USB: Universal-Scale Object Detection Benchmark

Yosuke Shinya*
https://shinya7y.github.io/
independent researcher
Tokyo, Japan

Hinagiku Kenzan! ©Minene Sakurano

Figure 1: Universal-scale object detection. For realizing human-level perception, object detection systems must detect both tiny and large objects, even if they are out of natural image domains. To this end, we introduce the *Universal-Scale object detection Benchmark (USB)* that consists of the COCO dataset (left), Waymo Open Dataset (middle), and Manga109-s dataset (right).

Abstract

Benchmarks, such as COCO, play a crucial role in object detection. However, existing benchmarks are insufficient in scale variation, and their protocols are inadequate for fair comparison. In this paper, we introduce the Universal-Scale object detection Benchmark (USB). USB has variations in object scales and image domains by incorporating COCO with the recently proposed Waymo Open Dataset and Manga109-s dataset. To enable fair comparison and inclusive research, we propose training and evaluation protocols. They have multiple divisions for training epochs and evaluation image resolutions, like weight classes in sports, and compatibility across training protocols, like the backward compatibility of the Universal Serial Bus. Specifically, we request participants to report results with not only higher protocols (longer training) but also lower protocols (shorter training). Using the proposed benchmark and protocols, we conducted extensive experiments using 15 methods and found weaknesses of existing COCO-biased methods. The code is available at `https://github.com/shinya7y/UniverseNet`.

1 Introduction

Humans can detect various objects. See Figure 1. One can detect close equipment in everyday scenes, far vehicles in traffic scenes, and texts and persons in manga (Japanese comics). If computers can automatically detect various objects, they will yield significant benefits

*This work was done independently of the author's employer.

to humans. For example, they will help impaired people and the elderly, save lives by autonomous driving, and provide safe entertainment during pandemics by automatic translation.

Researchers have pushed the limits of object detection systems by establishing datasets and benchmarks [36]. One of the most important milestones is PASCAL VOC [15]. It has enabled considerable research on object detection, leading to the success of deep learning-based methods and successor datasets such as ImageNet [49] and COCO [33]. Currently, COCO serves as *the* standard dataset and benchmark for object detection because it has several advantages over PASCAL VOC [15]. COCO contains more images, categories, and objects (especially small objects) in their natural context [33]. Using COCO, researchers can develop and evaluate methods for multi-scale object detection. However, the current object detection benchmarks, especially COCO, have the following two problems.

Problem 1: Variations in object scales and image domains remain limited. To realize human-level perception, computers must handle various object scales and image domains as humans can. Among various domains [61], the traffic and artificial domains have extensive scale variations (see Sec. 3.3). COCO is far from covering them. Nevertheless, the current computer vision community is overconfident in COCO results. For example, most studies on state-of-the-art methods in 2020 only report COCO results [9, 10, 29, 31, 60, 64] or those for bounding box object detection [4, 14, 46, 57]. Readers cannot assess whether these methods are specialized for COCO or generalizable to other datasets and domains.

Problem 2: Protocols for training and evaluation are not well established. There are standard experimental settings for the COCO benchmark [8, 20, 31, 34, 35, 59, 64]. Many studies train detectors within 24 epochs using a learning rate of 0.01 or 0.02 and evaluate them on images within 1333×800. These settings are not obligations but non-binding agreements for fair comparison. Some studies do not follow the settings for accurate and fast detectors[1]. Their abnormal and scattered settings hinder the assessment of the most suitable method. Furthermore, by "buying stronger results" [50], they build a barrier for those without considerable funds to develop and train detectors.

This study makes the following two contributions to resolve the problems.

Contribution 1: We introduce the *Universal-Scale object detection Benchmark (USB)* that consists of three datasets. In addition to COCO, we selected the Waymo Open Dataset [54] and Manga109-s [3, 41] to cover various object scales and image domains. They are the largest public datasets in their domains and enable reliable comparisons. To the best of our knowledge, USB is the first benchmark beyond COCO that evaluates finer scale-wise metrics across multiple domains. We conducted extensive experiments using 15 methods and found weaknesses of existing COCO-biased methods.

Contribution 2: We established the *USB protocols* for fair training and evaluation, inspired by weight classes in sports and the backward compatibility of the Universal Serial Bus. Specifically, USB protocols enable fair and easy comparisons by defining multiple divisions for training epochs and evaluation image resolutions. Furthermore, we introduce compatibility across training protocols by requesting participants to report results with not only higher protocols (longer training) but also lower protocols (shorter training). To the best of our knowledge, our training protocols are the first ones that allow for both fair comparisons with shorter training and strong results with longer training. Our protocols promote inclusive, healthy, and sustainable object detection research.

[1] YOLOv4 was trained for 273 epochs [4], DETR for 500 epochs [7], EfficientDet-D6 for 300 epochs [57], and EfficientDet-D7x for 600 epochs [58]. SpineNet uses a learning rate of 0.28 [14], and YOLOv4 uses a searched learning rate of 0.00261 [4]. EfficientDet finely changes the image resolution from 512×512 to 1536×1536 [57].

