BigDetection: A Large-scale Benchmark for Improved Object Detector Pre-training

Likun Cai[1]* Zhi Zhang[2] Yi Zhu[2] Li Zhang[1] Mu Li[2] Xiangyang Xue[1]
[1]Fudan University [2]Amazon Inc.

Abstract

Multiple datasets and open challenges for object detection have been introduced in recent years. To build more general and powerful object detection systems, in this paper, we construct a new large-scale benchmark termed BigDetection. Our goal is to simply leverage the training data from existing datasets (LVIS, OpenImages and Object365) with carefully designed principles, and curate a larger dataset for improved detector pre-training. Specifically, we generate a new taxonomy which unifies the heterogeneous label spaces from different sources. Our BigDetection dataset has 600 object categories and contains over 3.4M training images with 36M bounding boxes. It is much larger in multiple dimensions than previous benchmarks, which offers both opportunities and challenges. Extensive experiments demonstrate its validity as a new benchmark for evaluating different object detection methods and its effectiveness as a pre-training dataset. The code and models are available at https://github.com/amazon-research/bigdetection.

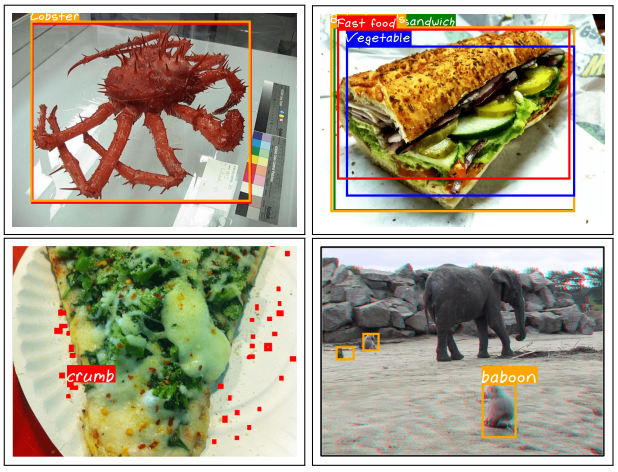


Figure 1. Visual examples from OpenImages (top) and LVIS (bottom) datasets. Top left (wrong label): "*Crab*" mistakenly labeled as "*Lobster*". Top right (bbox overlapping): bboxes with different class labels locate at the same place. Bottom left (uninformative annotations): class "*crumb*" may not be useful for detector pre-training. Bottom right (long-tail): there is only one image with "*baboon*" in the dataset.

1. Introduction

Back in 2014, Microsoft COCO dataset [30] was an extremely challenging benchmark where best performing methods were claiming average precision scores less than 20 AP across all 80 categories. Now, state-of-the-art detectors [9, 55] are already able to achieve 60+ AP on COCO test-dev. As a golden standard, COCO has incubated many popular object detection algorithms.

To build more robust and general object detection systems, several larger-scale object detection datasets have been released, such as OpenImages [23], Objects365 [39], and LVIS [21]. However, each dataset has its own limitations and challenges. For example, OpenImages has around 10% bounding box annotations that are machine-generated, which may cause problems like wrong label and bounding box overlapping (Fig. 1 top). LVIS aims to craft a diverse set of densely annotated labels covering more than 1200 categories, but may bring problems like uninformative annotation and serious long-tail distribution (Fig. 1 bottom). Object365 has a relatively smaller vocabulary which may miss common object categories like insect.

In this work, we introduce a new large-scale object detection benchmark, termed *BigDetection*. Our goal is to simply leverage the training data from existing datasets (like LVIS, OpenImages and Objects365) with carefully designed principles, so that we can curate a larger dataset more suitable for object detector pre-training. Different from literature in multi-dataset detector training [52, 58, 61], we use language model to build our initial unified label space across datasets and perform manual verification to obtain the final taxonomy as shown in Fig. 2. Our BigDetection dataset has 600 object categories and contains 3.4M training images with 36M bounding boxes. We show the statistics comparison to other datasets in Tab. 1. In addition, we

*Work done during an internship at Amazon.

	Train		Val		Num. classes
	Num. images	Num. boxes	Num. images	Num. boxes	
LVIS [21]	100K	1.27M	19K	244K	1203
OpenImages [23]	1.74M	14.61M	41K	303K	600
Objects365 [39]	1.72M	22.89M	80K	1.06M	365
BigDetection	3.48M	35.96M	141K	1.58M	600

Table 1. Comparison of the dataset statistics among popular large-scale object detection benchmarks.

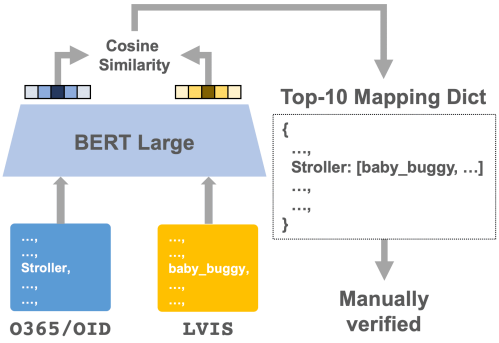


Figure 2. Overview of our category mapping pipeline, which is used to generate the unified label space of BigDetection. See text in Sec. 3.2 for more details.

perform various experiments to demonstrate its validity as a new benchmark for evaluating different object detection methods, and its effectiveness as a pre-training dataset. In particular, as we can see in Tab. 3, a CBNetv2 [28] model with Swin-Base backbone [32] pre-trained on BigDetection can achieve 59.8 AP on COCO test-dev set. It is surprising to find that this performance is competitive with the same model using Swin-Large backbone without pre-training on BigDetection. Note that Swin-Large is twice heavier than the Swin-Base model. In addition, following a partially labeled data setting [43] on COCO, BigDetection pre-training is shown to be extremely data efficient. *e.g.*, 25.3 AP on COCO validation set using only 1% COCO training data.

