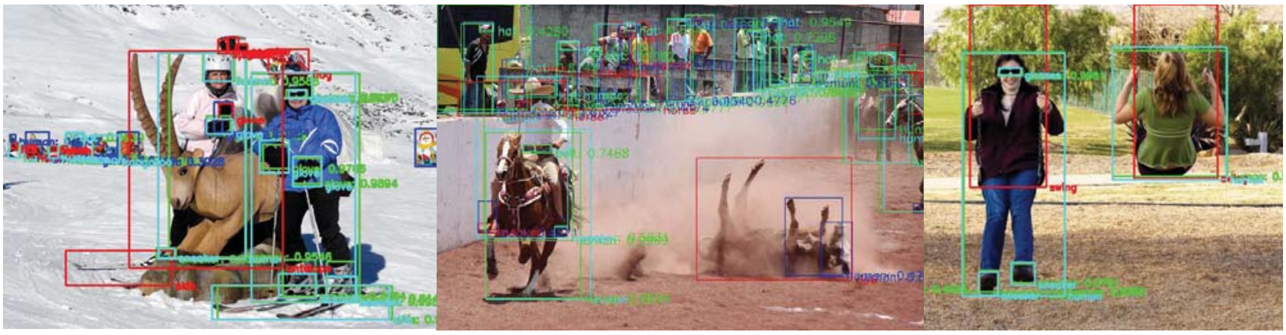

Figure 9. An illustration of the results on the Objects365 dataset. FPN with Resnet50 backbone is utilized. The green bounding-boxes denote the ground-truth (GT) which are positive matched by the predicted bounding-boxes colored with light-blue. Red and blue bounding-boxes denote the false negative GT and false positive predictions, respectively.

Pretraining Dataset	mAP
None	63.4
ImageNet	80.2
ImageNet -> COCO 540K iters	85.1
OpenImages	82.4
Objects365	86.2
ImageNet -> Objects365	**86.7**

Table 7. Generalization ability of object detection results on the PASCAL VOC dataset. The results are implemented based on FPN with Resnet50 backbone.

Pretraining Dataset	MR
None	39.49
ImageNet	18.04
ImageNet -> COCO 540K iters	16.24
OpenImages	16.78
Objects365	**12.12**
ImageNet -> Objects365	12.52

Table 8. Generalization ability of pedestrian detection results on the CityPersons dataset. The results are implemented based on FPN with Resnet50 backbone.

improvement of localization ability. According to the gap between "FN" and "Loc", the Objects365 pretrained model can remarkably improve 2.9 points from 0.689 to 0.718 due to the improvement of the classification ability by involving more object categories. Also, the localization ability is improved by 0.9 if we compare the gap between C75 and Loc.

VOC Detection As shown in Table 7, our Objects365 can provide superior finetuning ability compared with ImageNet and OpenImages. More specifically, we have 6 points gain (86.2 vs 80.2) compared with the ImageNet pre-trained model and more than 1.1 points gain (86.2 vs 85.1) compared with the COCO 540K iterations finetuned model.

CityPersons Besides general object detection, it is also important to validate the generalization ability on the specific object detection problem like pedestrian detection.

In this experiment, we adopt CityPersons [36], one of the standard benchmarks for pedestrian detection. The evaluation follows MR (miss-rate), which is widely used in the pedestrian detection community. We employ FPN as our baseline algorithm. We train the network for 180k iterations.

Table 8 shows the results of pedestrian detection on the CityPersons dataset. We find that pretraining with Objects365 can significantly outperform the baseline algorithms with more than 4 points gain.

VOC Segmentation In addition to the generalization ability for the detection task, we also validate the generalization ability of our Objects365 based on two semantic segmentation benchmarks: PASCAL VOC [8] and ADE [38].

PASCAL VOC contains 20 object categories with pixel-level annotations. The metric of mIOU (mean Intersection-over-Union) is employed to evaluate the performance. PSPNet [37] with Resnet50 backbone is adopted as the baseline algorithm.

According to Table 9, our algorithm can provide meaningful performance gain by pretraining the backbone part based on Objects365. More specifically, we have more than 2 points over the ImageNet pre-trained (76.7 vs 74.5) and OpenImages pre-trained (76.7 vs 74.1) models, and 1.8 points (76.7 vs 74.9) over the COCO 540K iterations finetuned model.

ADE ADE [38] is a scene parsing dataset introduced in 2017 with around 20K training images from 150 object categories. We report the results on the validation set with 2000 images. According to the Table 10, our Objects365 pretraining model has large advantages over other pretraining models, i.e., 1.7 points (42.9 vs 41.2) over the ImageNet pretraining model and 1.4 points (42.9 vs 41.5) over the COCO 540K iterations finetuned model.

Pretraining Dataset	mIOU
None	58.3
ImageNet	74.5
ImageNet -> COCO 540K iters	74.9
OpenImages	74.1
Objects365	**76.7**
ImageNet -> Objects365	76.6

Table 9. Generalization ability of semantic segmentation results on the PASCAL VOC dataset. The results are implemented based on PSPNet with Resnet50 backbone.