2 Related Work

Multi-scale object detection. Detecting multi-scale objects is a fundamental challenge in object detection [5, 36]. Various components have been improved, including backbones and modules [11, 16, 22, 24, 56], necks [12, 34, 57, 60], heads and training sample selection [37, 47, 64], and multi-scale training and testing [48, 53, 64] (see Supp. B for details). Unlike most prior studies, we analyzed their methods across various object scales and image domains through the proposed benchmark.

Single-domain benchmarks. There are numerous object detection benchmarks that specialize in a specific domain or consider natural images as a single generic domain. For specific (category) object detection, recent benchmarks such as WIDER FACE [62] and TinyPerson [63] contain tiny objects. For autonomous driving, KITTI [18] and Waymo Open Dataset [54] mainly evaluate three categories (car, pedestrian, and cyclist) in their leaderboards. For generic object detection, PASCAL VOC [15] and COCO [33] include 20 and 80 categories, respectively. The number of categories has been further expanded by recent benchmarks, such as Open Images [30], Objects365 [51], and LVIS [21]. The above datasets comprise photographs, whereas Clipart1k, Watercolor2k, Comic2k [27], and Manga109-s [3, 41] comprise artificial images. Although Waymo Open Dataset [54] and Manga109-s [3, 41] have extensive scale variations (see Sec. 3.3), scale-wise metrics have not been evaluated [43, 54]. Unlike the above benchmarks, our USB consists of multiple domains and contains many instances in both photographs and artificial images, and we can evaluate the generalization ability of methods.

Cross-domain benchmarks. To avoid performance drops in target domains without labor-intensive annotations, many studies have tackled domain adaptation of object detection [44]. Some datasets have been proposed for this setting [27, 28]. Typically, there is a strong constraint to share a label space. Otherwise, special techniques are needed for training, architectures, unified label spaces [65, 66], and partial or open-set domain adaptation [44]. In contrast, we focus on fully supervised object detection, which allows us to analyze many standard detectors.

Multi/universal-domain benchmarks. Even if target datasets have annotations for training, detectors trained and evaluated on a specific dataset may perform worse on other datasets or domains. To address this issue, some benchmarks consist of multiple datasets. In the Robust Vision Challenge (RVC) 2020 [1], detectors were evaluated on three datasets in the natural and traffic image domains. A few studies have explored the two domains by enriching RVC [66] or making a unique combination [65], although they focus on methods for unified detectors. For *universal-domain* object detection, the Universal Object Detection Benchmark (UODB) [61] comprises 11 datasets in the natural, traffic, aerial, medical, and artificial image domains. Although it is suitable for evaluating detectors in various domains, variations in object scales are limited. Unlike UODB, our USB focuses on *universal-scale* object detection. The datasets in USB contain more instances, including tiny objects, than the datasets used in UODB.

Criticism of experimental settings. For fair, inclusive, and efficient research, many studies have criticized experimental settings (*e.g.*, [42, 50]). These previous studies do not propose fair and practical protocols for object detection benchmarks. As discussed in Sec. 1, the current object detection benchmarks allow extremely unfair settings (*e.g.*, 25$\times$ epochs). We resolved this problem by establishing protocols for fair training and evaluation.

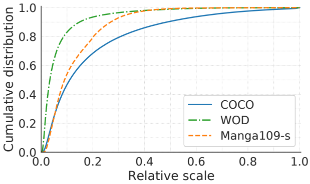

Figure 2: Distributions of objects' relative scale [26, 53]. USB covers diverse scale variations.

Benchmark	Dataset	Boxes	Images	B/I	Scale variation†
USB (Ours)	COCO [33]	**897 k** (**3.1**×)	**123 k**	**7.3**	88.8 (1.0×)
	WOD [54] v1.2 `f0`	**1.0 M** (**29**×)	**100 k**	**10.0**	**96.7** (**5.8**×)
	Manga109-s [3, 41]	**401 k** (**63**×)	**8.2 k**	**49.2**	**28.6** (**1.5**×)
UODB [61]	COCO [33] `val2014`	292 k	41 k	7.2	**89.6**
	KITTI [18]	35 k	7.5 k	4.7	16.6
	Comic2k [27]	6.4 k	2.0 k	3.2	19.1

Table 1: Statistics of datasets in USB and counterpart datasets in UODB [61]. Values are based on publicly available annotations. B/I: Average number of boxes per image. †: Calculated by the ratio of the 99 percentile to 1 percentile of relative scale.

3 Benchmark Protocols of USB

Here, we present the principle, datasets, protocols, and metrics of USB. See Supp. C for additional information.

3.1 Principle

We focus on the *Universal-Scale Object Detection (USOD)* task that aims to detect various objects in terms of object scales and image domains. Unlike separate discussions for multi-scale object detection (Sec. 2) and universal (-domain) object detection [61], USOD does not ignore the relation between scales and domains (Sec. 3.3).