Our contributions can be summarized as follows:

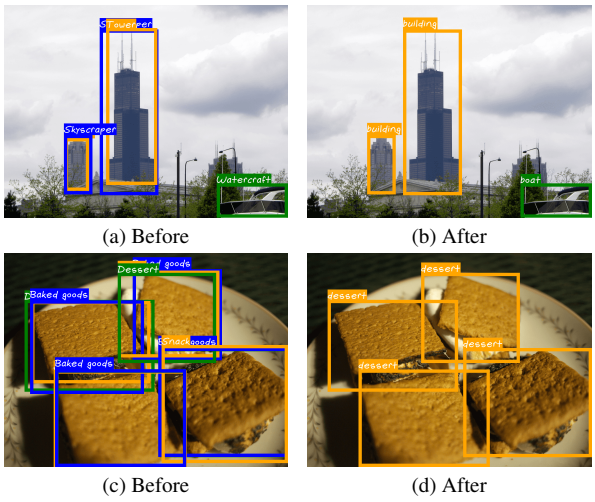
- We introduce a new object detection dataset, BigDetection, which is much larger in multiple dimensions than previous benchmarks. It could serve as a more challenging benchmark for evaluating different object detection methods.
- We show effectiveness of BigDetection as a pre-training dataset. We obtain state-of-the-art results on COCO validation and test-dev sets, as well as under data-efficiency settings.
- We perform extensive ablation studies to provide good practices when training object detectors on large-scale datasets.

2. Related Work

Datasets for object detection Large-scale datasets with high-quality annotations play a crucial role in advancement of better computer vision models. In terms of object detection, PASCAL VOC [13] is one early benchmark containing 20 classes over 17k images. Despite its relatively small scale compared to datasets nowadays, PASCAL VOC has successfully bred many object detectors including both classical detectors [15, 53] and deep learning detectors [19, 20, 22]. Then comes Microsoft COCO [30] in year 2014, which is the most widely adopted benchmark for object detection to present. It contains 118k images and 860k instance annotations over 80 classes. Thanks to its large-scale and great quality, COCO together with deep learning have revolutionized the landscape of computer vision. Recently, with extensive high quality labeling efforts, larger scale datasets like LVIS [21], OpenImages [23] and Objects365 [39] are introduced with millions of instance-level annotations. They enable us to learn diversified and fine-grained object concepts, as well as explore the possibility of few-/zero-shot learning on new scenes. There are also more datasets for object detection in specific domains, such as [8, 17, 35, 40, 44, 48], to support various use cases.

Multi-dataset detector training Annotating gigantic datasets by human labor is not scalable. Hence some recent work start to explore multi-dataset training strategy, whose goal is to learn better feature representations from more labeled data given existing datasets.

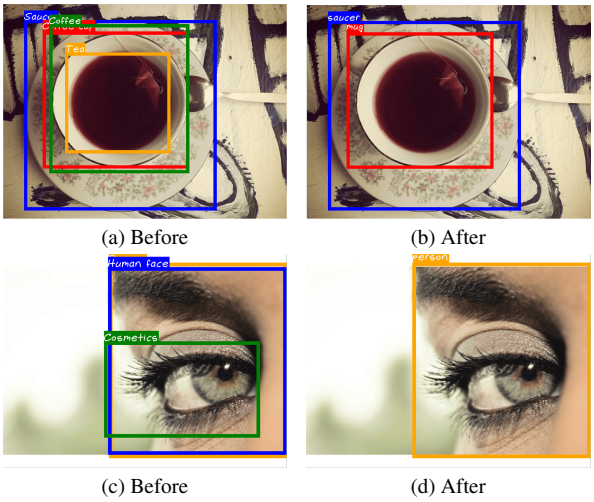
One early attempt [52] proposes to train a universal object detector with domain attention on multiple datasets. All parameters and computations are shared so that one detector can leverage knowledge across domains. In order to address the partial annotation problem when using multiple datasets with heterogeneous label space, UOD [58] exploits a pseudo labeling mechanism to unify the label space for training a single detector. Following [52], Zhou et al. [61] proposes a weighted graph matching behind split classifiers to automatically generate a common taxonomy. This framework can generalize better to new test domains without prior knowledge, and achieves great zero-shot performance. In order to alleviate the scale variation problem, USB [42]



(a) Before (b) After

(c) Before (d) After

Figure 3. Visual examples of bounding box overlapping problem in OpenImages dataset. Left column: original annotations with multiple boxes over the same object. Right column: annotations after our bounding boxes de-overlapping.



(a) Before (b) After

(c) Before (d) After

Figure 4. Visual examples of object categories that might confuse model training, such as "*Coffee*" and "*Coffee cup*", "*Cosmetic*" and "*Human eye*". Left column: original annotations. Right column: categories we keep in BigDetection.

introduces a universal-scale object detection benchmark to enable multi-scale object detection. Their proposed UniverseNet achieves top performance on two challenges.