Pretraining Dataset	mIOU
None	30.2
ImageNet	41.2
ImageNet -> COCO 540K iters	41.5
OpenImages	40.7
Objects365	42.9
ImageNet -> Objects365	**43.3**

Table 10. Generalization ability of semantic segmentation results on the ADE dataset. The results are implemented based on PSPNet with Resnet50 backbone.

4.3.3 Speed Up Finetuning

According to our experiments on COCO, we find that Objects365 dataset could help researchers to accelerate their finetuning processes. As shown in Table 11, the algorithm trained only 12K iterations based on the Objects365 pre-trained model can have the comparable performance against the model trained with 540K iterations based on the ImageNet pre-trained model. Therefore, by utilizing the Objects365 pretraining, we can obtain more than 10-20 times faster training time without compromising the performance on the COCO benchmark. This significantly reduces the training cost and speed-up the innovation cycle.

4.3.4 Upper-bound

As studied in [15], given sufficient training time, training from scratch can obtain comparable performance as the training from ImageNet pretraining. The model pre-trained with our Objects365 dataset can significantly outperform the model with sufficient long training time, 42.0 vs 39.3 as shown in Table 11. It validates that the Objects365 pre-trained model can further push the upper-bound results for the existing algorithms. There are two potential reasons for the large improvement. First, the feature learned on larger scale dataset is better. Second, the pretraining of both backbone and head provides meaningful gains compared with pretraining the backbone only.

Train Dataset	Iterations	mmAP
ImageNet-> COCO	90K	36.4
ImageNet-> COCO	180K	38.3
ImageNet-> COCO	420K	39.3
ImageNet-> COCO	540K	39.3
Objects365-> COCO	12K	39.0
Objects365-> COCO	18K	40.0
Objects365-> COCO	36K	41.1
Objects365-> COCO	90K	42.0

Table 11. Comparison of the training time for the COCO general detection task. The algorithm is implemented based on the FPN with the Resnet50 backbone. Iterations denotes the number of iterations for the COCO training.

Method	Pretrain Part	Iters	mmAP
ImageNet	Backbone	90K	36.4
ImageNet	Backbone	180K	38.3
ImageNet	Backbone	540K	39.3
Objects365	Backbone	90K	37.8
Objects365	Backbone	180K	39.4
Objects365	Backbone	540K	40.3
Objects365	Backbone & Head	90K	**42.0**

Table 12. Comparison of the pretraining backbone only against pretraining both the backbone and head on the COCO benchmark. The results are implemented based on FPN with Resnet50 backbone. "Iters" denotes the number of iterations for the COCO training.

4.3.5 Pretrain Backbone vs Pretrain Backbone+Head

One of the main advantages of pretraining with our Objects365 benchmark is due to the pretraining weights of both the backbone and head parts, instead of ImageNet pretraining which only provides the weights for the backbone part. In Table 12, we show an experiment in which by dropping the pretraining weights (randomly initializing the weights) of the head part from a pre-trained Objects365 model, the performance of the finetuned detector drops around 2.6 points (42.0 vs 39.4) on the COCO benchmark. This validates that pretraining head is also important.

5. Conclusion

In this paper, we present a large-scale, high-quality object detection dataset, Objects365, which establishes a new challenge and benefits the many existing localization-sensitive vision tasks. In the future, we plan to investigate bigger models than ResNet-50.

Acknowledgement

This research was partially supported by National Key RD Program of China (No. 2017YFA0700800), Beijing S&T Program (No. Z181100008918016), and Beijing Academy of Artificial Intelligence (BAAI).

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] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.

[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] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.

[6] Jia Deng, Olga Russakovsky, Jonathan Krause, Michael S Bernstein, Alex Berg, and Li Fei-Fei. Scalable multi-label annotation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3099–3102. ACM, 2014.

[7] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):743–761, 2012.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[9] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[10] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[11] 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.

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

[13] 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.

[14] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. `https://github.com/facebookresearch/detectron`, 2018.

[15] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. *arXiv preprint arXiv:1811.08883*, 2018.

[16] 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.

[17] 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.

[18] Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. In *European conference on computer vision*, pages 340–353. Springer, 2012.

[19] Vidit Jain and Erik Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical report, UMass Amherst Technical Report, 2010.

[20] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018.

[21] Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. Detnet: A backbone network for object detection. *ECCV*, 2018.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] Chao Peng, Tete Xiao, Zeming Li, Yuning Jiang, Xiangyu Zhang, Kai Jia, Gang Yu, and Jian Sun. Megdet: A large mini-batch object detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6181–6189, 2018.

[27] 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.

[28] 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.

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

[30] 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.

[31] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: a benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018.

[32] Bharat Singh and Larry S Davis. An analysis of scale invariance in object detection snip. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3578–3587, 2018.

[33] Bharat Singh, Mahyar Najibi, and Larry S Davis. Sniper: Efficient multi-scale training. In *Advances in Neural Information Processing Systems*, pages 9333–9343, 2018.

[34] Zhe Wu, Navaneeth Bodla, Bharat Singh, Mahyar Najibi, Rama Chellappa, and Larry S Davis. Soft sampling for robust object detection. *arXiv preprint arXiv:1806.06986*, 2018.

[35] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5525–5533, 2016.

[36] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3221, 2017.

[37] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2881–2890, 2017.

[38] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 633–641, 2017.

[39] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. *arXiv preprint arXiv:1811.11168*, 2018.