For various applications and users, benchmark protocols should cover from short to long training and from small to large test scales. On the other hand, they should not be scattered for meaningful benchmarks. To satisfy the conflicting requirements, we define multiple divisions for training epochs and evaluation image resolutions. Furthermore, we urge participants who have access to extensive computational resources to report results with standard training settings. This request enables fair comparison and allows many people to develop and compare object detectors.

3.2 Definitions of Object Scales

Following [63], we consider two types of object scales. The absolute scale is calculated as $\sqrt{wh}$, where w and h denote the object's width and height, respectively. The relative scale is calculated as $\sqrt{\frac{wh}{WH}}$, where W and H denote the image's width and height, respectively.

3.3 Datasets

To establish USB, we selected the COCO [33], Waymo Open Dataset (WOD) [54], and Manga109-s (M109s) [3, 41]. WOD and M109s are the largest public datasets with many small objects in the traffic and artificial domains, respectively. Object scales in these domains vary significantly with distance and viewpoints, unlike those in the medical and aerial domains[2]. USB covers diverse scale variations qualitatively (Figure 1) and quantitatively (Figure 2). As shown in Table 1, these datasets contain more instances and larger scale variations [53] than their counterpart datasets in UODB [61]. USOD needs to evaluate detectors

[2]Aerial datasets contain abundant small objects but scarce large ones (see Table 4 in [13]). WOD has larger scale variation by distance variation, where 1% of objects are larger than 1/4 of the image area.

Protocol	Fair	Suitable for each model	Strong results	Selectable divisions	Comparable across divisions
A) Standard (short) training	✓				
B) Lawless (no regulations)		✓	✓		
C) Ours w/o compatibility	✓	✓	✓	✓	
D) Ours	✓	✓	✓	✓	✓

Table 2: Comparison of training protocols.

on datasets with many instances because more instances enable more reliable comparisons of scale-wise metrics.

For the first dataset, we adopted the COCO dataset [33]. COCO contains natural images of everyday scenes collected from the Internet. Annotations for 80 categories are used in the benchmark. As shown in Figure 1 (left), object scales mainly depend on categories and distance. Although COCO contains objects smaller than those of PASCAL VOC [15], objects in everyday scenes (especially indoor scenes) are relatively large. Since COCO is the current standard dataset for multi-scale object detection, we adopted the same training split `train2017` as the COCO benchmark to eliminate the need for retraining across benchmarks. We adopted the `val2017` split (also known as `minival`) as the test set.

For the second dataset, we adopted the WOD, which is a large-scale, diverse dataset for autonomous driving [54] with many annotations for tiny objects (Figure 2). The images were recorded using five high-resolution cameras mounted on vehicles. As shown in Figure 1 (middle), object scales vary mainly with distance. The full data splits of WOD are too large for benchmarking methods. Thus, we extracted 10% size subsets from the predefined training split (798 sequences) and validation split (202 sequences) [54]. Specifically, we extracted splits based on the ones place of the frame index (frames 0, 10, ..., 190) in each sequence. We call the subsets `f0train` and `f0val` splits. Each sequence in the splits contains ∼20 frames (20 s, 1 Hz), and each frame contains five images for five cameras. We used three categories (vehicle, pedestrian, and cyclist) following the official *ALL_NS* setting [2] used in WOD competitions.

For the third dataset, we adopted the M109s [3, 41]. M109s contains artificial images of manga (Japanese comics) and annotations for four categories (body, face, frame, and text). Many characteristics differ from those of natural images. Most images are grayscale. The objects are highly overlapped [43]. As shown in Figure 1 (right), object scales vary unrestrictedly with viewpoints and page layouts. Small objects differ greatly from downsampled versions of large objects because small objects are drawn with simple lines and points. For example, small faces look like a sign (∵). This characteristic may ruin techniques developed mainly for natural images. We carefully selected 68, 4, and 15 volumes for training, validation, and testing splits, and we call them the `68train`, `4val`, and `15test`, respectively.

3.4 Motivation of Training Protocols

We describe the motivation of our training protocols with Table 2, which compares existing protocols (A and B) and novel protocols (C and D). Protocol A is the current standard training protocol within 24 epochs, popularized by successive detectors, Detectron [20], and MMDetection [8]. This protocol is fair but not suitable for slowly convergent models (*e.g.*, DETR [7]). Protocol B is lawless without any regulations. Participants can train their models with arbitrary settings suitable for them, even if they are unfair settings (*e.g.*, standard training for existing methods and longer training for proposed ones). Since object

Protocol	Max epoch	AHPO	Compatibility	Example
USB 1.0	24	✗	—	2× schedule [20, 23]
USB 2.0	73	✗	USB 1.0	6× schedule [23]
USB 3.0	300	✗	USB 1.0, 2.0	EfficientDet-D6 [57]
USB 3.1	300	✓	USB 1.0, 2.0, 3.0	YOLOv4 [1]
Freestyle	∞	✓	—	EfficientDet-D7x [58]

Table 3: USB training protocols. AHPO: Aggressive hyperparameter optimization.