Different from the above work, we construct the final unified label space through a carefully designed mapping pipeline and strict manual verification, which makes our unified label space more credible than those machine generated results. In addition, existing object detectors can be trained directly on our dataset without any modifications like split classifiers, domain attention or graph matching. Thus our composite dataset provides a new benchmark that easily enables fair comparison. There is a recent work, MSeg [24], that is similar to us in terms of manually building a composite dataset. However, MSeg is designed for semantic segmentation and it only contains 200k images over 194 semantics classes. Our composite dataset is significantly larger, and we provide both clear benefits of pre-training and large-scale analysis.

Object detectors Given these well annotated datasets, deep learning based object detectors have made significant progress over the past decade. Based on the network design, existing object detectors using convolutional neural networks (CNNs) can be roughly divided into two types: single-stage detector [2, 14, 16, 25, 27, 31, 36, 37, 47, 62] and two-stage detector [4, 19, 20, 38, 60]. The two-stage models usually offer better performance while the one-stage models run with faster speed. Recently, with the rise of transformer [12, 49] in computer vision, some works investigate the combination of CNNs and transformer for improved object detection [1, 5, 10, 45, 51, 59, 63]

3. BigDetection Dataset

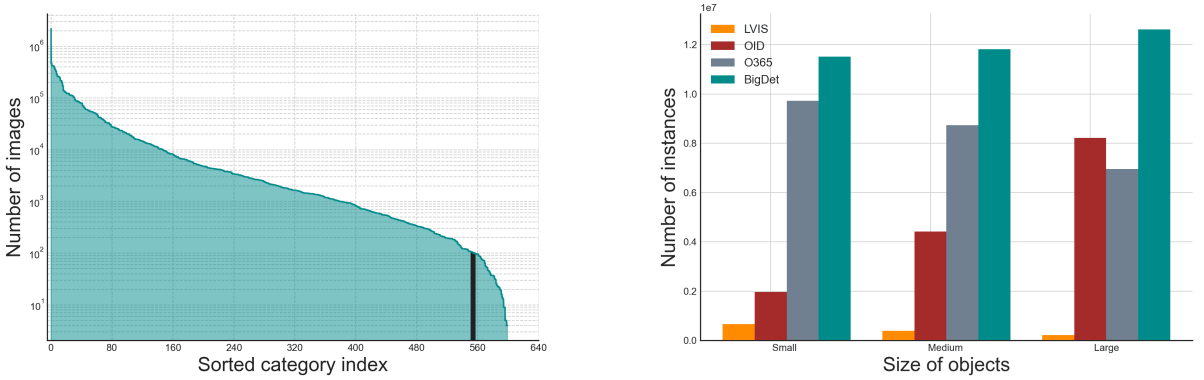
The goal of this work is to construct an evolving object detection benchmark designed to incubate next generation object detectors. Our basic idea is to simply leverage the training data from several existing datasets, with carefully designed principles to construct a larger dataset more suitable for pre-training.

3.1. Existing Datasets and Limitations

We first give a brief review on three existing large-scale object detection datasets LVIS [21], OpenImages [23] and Objects365 [39]. All three datasets have been widely used for object detection pre-training.

LVIS V1.0 LVIS is a dataset designed for large vocabulary instance recognition. It collects high-quality object bounding boxes and segmentation masks for over 1200 object categories using samples of COCO [30]. However, LVIS naturally has an extremely long-tailed distribution. Nearly half of the categories in LVIS have few training examples (e.g., ≤ 20). Besides, given its object categories are more than 10 times of COCO, LVIS has some uninformative annotations, such as the "*crumb*" example in Fig 1. Both attributes make LVIS unsuitable as a pre-training dataset.

OpenImages V6 OpenImages (OID) is a large-scale dataset of about 9M images with rich annotations, including image-level labels, object bounding boxes, object segmentation masks, visual relationships, localized narratives, etc. In terms of object detection, OpenImages has 14.6M bounding boxes over 600 object classes. 90% of these boxes are manually drawn by professional annotators using clicking



(a) (b)

Figure 5. (a) Number of images per category of BigDetection. BigDetection have 555 frequent categories (black line) out of 600, which means it suffers less from long-tail problem. (b) Number of instances in different object sizes. We find that OpenImages and Objects365 are biased towards certain scale, while BigDetection is balanced across object scales.

interface [34], while the remaining 10% are produced semi-automatically using [33]. We find that there still exists a fair amount of annotations with poor quality. For example, we observe bounding box overlapping problem. As can be seen in Fig. 3a and Fig. 3c, several bounding boxes with similar size locate at the same place, but have different class labels. This may confuse model training. We also argue that some categories in OpenImages may not be useful for general detector pre-training, such as "*tea*" and "*cosmetics*" in Fig. 4a and Fig. 4c.

Objects365 Objects365 is another large-scale dataset designed for object detection pre-training. It contains around 1.72M images with more than 22.8M bounding boxes over 365 categories. Comparing with OpenImages, Objects365 is close in terms of dataset scale, but has a smaller vocabulary. This may not cover enough semantic concepts to pre-train a universal object detector and generalize to other domains.

