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 longer training time like 540K iterations, our Objects365 pre-trained 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 pre-trained 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.

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

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

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.
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.

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Validation</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Images</td>
<td>600k</td>
<td>38k</td>
<td>100k</td>
</tr>
<tr>
<td>Boxes</td>
<td>9623k</td>
<td>479k</td>
<td>1700k</td>
</tr>
</tbody>
</table>

Table 2. The split of our Objects365 benchmark.

Figure 5 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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>OpenImage</th>
<th>COCO</th>
<th>Objects365</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall(%)</td>
<td>43</td>
<td>83</td>
<td>92</td>
</tr>
</tbody>
</table>

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=.5:.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 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
pre-trained weights and ImageNet pre-trained weights. As shown in the last row of the Table 5, we find that the pre-trained 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).

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.

<table>
<thead>
<tr>
<th>Pretraining Dataset</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>63.4</td>
</tr>
<tr>
<td>ImageNet</td>
<td>80.2</td>
</tr>
<tr>
<td>ImageNet -&gt; COCO 540K iters</td>
<td>85.1</td>
</tr>
<tr>
<td>OpenImages</td>
<td>82.4</td>
</tr>
<tr>
<td>Objects365</td>
<td>86.2</td>
</tr>
<tr>
<td>ImageNet -&gt; Objects365</td>
<td><strong>86.7</strong></td>
</tr>
</tbody>
</table>

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

The 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.
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.

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