Protocol	Max reso.	Typical scale	Reference
Standard USB	1,066,667	1333× 800	Popular in COCO [8, 20, 33]
Mini USB	262,144	512× 512	Popular in VOC [15, 37]
Micro USB	50,176	224× 224	Popular in ImageNet [22, 49]
Large USB	2,457,600	1920×1280	WOD front cameras [54]
Huge USB	7,526,400	3360×2240	WOD methods ([26], ours)
Freestyle	∞	—	—

Table 4: USB evaluation protocols.

detectors can achieve high accuracy with long training schedules and strong data augmentation [14, 19, 58], participants can buy stronger results [50].

Since both existing protocols A and B have advantages and disadvantages, we considered novel protocols to bridge them. We first defined multiple divisions for training epochs, inspired by weight classes in sports. This Protocol C enables fair comparison in each division. Participants can select divisions according to their purposes and resources. However, we cannot compare models across divisions. To resolve this, we propose Protocol D by introducing backward compatibility like the Universal Serial Bus. As described above, *our protocols introduce a completely different paradigm from existing limited or unfair protocols.*

The training protocols mainly target resource-intensive factors that can increase the required resources 10 times or more. This decision improves fairness without obstructing novel methods and practical settings that researchers can adopt without many resources. We do not adopt factors that have large overlaps with inference efficiency, which has been considered in many previous studies.

3.5 Training Protocols

For fair training, we propose the *USB training protocols* shown in Table 3. By analogy with the backward compatibility of the Universal Serial Bus[3], USB training protocols emphasize compatibility between protocols. Importantly, *participants should report results with not only higher protocols but also lower protocols.* For example, when a participant trains a model for 150 epochs with standard hyperparameters, it corresponds to USB 3.0. The participant should also report the results of models trained for 24 and 73 epochs in a paper. This reveals the effectiveness of the method by ablating the effect of long training. The readers of the paper can judge whether the method is useful for standard epochs. Since many people do not have access to extensive computational resources, such information is important.

The number of maximum epochs for USB 1.0 is 24, following a popular setting in COCO [8, 20]. We adopted 73 epochs for USB 2.0, where models trained from scratch can catch up with those trained from ImageNet pre-trained models [23]. This serves as a guideline for comparison between models with and without pre-training, although perfectly fair comparisons are impossible considering the large differences caused by pre-training [52]. We adopted 300 epochs for USB 3.x such that YOLOv4 [1] and most EfficientDet models [58] correspond to this protocol. Models trained for more than 300 epochs are regarded as Freestyle. They are not suitable for benchmarking methods, although they may push the empirical limits of detectors [7, 58]. The correspondences between Tables 2 and 3 are as follows: Protocol A corresponds to only USB 1.0; Protocol B corresponds to only Freestyle; Protocol C corresponds to all protocols (divisions) in Table 3 without compatibility; and Protocol D corresponds to all protocols (divisions) in Table 3 with compatibility.

[3] Higher protocols can adapt the data transfer rate to lower protocols.

In addition to long training schedules, hyperparameter optimization is resource-intensive. If authors of a paper fine-tune hyperparameters for their architecture, other people without sufficient computational resources cannot compare methods fairly. For hyperparameters that need to be tuned exponentially, such as learning rates and $1-m$ where m denotes momentum, the minimum ratio of hyperparameter choices should be greater than or equal to 2 (*e.g.*, choices $\{0.1, 0.2, 0.4, 0.8, ...\}$, $\{0.1, 0.2, 0.5, 1.0, ...\}$, and $\{0.1, 0.3, 1.0, ...\}$). For hyperparameters that need to be tuned linearly, the number of choices should be less than or equal to 11 (*e.g.*, choices $\{0.0, 0.1, 0.2, ..., 1.0\}$). When participants perform aggressive hyperparameter optimization (AHPO) by manual fine-tuning or automatic algorithms, 0.1 is added to their number of protocols. They should report both results with and without AHPO. To further improve fairness without sacrificing the protocols' simplicity, we consider it a kind of AHPO to use data augmentation techniques that more than double the time per epoch.

For models trained with annotations other than 2D bounding boxes (*e.g.*, segmentation, keypoint, caption, and point cloud), 0.5 is added to their number of protocols. Participants should also report results without such annotations if possible for their algorithms.

For ease of comparison, we limit the pre-training datasets to the three datasets and ImageNet-1k (ILSVRC 1,000-class classification) [49]. Other datasets are welcome only when the results with and without additional datasets are reported. Participants should describe how to use the datasets (*e.g.*, fine-tuning models on WOD and M109s from COCO pre-trained models, or training a single model jointly [61, 66] on the three datasets).

3.6 Evaluation Protocols

For fair evaluation, we propose the *USB evaluation protocols* shown in Table 4. By analogy with the size variations of the Universal Serial Bus connectors for various devices, USB evaluation protocols have variations in test image scales for various devices and applications.