3.2. Building a Unified Label Space

Despite being large-scale, these three datasets have their own limitations and challenges, such as LVIS being too fine-grained, noisy annotations in OpenImages and relatively small number of object categories for Objects365. It would be ideal if we can find a way to combine the datasets and alleviate their individual limitations. However, this is non-trivial given the heterogeneous label spaces.

As we mentioned in Sec. 2, there are some studies on multi-dataset detector training, such as using split classifiers [52, 61] and pseudo labeling [58]. But considering the noisy annotations and domain gap among different datasets, we would like to clean the data before model training and combine datasets in a more careful manner.

Our goal is to merge the datasets under one unified label space, and train a single detector on it. In order to create the unified label space, we introduce a category mapping pipeline using language models, which is illustrated in Fig. 2. First, we adopt LVIS's object categories as the initial vocabulary, since LVIS dataset has the largest taxonomy with the most fine-grained annotations. Second, we utilize a Bert-Large model[1] [11] to extract features of category words in each dataset. Third, we compute a cosine similarity between each category word of Objects365/OpenImages and that of LVIS. The intuition is the more similar the features are, the higher possibility those categories can be merged. Thus, an initial category mapping dictionary will be generated by collecting the top-10 similar pairs. In the end, to further enhance the validity of the final vocabulary, we manually verify each matching pair in the dictionary with the following principles:

Classes matching We notice that some object categories should belong to the same semantic concept, but their feature similarity is low due to different category words, such as "*Remote*" (Objects365) and "*remote_control*" (LVIS). In this case, we will perform a manual merge. For some categories in OpenImages and Objects365 that never occurred in LVIS, we will just adopt them as new classes.

Classes merging Since the class granularity of each dataset is different, some categories have inclusion relationship in semantic space. In order to obtain a unified label space, we simply merge these categories into a single one. For example, we merge different bird specifies to the "*bird*" class.

Classes removing We argue that some non-discriminating categories or classes with too few training examples are not suitable for general detector pre-training. These classes will be directly removed. Some examples are illustrated in Fig. 1.

BBoxes de-overlapping We find that even after removing some classes, there still exists a large number of overlapped

[1] https://huggingface.co/bert-large-uncased

bounding boxes. In order to filter them credibly, we first collect all category pairs with bounding boxes IoU greater than 0.65. Then for object categories that always co-occur, we keep them if they are supposed to be multi-labels for the same object. Otherwise we remove them from the annotation set. We show some de-overlapping results in Fig. 3b and Fig. 3d.

3.3. BigDetection and Its Statistics

At this point, we have a unified label space that can combine the training data of Objects365, OpenImages, and LVIS. This allows us to build the largest existing object detection dataset, thus we name it BigDetection.

In Tab. 1, we show its statistics comparison to several other large-scale datasets. In terms of training set, BigDetection has around 36M bounding boxes in 3.4M images for 600 object categories. On average, there are 10.3 annotated bounding boxes per image.

In addition, we plot the number of images in each category in Fig. 5a. According to LVIS [21], a category is considered as *frequent* if there are more than 100 images it appears. In BigDetection, we have 555 *frequent* classes, which surpasses OpenImages (540) and Objects365 (363). Since the majority of classes are frequent, BigDetection suffers less from long-tail problem, which makes it more ideal for object detector pre-training.

In terms of object sizes, we plot the number of instances in each scale bin[2] for different datasets in Fig. 5b. We can see that OpenImages and Objects365 are biased towards certain scale, while BigDetection is balanced across object scales. We will show later that this property helps detector in reducing localization errors.

4. Pre-training on BigDetection

Large datasets are usually good for model pre-training, but they also pose challenges. For example, BigDetection has a serious class imbalance problem. Some classes like "*person*" have greatly more annotations than others like "*ferret*". In addition, BigDetection suffers from the partial annotation problem when merging the datasets. Some object categories annotated in one dataset could be considered as "*background*" in another dataset. In this section, we investigate effective methods to handle class imbalance and partial annotation problems during model training.

4.1. Class Imbalance

There are several widely adopted strategies to alleviate class imbalance problem, like loss re-weighting [26, 46, 50], data re-sampling [6, 21, 41] and data augmentation [18, 56]. In this work, we explore all of them and find that data re-sampling, especially class-aware sampling [41], is most effective.

To be specific, we use fixed class weights derived from the dataset to perform loss re-weighting. The more samples one class contains, the lower weight will be assigned to that class when computing the loss. However, this does not help the training since BigDetection is so imbalanced, which leads to slow convergence. For data augmentation, we adopt the recent CopyPaste [18] augmentation who achieves great performance on instance segmentation. The core idea of CopyPaste is to randomly paste masked objects from one image onto another inside a training batch. Unfortunately, BigDetection only has bounding box annotations. Directly pasting the boxed image patches will inevitably introduce unnecessary noise. For data re-sampling, we use class-aware sampling (CAS) method following [41]. CAS samples each class with equal probability, which is ideal for datasets with imbalanced classes. Note that since each sample contains multiple instances of different categories, the sampled data will not be completely balanced. We will present the experimental results in later Sec. 5.3.

4.2. Partial Annotations

In order to address the partial annotation problem when merging different datasets, we adopt a self-training approach similar to [58, 64] to complement the ground truth annotations. The goal is to generate additional pseudo annotations that were not manually labeled in the dataset.

In our work, using self-training is more straightforward than [58] since we already have a unified label space. To be specific, we first train a teacher model on BigDetection. Then the teacher model is used to generate pseudo annotations for the train set of BigDetection. Noted for object detection, pseudo annotations include two elements: pseudo labels for classification, and pseudo bounding boxes for localization. The credibility of pseudo labels and the maximum overlap area of pseudo boxes can be adjusted by changing the values of score threshold and NMS threshold of the teacher model, respectively. The last step is to incorporate these new pseudo annotations to the ground truth and train a better student model. However, in order to ensure that these pseudo boxes capture the missing objects without introducing more noise, we add an additional step to filter the pseudo annotations. Namely, one pseudo annotation will be removed if the IoU between its box and any ground truth box is greater than 0.6. Once filtered, the remaining pseudo annotations will be used to augment the training set.

We find that even two detectors have similar mAP on a dataset, their precision for each class differs greatly due to the different training setting. To further improve the credibility of pseudo annotations, we adopt a multi-teacher strategy. Suppose the ground truth annotation set

[2]The three object scales follow the definition in COCO [30]: Small $< 32 \times 32$, $32 \times 32 <$ Medium $< 96 \times 96$, $96 \times 96 <$ Large

Detector	Backbone	BigDetection			COCO	
		AP	AP_{50}	AP_{75}	AP	AP^*
YOLOv3 [14]	D53	9.7	17.4	9.7	21.8	30.5(+8.7)
Deformable DETR	R50	13.1	19.3	14.2	37.4	39.9(+2.5)
Faster R-CNN [38]	R50	18.9	28.8	20.5	35.7	38.8(+3.1)
Faster R-CNN [38]	R50-FPN	19.4	29.3	21.3	37.9	40.5(+2.6)
CenterNet2 [60]	R50-FPN	23.1	30.2	24.9	42.9	45.3(+2.4)
Cascade R-CNN [4]	R50-FPN	24.1	33.0	25.8	42.1	45.1(+3.0)

Table 2. BigDetection as a challenging and effective pre-training new benchmark. First, we provide comparison of popular object detection methods on BigDetection validation. All models are trained with an $8\times$ schedule to enable fair comparison. Then we show the finetuning results on COCO validation set after $1\times$ finetuning. AP^* indicates that models are pre-trained on BigDetection.

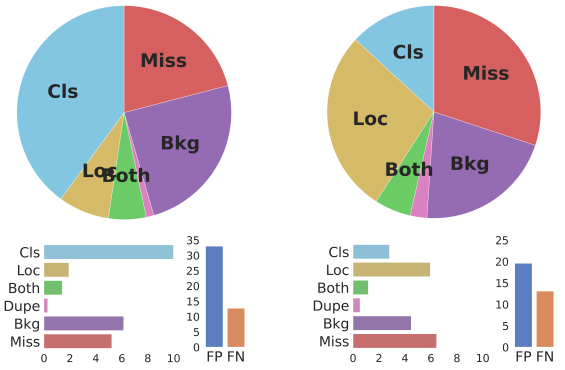


Figure 6. Error diagnose by TIDE [3]. Left: BigDetection. Right: COCO.

for sample i is Y^i_{gt}, and we have multiple teacher models $[t_1, t_2, \cdots, t_k]$. Each teacher model generates a pseudo annotation set $Y^i_{t_j}, j = 1, \cdots, k$. The final annotation set of sample i will be obtained:

$$\widetilde{Y}^i = \text{NMS}(Y^i_{t_1}, \cdots, Y^i_{t_k}; \tau) \cup Y^i_{gt}$$

where NMS (non-maximum suppression) is utilized to de-overlap multiple pseudo annotation sets and τ is the threshold. We set $k = 2$ throughout this work. More details can be found in the supplementary materials.

5. Experiments

Setup and evaluation protocol BigDetection is split into train set (`bigdet_train`) and validation set (`bigdet_val`). When using it as a new benchmark, we train different detection models on `bigdet_train` and evaluate their performance on `bigdet_val`. When using it as a pre-training dataset, we first pre-train the detection models on `bigdet_train`, and then finetune them on COCO train set, and report results on either COCO validation or test-dev set. For both evaluations on `bigdet_val` and COCO, we follow the standard COCO style metrics to report mean average precision (mAP) under different IoU thresholds and object scales. We also adopt a partially labeled data setting to study data efficiency. Pre-trained models will be finetuned on COCO using 1%, 2%, 5% and 10% labeled data.

Implementation details We adopt CenterNet2 [60] equipped with ResNet-50 and FPN [29] to provide baseline results and conduct ablation study. Our implementation is based on Detectron2 [54]. Most hyperparameters follow the default setting of CenterNet2 unless otherwise stated. Specifically, we train the detector with an SGD optimizer for $8\times$ (720K iterations) on BigDetection pre-training and $1\times$ (90K iterations) on COCO finetuning. Base learning rate is set to 0.02 and is dropped at iterations 660K/60K and 700K/80K. We use 8 V100 GPUs, with 2 samples per GPU. Multi-scale training is adopted with the short edge in range $[640, 800]$ and the long edge up to 1333. No extra data augmentations are used, such as Jittering, Mosaic [56], CopyPaste [18] or Mix-up [57].