The maximum resolution for Standard USB follows the popular test scale of 1333×800 in the COCO benchmark [8, 20]. For Mini USB, we limit the resolution based on 512×512. This resolution is popular in the PASCAL VOC benchmark [15, 37], which contains small images and large objects. It is also popular in real-time detectors [4, 57]. We adopted a further small-scale 224×224 for Micro USB. This resolution is popular in ImageNet classification [22, 49]. Although small object detection is extremely difficult, it is suitable for low-power devices. Additionally, this protocol enables people to manage object detection tasks using one or few GPUs. To cover larger test scales than Standard USB, we define Large USB and Huge USB based on WOD resolutions (see Supp. E for the top methods). Although larger inputs (regarded as Freestyle) may be preferable for accuracy, excessively large inputs reduce the practicality of detectors.

In addition to test image scales, the presence and degree of Test-Time Augmentation (TTA) make large differences in accuracy and inference time. When using TTA, participants should report its details (including scales of multi-scale testing) and results without TTA.

3.7 Evaluation Metrics

We mainly use the COCO metrics [32, 33] to evaluate the performance of detectors on each dataset. We provide data format converters for WOD[4] and M109s[5]. The COCO-

[4] https://github.com/shinya7y/WaymoCOCO
[5] https://github.com/shinya7y/manga109api

style AP (CAP) for a dataset d is calculated as $\mathrm{CAP}_d = \frac{1}{|T|}\sum_{t \in T}\frac{1}{|C_d|}\sum_{c \in C_d}\mathrm{AP}_{t,c}$, where $T = \{0.5, 0.55, ..., 0.95\}$ denotes the predefined 10 IoU thresholds, C_d denotes categories in the dataset d, and $\mathrm{AP}_{t,c}$ denotes Average Precision (AP) for an IoU threshold t and a category c. For detailed analysis, five additional AP metrics (averaged over categories) are evaluated. AP_{50} and AP_{75} denote AP at single IoU thresholds of 0.5 and 0.75, respectively. AP_S, AP_M, and AP_L are variants of CAP, where target objects are limited to small (area $\leq 32^2$), medium ($32^2 \leq$ area $\leq 96^2$), and large ($96^2 \leq$ area) objects, respectively. The area is measured using mask annotations for COCO and bounding box annotations for WOD and M109s.

As the primary metric for USB, we use the mean COCO-style AP (mCAP) averaged over all datasets D as $\mathrm{mCAP} = \frac{1}{|D|}\sum_{d \in D}\mathrm{CAP}_d$. Since USB adopts the three datasets described in Sec. 3.3, $\mathrm{mCAP} = (\mathrm{CAP}_{\mathrm{COCO}} + \mathrm{CAP}_{\mathrm{WOD}} + \mathrm{CAP}_{\mathrm{M109s}})/3$. Similarly, we define five metrics from AP_{50}, AP_{75}, AP_S, AP_M, and AP_L by averaging them over the datasets.

The three COCO-style scale-wise metrics (AP_S, AP_M, and AP_L) are too coarse for detailed scale-wise analysis. They confuse objects of significantly different scales. For example, the absolute scale of a large object might be 100 or 1600. Thus, we introduce finer scale-wise metrics. We define the *Absolute Scale AP (ASAP)* and *Relative Scale AP (RSAP)* using exponential thresholds. ASAP partitions object scales based on absolute scales $(0, 8, 16, 32, ..., 1024, \infty)$, while RSAP partitions object scales based on relative scales $(0, \frac{1}{256}, \frac{1}{128}, ..., \frac{1}{2}, 1)$. We call the partitions by their maximum scales.

For ease of quantitative evaluation, we limit the number of detections per image to 100 across all categories [32]. For qualitative evaluation, participants may raise the limit to 300 because 1% of images in the M109s `15test` set contain more than 100 annotations.

4 Experiments

Here, we present benchmark results and analysis on USB. See Supp. E for the details of the experimental settings and results, including additional analysis and ablation studies.

4.1 Experimental Settings

We compared and analyzed 15 methods. With the ResNet-50-B [22, 24] backbone, we compared popular baseline methods: (1) Faster R-CNN [47] with FPN [34], (2) Cascade R-CNN [6], (3) RetinaNet [35], (4) ATSS [64], (5) GFL [31], (6) DETR [7], (7) Deformable DETR [67], and (8) Sparse R-CNN [55]. With ATSS [64], we compared recent representative backbones and necks: (9) Swin-T [38], (10) ConvNeXt-T [39], (11) SEPC without iBN [60], and (12) DyHead [12]. For a strong baseline, we trained (13) YOLOX-L [17], which adopts strong data augmentation. We designed two additional detectors for USOD by collecting methods for multi-scale object detection. (14) UniverseNet: ATSS [64] with SEPC (without iBN) [60], Res2Net-50-v1b [16], Deformable Convolutional Networks (DCN) [11], and multi-scale training. (15) UniverseNet-20.08: A variant of UniverseNet designed around August 2020 with GFL [31], SyncBN [45], iBN [60], and the light use of DCN [11, 60]. See Supp. D for the details of the methods and architectures used in UniverseNets.