When comparing to state-of-the-art detectors on COCO, we adopt CBNetV2 [28] equipped with a Swin-Transformer-Base backbone. For a fair comparison, the training strategy and all hyperparameters follow the default setting in [28] implemented with MMDetection [7]. Again, we do $8\times$ schedule for pre-training stage on BigDetection and $1\times$ for COCO finetuning.

5.1. A New Object Detection Benchmark

To provide a rough picture of how challenging BigDetection is, we select several most popular object detectors to evaluate their performance on `bigdet_val`. The methods include Faster R-CNN [38], Faster R-CNN with FPN [29], Cascade R-CNN [4], YOLOv3 [14], CenterNet2 [60] and DETR [5], which represent a variety of object detection models (*e.g.*, two-stage/one-stage, anchor-based/anchor-free, CNN-based/transformer-based). All these models are trained on `bigdet_train` for $8\times$ schedule to provide fair comparison. Other hyperparameters follow their default setting in MMDetection [7].

Method	Backbone	TTA	AP_{val}	AP_{test}	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
CBNetV2 [28]	Swin-B	✗	58.4	58.7	76.9	64.3	40.7	62.0	72.0
CBNetV2 [28]	Swin-B	✓	58.9	59.3	77.6	65.0	41.7	62.7	72.5
CBNetV2 (BigDet)	Swin-B	✗	59.1	59.5	77.3	65.3	42.0	62.4	72.7
CBNetV2 (BigDet)	Swin-B	✓	**59.5**	**59.8**	**77.9**	**65.6**	**42.2**	**62.9**	**73.0**

Table 3. Comparison with state-of-the-art object detection methods on COCO validation and test-dev sets. BigDet: pretrained on BigDetection dataset. TTA: test-time augmentation. Our CBNetv2 with Swin-B backbone achieves 59.8 AP on COCO test-dev.

Method	1%	2%	5%	10%
Supervised†	9.8	14.3	21.2	26.2
STAC† [43]	14.0	18.3	24.4	28.6
SoftTeacher† [55]	20.5	26.5	30.7	34.0
Ours	**26.1**	**29.3**	**31.9**	**34.1**

Table 4. Comparison with different methods under partially labeled COCO. BigDetection pre-training is particularly beneficial when dealing with insufficient training data. † indicates using strong augmentations, such as Cutout.

As can be seen in Tab. 2, good detectors on COCO also perform well on BigDetection, e.g., CenterNet2 and Cascade R-CNN are top performers on both datasets. However, we see that even the best result on BigDetection is 24.1 AP, which is close to the initial results when COCO was introduced in 2014. This suggests that BigDetection is a much more challenging dataset than COCO. We hope BigDetection can help advance the development of next-generation object detection algorithms. Unless otherwise stated, we use CenterNet2 for most experiments in the following sections. We find that CenterNet2 often show better generalization during fine-tuning.

In addition, we use our trained CenterNet2 model to perform an error diagnosis by TIDE [3]. The results can be visualized in Fig. 6. By comparing the result on BigDetection (left) and the result on COCO (right), we can see that the main difference lies in the Cls and Loc categories. BigDetection shows more Cls errors since it has much more object categories than COCO. At the same time, BigDetection makes far fewer Loc errors than COCO. Even within BigDetection, Loc errors are fewer than Miss and Bkg. We believe this is because BigDetection is more balanced across object scales as mentioned in Fig. 5b, so that it can better handle small and medium objects. This observation is also supported by our results later in Tab. 3 that pre-training on BigDetection improves AP_S significantly on COCO.

5.2. Generalization to COCO

Baseline We first show that BigDetection pre-training provides significant benefits for different detector architectures (single-stage or two-stage, anchor-based or anchor-free). Following the model set in Sec. 5.1, we finetune each model on COCO train split with $1\times$ schedule, and ImageNet pretrained checkpoints are adopted for comparison. After pretraining on BigDetection, most detectors gain $2 \sim 3$ AP improvement, and especially YOLOv3 even gains 8.7 AP improvement. These results suggest that BigDetection forms a strong pre-train dataset to provide better feature representation for downstream transfer.

Comparison to state-of-the-art We would like to show how far BigDetection can advance performance of current strongest detectors. We adopt CBNetV2 with Swin-transformer backbone as our baseline [28].

The results are shown in Tab. 3. We have several observations. First, with pre-training on BigDetection, we can further improve this strong baseline by 0.7 AP (58.4 $\rightarrow$ 59.1). In particular, most improvements come from small objects, i.e., AP_S increases from 40.7 to 42. Second, combined with test-time augmentation, our CBNetV2 model with Swin-Base backbone pre-trained on BigDetection achieves superior performance on both COCO validation and test-dev sets, 59.5 and 59.8 AP respectively. We want to point out that this performance with Swin-Base backbone is even competitive to CBNetv2 with Swin-Large backbone without pre-training on BigDetection. Note that Swin-Large is twice heavier than Swin-Base, which supports well that such pre-training is useful.

Data-efficiency One of the great benefits of pre-training on a large-scale dataset is a well-trained model only needs a few target labels to perform considerably well. Here, we show BigDetection pre-training is helpful across a variety of dataset sizes and helps data efficiency.

Following the partially labeled data setting introduced in STAC [43], Faster R-CNN [38] with FPN is adopted for fair comparison. The finetuning is done on COCO using 1%, 2%, 5% and 10% samples of train split. Tab. 4 summarizes the results. Our method significantly improves the performance upon supervised baseline and STAC. Compared to STAC, we obtain **12.1**, **11**, **7.5** and **5.5** AP gain under difference dataset sizes. We also compare our method to a recent work SoftTeacher [55], which is an end-to-end self-training method for object detection. Interestingly, our method shows a significant performance improvement (**5.6** AP) on 1% COCO setting, and performs still better when more data is introduced. Note that SoftTeacher uses longer training schedule and strong augmentations. In [65], the re-

	AP	AP_{50}	AP_{75}	COCO
IAS	12.8	17.2	13.9	44.9
RFS	20.4	26.9	21.9	44.2
CAS	**23.1**	**30.2**	**24.9**	**45.3**

Table 5. Ablation study on the effectiveness of different data samplers used to deal with class imbalanced problem. IAS: instance-aware sampling. RFS: repeated factor sampling. CAS: class-aware sampling. First three columns show results on BigDetection validation set, while the last column shows results on COCO.

sults show that self-training works better than pre-training across dataset sizes with a lowest data regime of 20%. However our work suggests that BigDetection pre-training is particularly useful when dealing with extremely insufficient training data ($1 \sim 10\%$).

5.3. Ablation studies

Regarding data re-sampling Recall in Sec. 4.1, data re-sampling is the most effective method to alleviate class imbalance problem. Here, we ablate the effects of using different samplers in Tab. 5.

IAS selects each training sample with equal probability, thus it is expected to perform poorly on class imbalanced datasets. RFS is designed for low-shot long-tailed data in LVIS [21]. It first assigns a pre-computed repeat factor for each category. Then the maximum factor of labeled categories will be chosen for each image. Since BigDetection does not show long-tail phenomenon, RFS performs mediocre on `bigdet_val`, and even affects the generalization ability to COCO. CAS offers the best performance as it is designed to handle class imbalance. It is simple, and thus scales well on large-scale dataset.

Regarding different pre-training datasets It is important to show how BigDetection compares to other large-scale datasets when used as pre-training dataset. We use fine-tuning results on COCO to reflect the capability of the pre-trained models.

In terms of baseline, we use models directly trained on COCO with $1\times$ and $8\times$ schedules. For other datasets, we always pre-train for $8\times$ schedule and finetune on COCO for $1\times$ schedule to enable fair comparison. CAS is adopted as data sampler for OpenImages, Objects365 and BigDetection, except for LVIS. Since RFS is shown to have more reasonable performance on LVIS. As shown in Tab. 6, our model pre-trained on BigDetection improves over the baseline by a notable margin ($43.8 \rightarrow 45.7$). It also outperforms models pre-trained on those individual datasets, which suggests its better potential in pre-training. Furthermore, Tab. 6 also shows individual contributions from using CAS and self-training, i.e., CAS brings 1.5 AP improvement while self-training brings another 0.4 AP improvement.

	Sampler	Schedule	AP
COCO	-	$1\times$	42.9
COCO	-	$8\times$	43.8
LVIS	RFS	$1\times$+$1\times$	37.8
OpenImages	CAS	$8\times$+$1\times$	44.0
Objects365	CAS	$8\times$+$1\times$	45.1
BigDetection	CAS	$8\times$+$1\times$	45.3
BigDetection†	CAS	$8\times$+$1\times$	**45.7**

Table 6. Ablation study on generalization ability to COCO using different pre-training datasets. † indicates using additional pseudo annotations generated by self-training. We show that BigDetection serves as a better pre-training dataset.

6. Limitations

Here we list several limitations that remain unsolved in our dataset. **Scalability** Despite our initial category mapping dictionary is automatically generated with the help of large language models, we heavily rely on manual inspection as described in Sec. 3.2 to build the final unified label space. This is more reliable than machine generated annotations, but also sacrifices scalability to some extent. **Data sampling** We use CAS to handle class imbalance problem, but it introduces a side problem. Within an $8\times$ schedule, the model may not see all the images in the dataset. This indicates a low utilization of the data. **Noisy annotations** Following our dataset merging principles, some noisy annotations from OpenImages have been removed, but some remain. Our work mainly aims at how to build a unified taxonomy to leverage existing datasets. And learning from noisy data will be a promising direction.

7. Conclusion

In this paper, we have presented BigDetection, an evolving large-scale object detection dataset. It is much larger in multiple dimensions (object categories, training images, bounding box annotations) than previous benchmarks. It could serve as a new challenging benchmark for evaluating different object detection methods, since state-of-the-art detectors only achieve around 30 AP on its validation set. We also show its effectiveness when used as a pre-training dataset. After pre-training on it, we achieve superior generalization performance on COCO validation and test-dev sets, as well as strong data-efficiency results. BigDetection presents both opportunities and challenges. We hope it can be used to incubate next-generation object detectors.