Hyperparameters	COCO	WOD	M109s
Learning rate for multi-stage detectors	0.02	0.02	0.16
Learning rate for single-stage detectors	0.01	0.01	0.08
Test scale	1333×800	1248×832	1216×864

Hyperparam.	Common
Epoch	12
Batch size	16
Momentum	0.9
Weight decay	10^{-4}

Table 5: Default hyperparameters. See Supp. E for exceptions.

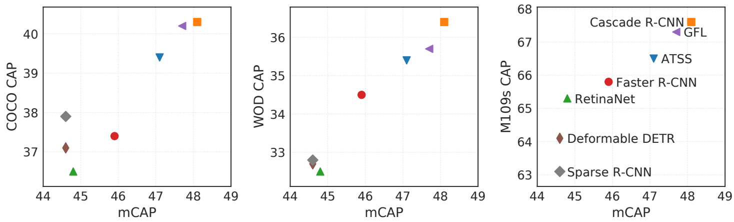

Figure 3: Correlation between mCAP and CAP on each dataset.

Method	mCAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	COCO	WOD	M109s
Faster R-CNN [47]	45.9	68.2	49.1	15.2	38.9	62.5	37.4	34.5	65.8
Cascade R-CNN [6]	**48.1**	**68.5**	**51.5**	15.6	**41.3**	**65.9**	**40.3**	**36.4**	**67.6**
RetinaNet [35]	44.8	66.0	47.4	12.9	37.3	62.6	36.5	32.5	65.3
ATSS [64]	47.1	68.0	50.2	15.5	39.5	64.7	39.4	35.4	66.5
GFL [31]	47.7	68.3	50.6	**15.8**	39.9	65.8	40.2	35.7	67.3
DETR [7]	23.7	45.9	21.6	2.8	13.8	42.1	22.2	17.8	31.2
Deform. DETR [67]	44.6	67.0	47.3	13.8	36.1	62.6	37.1	32.7	64.1
Sparse R-CNN [55]	44.6	65.4	46.9	14.4	35.8	63.0	37.9	32.8	63.1

Table 6: Results of popular baseline methods.

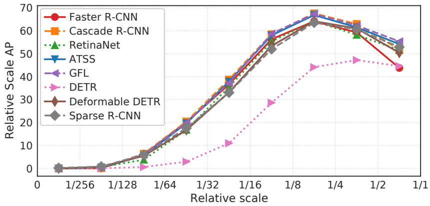

Figure 4: Relative Scale AP of popular baseline methods.

Backbone	mCAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	COCO	WOD	M109s
ResNet-50-B [24]	47.1	68.0	50.2	15.5	39.5	64.7	39.4	35.4	66.5
Swin-T [38]	49.0	70.6	52.0	17.2	41.8	67.2	43.7	37.2	66.2
ConvNeXt-T [39]	**50.4**	**71.8**	**53.7**	**17.3**	**43.0**	**69.0**	**45.5**	**38.3**	**67.4**

Table 7: ATSS [64] with different backbones.

Neck	mCAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	COCO	WOD	M109s
FPN [34]	47.1	68.0	50.2	15.5	39.5	64.7	39.4	35.4	66.5
FPN+SEPC [60]	48.1	68.5	51.2	15.5	40.5	66.8	42.1	35.0	67.1
FPN+DyHead [12]	**49.4**	**69.8**	**52.9**	**16.8**	**43.0**	**67.8**	**43.3**	**37.1**	**67.9**

Table 8: ATSS [64] with different necks.

Method	mCAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	COCO	WOD	M109s
YOLOX-L [17]	51.0	72.6	54.7	**21.2**	**45.9**	65.0	41.1	**41.6**	**70.2**
UniverseNet	51.4	72.1	55.1	18.4	45.0	70.7	46.7	38.6	68.9
UniverseNet-20.08	**52.1**	**72.9**	**55.5**	19.2	45.8	**70.8**	**47.5**	39.0	69.9

Table 9: Results of strong baseline methods.

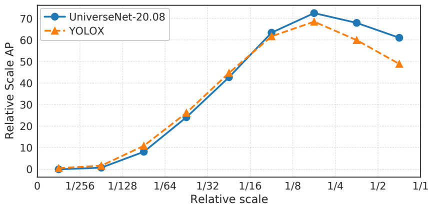

Figure 5: Relative Scale AP of strong baseline methods.

Our code is built on MMDetection [8]. We trained models with Stochastic Gradient Descent (SGD) or AdamW [40]. COCO models other than YOLOX [17] were fine-tuned from ImageNet [49] pre-trained backbones. We trained the models for WOD and M109s from the corresponding COCO pre-trained models (some COCO models from MMDetection [8]). The default hyperparameters are listed in Table 5. Test scales were determined within the Standard USB protocol, considering the typical aspect ratio of the images in each dataset.

4.2 Benchmark Results on USB

Main results. We trained and evaluated the eight popular methods on USB. All the methods follow the Standard USB 1.0 protocol. The results are shown in Table 6. Cascade R-CNN [6] achieves the highest results in almost all metrics. The accuracy of DETR [7] is low by a large margin. We show the correlation between mCAP and CAP on each dataset in Figure 3. Faster R-CNN [47] is underestimated on COCO. Although Sparse R-CNN [55] is much more accurate than RetinaNet [35] on COCO, this is not true on the other datasets. These results show the limitation of benchmarking with COCO only.

Backbones and necks. Tables 7 and 8 show the comparison results of the backbones and necks, respectively. Swin-T [38] shows lower AP than ResNet-50-B [22, 24] on M109s. SEPC [60] deteriorates WOD CAP.

Strong baselines. Table 9 shows the results of the three strong baselines. UniverseNet-20.08 achieves the highest mCAP of 52.1%. YOLOX-L [17] shows better results on WOD and M109s, which contain many small objects, possibly due to better AP_S.

Scale-wise AP. We show RSAP on USB in Figures 4 and 5. Since the proposed metrics partition object scales evenly-spaced exponentially, we can confirm the continuous change. RSAP does not increase monotonically but rather decreases at relative scales greater than $1/4$. We cannot find this weakness from the coarse COCO-style scale-wise AP in Table 6 *etc*. The difficulty of very large objects may be caused by truncation or unusual viewpoints [25]. The results also show that different methods are good at different scales. We need further analysis in future research to develop methods that can detect both tiny and large objects.

Details on each dataset. We show detailed results on each dataset in Supp. E. AP_S on WOD is at most 12.0%, which is much lower than AP_S on COCO. This highlights the limitation of COCO and current detectors. Adding SEPC [60] to ATSS [64] decreases all metrics on WOD except for AP_L. We found that this reduction does not occur at large test scales in higher USB evaluation protocols. Improvements by ATSS [64] on M109s are smaller than those on COCO and WOD due to the drop of face AP. We conjecture that this phenomenon comes from the domain differences discussed in Sec. 3.3 and prior work [43].

Qualitative results. We show some qualitative results of the best detector (UniverseNet-20.08) in Figure 1. Although most detections are accurate, it still suffers from classification error, localization error, and missing detections of tiny vehicles and small manga faces.

5 Conclusions and Discussions

We introduced USB, a benchmark for universal-scale object detection. To resolve unfair comparisons in existing benchmarks, we established USB training/evaluation protocols. With the benchmark, we found weaknesses in existing methods to be addressed in future research.

There are several limitations to this work. (1) USB has imbalances in domains and categories because it depends on the existing datasets that have large scale variations. It will be an important direction to construct a well-balanced and more comprehensive benchmark that contains more domains and categories. (2) The architectures and results of the 15 methods are still biased toward COCO due to development and pre-training on COCO. Less biased and more universal detectors should be developed in future research. (3) We could not train detectors with higher protocols than USB 1.0 due to limited resources. Although the compatibility enables comparison in low protocols, still only well-funded researchers can compare detectors in high protocols. Other efforts are also needed to ensure fairness and inclusion in research. See Supp. A for discussion on other limitations and research ethics.

The current computer vision community places a high value on state-of-the-art results. Thus, there is a large incentive to make unfair comparisons for overly accurate results, like DETR [7] and EfficientDet [57]. We need to create a system that emphasizes fair comparisons. To improve effectiveness in broad areas, creating a checklist that can be incorporated into author/reviewer guidelines is a promising future direction. We believe that our work is an important step toward realizing fair and inclusive research by connecting various experimental settings.

Acknowledgments. We are grateful to Dr. Hirokatsu Kataoka for helpful comments. We thank all contributors for the datasets and software libraries. The original image of Figure 1 (left) is *satellite office* by Taiyo FUJII (CC BY 2.0).

References

[1] Robust Vision Challenge 2020. http://www.robustvision.net/, Accessed on Nov. 8, 2020.

[2] Waymo Open Dataset 2D detection leaderboard. https://waymo.com/open/challenges/2d-detection/, Accessed on June 18, 2020.

[3] Kiyoharu Aizawa, Azuma Fujimoto, Atsushi Otsubo, Toru Ogawa, Yusuke Matsui, Koki Tsubota, and Hikaru Ikuta. Building a manga dataset "Manga109" with annotations for multimedia applications. *IEEE MultiMedia*, 2020.

[4] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. YOLOv4: Optimal speed and accuracy of object detection. *arXiv:2004.10934*, 2020.

[5] Zhaowei Cai. *Towards Universal Object Detection*. PhD thesis, UC San Diego, 2019.

[6] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *CVPR*, 2018.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[8] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv:1906.07155*, 2019.

[9] Yihong Chen, Zheng Zhang, Yue Cao, Liwei Wang, Stephen Lin, and Han Hu. RepPoints v2: Verification meets regression for object detection. In *NeurIPS*, 2020.