References

[1] Josh Beal, Eric Kim, Eric Tzeng, Dong Huk Park, Andrew Zhai, and Dmitry Kislyuk. Toward transformer-based object detection. *arXiv preprint arXiv:2012.09958*, 2020. 3

[2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 3

[3] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman. Tide: A general toolbox for identifying object detection errors. In *ECCV*, 2020. 6, 7

[4] 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. 3, 6

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 3, 6

[6] Nadine Chang, Zhiding Yu, Yu-Xiong Wang, Anima Anandkumar, Sanja Fidler, and Jose M Alvarez. Image-level or object-level? a tale of two resampling strategies for long-tailed detection. *arXiv preprint arXiv:2104.05702*, 2021. 5

[7] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[8] 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. 2

[9] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7373–7382, 2021. 1

[10] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1601–1610, 2021. 3

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 4

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[13] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 2

[14] Ali Farhadi and Joseph Redmon. Yolov3: An incremental improvement. In *Computer Vision and Pattern Recognition*, pages 1804–02. Springer Berlin/Heidelberg, Germany, 2018. 3, 6

[15] 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, 2009. 2

[16] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017. 3

[17] 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. 2

[18] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2918–2928, 2021. 5, 6

[19] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2, 3

[20] 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. 2, 3

[21] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5356–5364, 2019. 1, 2, 3, 5, 8

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015. 2

[23] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. 1, 2, 3

[24] John Lambert, Zhuang Liu, Ozan Sener, James Hays, and Vladlen Koltun. Mseg: A composite dataset for multi-domain semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2879–2888, 2020. 3

[25] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018. 3

[26] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10991–11000, 2020. 5

[27] Zuoxin Li and Fuqiang Zhou. Fssd: feature fusion single shot multibox detector. *arXiv preprint arXiv:1712.00960*, 2017. 3

[28] Tingting Liang, Xiaojie Chu, Yudong Liu, Yongtao Wang, Zhi Tang, Wei Chu, Jingdong Chen, and Haibin Ling. Cbnetv2: A composite backbone network architecture for object detection. *arXiv preprint arXiv:2107.00420*, 2021. 2, 6, 7

[29] 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. 6

[30] 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. 1, 2, 3, 5

[31] 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. 3

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 2

[33] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. We don't need no bounding-boxes: Training object class detectors using only human verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 854–863, 2016. 4

[34] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation. In *Proceedings of the IEEE international conference on computer vision*, pages 4930–4939, 2017. 4

[35] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5296–5305, 2017. 2

[36] 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. 3

[37] 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. 3

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 3, 6, 7

[39] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8430–8439, 2019. 1, 2, 3

[40] 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. 2

[41] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *European conference on computer vision*, pages 467–482. Springer, 2016. 5

[42] Yosuke Shinya. Usb: Universal-scale object detection benchmark. *arXiv preprint arXiv:2103.14027*, 2021. 2

[43] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020. 2, 7

[44] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 2

[45] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris M Kitani. Rethinking transformer-based set prediction for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3611–3620, 2021. 3

[46] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11662–11671, 2020. 5

[47] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 3

[48] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 2

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3

[50] Jiaqi Wang, Wenwei Zhang, Yuhang Zang, Yuhang Cao, Jiangmiao Pang, Tao Gong, Kai Chen, Ziwei Liu, Chen Change Loy, and Dahua Lin. Seesaw loss for long-tailed instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9695–9704, 2021. 5

[51] Li Wang, Li Zhang, Yi Zhu, Zhi Zhang, Tong He, Mu Li, and Xiangyang Xue. Progressive coordinate transforms for monocular 3d object detection. In *NeurIPS*, 2021. 3

[52] Xudong Wang, Zhaowei Cai, Dashan Gao, and Nuno Vasconcelos. Towards universal object detection by domain attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7289–7298, 2019. 1, 2, 4

[53] Xiaoyu Wang, Ming Yang, Shenghuo Zhu, and Yuanqing Lin. Regionlets for generic object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 17–24, 2013. 2

[54] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. 2019. *URL https://github. com/facebookresearch/detectron2*, 2(3), 2019. 6

[55] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. *arXiv preprint arXiv:2106.09018*, 2021. 1, 7

[56] Cheng Zhang, Tai-Yu Pan, Yandong Li, Hexiang Hu, Dong Xuan, Soravit Changpinyo, Boqing Gong, and Wei-Lun Chao. Mosaicos: A simple and effective use of object-centric images for long-tailed object detection. *arXiv preprint arXiv:2102.08884*, 2021. 5, 6

[57] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 6

[58] Xiangyun Zhao, Samuel Schulter, Gaurav Sharma, Yi-Hsuan Tsai, Manmohan Chandraker, and Ying Wu. Object detection with a unified label space from multiple datasets. In *European Conference on Computer Vision*, pages 178–193. Springer, 2020. 1, 2, 4, 5

[59] Minghang Zheng, Peng Gao, Xiaogang Wang, Hongsheng Li, and Hao Dong. End-to-end object detection with adaptive clustering transformer. *arXiv preprint arXiv:2011.09315*, 2020. 3

[60] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. *arXiv preprint arXiv:2103.07461*, 2021. 3, 6

[61] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Simple multi-dataset detection. *arXiv preprint arXiv:2102.13086*, 2021. 1, 2, 4

[62] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 3

[63] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 3

[64] Yi Zhu, Zhongyue Zhang, Chongruo Wu, Zhi Zhang, Tong He, Hang Zhang, R Manmatha, Mu Li, and Alexander J Smola. Improving semantic segmentation via efficient self-training. *IEEE transactions on pattern analysis and machine intelligence*, 2021. 5

[65] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. Rethinking pre-training and self-training. *arXiv preprint arXiv:2006.06882*, 2020. 7