[10] Cheng Chi, Fangyun Wei, and Han Hu. RelationNet++: Bridging visual representations for object detection via transformer decoder. In *NeurIPS*, 2020.

[11] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017.

[12] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *CVPR*, 2021.

[13] Jian Ding, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Michael Ying Yang, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Object detection in aerial images: A large-scale benchmark and challenges. *TPAMI*, 2021.

[14] Xianzhi Du, Tsung-Yi Lin, Pengchong Jin, Golnaz Ghiasi, Mingxing Tan, Yin Cui, Quoc V. Le, and Xiaodan Song. SpineNet: Learning scale-permuted backbone for recognition and localization. In *CVPR*, 2020.

[15] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes challenge: A retrospective. *IJCV*, 2015.

[16] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2Net: A new multi-scale backbone architecture. *TPAMI*, 2021.

[17] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding YOLO series in 2021. *arXiv:2107.08430*, 2021.

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012.

[19] 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 *CVPR*, 2021.

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

[21] Agrim Gupta, Piotr Dollár, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[23] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking ImageNet pre-training. In *ICCV*, 2019.

[24] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *CVPR*, 2019.

[25] Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. In *ECCV*, 2012.

[26] Zehao Huang, Zehui Chen, Qiaofei Li, Hongkai Zhang, and Naiyan Wang. 1st place solutions of Waymo Open Dataset challenge 2020 – 2D object detection track. *arXiv:2008.01365*, 2020.

[27] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *CVPR*, 2018.

[28] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In *ICRA*, 2017.

[29] Kang Kim and Hee Seok Lee. Probabilistic anchor assignment with IoU prediction for object detection. In *ECCV*, 2020.

[30] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

[31] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized Focal Loss: Learning qualified and distributed bounding boxes for dense object detection. In *NeurIPS*, 2020.

[32] Tsung-Yi Lin, Piotr Dollár, et al. COCO API. https://github.com/cocodataset/cocoapi, Accessed on Nov. 8, 2020.

[33] 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 *ECCV*, 2014.

[34] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[35] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[36] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *IJCV*, 2020.

[37] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016.

[38] 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. In *ICCV*, 2021.

[39] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022.

[40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[41] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using Manga109 dataset. *Multimedia Tools and Applications*, 2017.

[42] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *ECCV*, 2020.

[43] Toru Ogawa, Atsushi Otsubo, Rei Narita, Yusuke Matsui, Toshihiko Yamasaki, and Kiyoharu Aizawa. Object detection for comics using Manga109 annotations. *arXiv:1803.08670*, 2018.

[44] Poojan Oza, Vishwanath A. Sindagi, Vibashan VS, and Vishal M. Patel. Unsupervised domain adaptation of object detectors: A survey. *arXiv:2105.13502*, 2021.

[45] 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 *CVPR*, 2018.

[46] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. DetectoRS: Detecting objects with recursive feature pyramid and switchable atrous convolution. *arXiv:2006.02334*, 2020.

[47] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[48] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *TPAMI*, 1998.

[49] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[50] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green AI. *Communications of the ACM*, 2020.

[51] 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 *ICCV*, 2019.

[52] Yosuke Shinya, Edgar Simo-Serra, and Taiji Suzuki. Understanding the effects of pre-training for object detectors via eigenspectrum. In *ICCV Workshop on Neural Architects*, 2019.

[53] Bharat Singh and Larry S. Davis. An analysis of scale invariance in object detection – SNIP. In *CVPR*, 2018.

[54] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo Open Dataset. In *CVPR*, 2020.

[55] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, and Ping Luo. Sparse R-CNN: End-to-end object detection with learnable proposals. In *CVPR*, 2021.

[56] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[57] Mingxing Tan, Ruoming Pang, and Quoc V. Le. EfficientDet: Scalable and efficient object detection. In *CVPR*, 2020.

[58] Mingxing Tan, Ruoming Pang, and Quoc V. Le. EfficientDet: Scalable and efficient object detection. *arXiv:1911.09070v7*, 2020.

[59] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *ICCV*, 2019.

[60] Xinjiang Wang, Shilong Zhang, Zhuoran Yu, Litong Feng, and Wayne Zhang. Scale-equalizing pyramid convolution for object detection. In *CVPR*, 2020.

[61] Xudong Wang, Zhaowei Cai, Dashan Gao, and Nuno Vasconcelos. Towards universal object detection by domain attention. In *CVPR*, 2019.

[62] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. WIDER FACE: A face detection benchmark. In *CVPR*, 2016.

[63] Xuehui Yu, Yuqi Gong, Nan Jiang, Qixiang Ye, and Zhenjun Han. Scale match for tiny person detection. In *WACV*, 2020.

[64] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *CVPR*, 2020.

[65] 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 *ECCV*, 2020.

[66] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Simple multi-dataset detection. In *CVPR*, 2022.

[67] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *ICLR*, 2021.