USB: Universal-Scale Object Detection Benchmark

Yosuke Shinya
https://shinya7y.github.io/

independent researcher
Tokyo, Japan

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
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 [35]. 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 $1333 \times 800$. These settings are not obligations but non-binding agreements for fair comparison. Some studies do not follow the settings for accurate and fast detectors1. 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.

---

1YOLOv4 was trained for 273 epochs [4], DETR for 500 epochs [8], 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 $512 \times 512$ to $1536 \times 1536$ [57].
2 Related Work

Multi-scale object detection. Detecting multi-scale objects is a fundamental challenge in object detection [1, 37]. Various components have been improved, including backbones and modules [1, 11, 22, 24, 56], necks [12, 34, 57, 60], heads and training sample selection [11, 12, 24], and multi-scale training and testing [13, 37, 60] (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 [13, 24]. 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 [65]. 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 [61] 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× epochs). We resolved this problem by establishing protocols for fair training and evaluation.
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.

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.

---

[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.
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
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 [4] 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 [4, 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, \ldots\}, \{0.1, 0.2, 0.5, 1.0, \ldots\}, and \{0.1, 0.3, 1.0, \ldots\}). 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, \ldots, 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). 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 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. For Mini USB, we limit the resolution based on 512×512. This resolution is popular in the PASCAL VOC benchmark, which contains small images and large objects. It is also popular in real-time detectors. We adopted a further small-scale 224×224 for Micro USB. This resolution is popular in ImageNet classification. 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.

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 to evaluate the performance of detectors on each dataset. We provide data format converters for WOD and M109s. The COCO-
style AP (CAP) for a dataset \( d \) is calculated as 
\[
\text{CAP}_d = \frac{1}{|T|} \sum_{t \in T} \frac{1}{|C_d|} \sum_{c \in C_d} \text{AP}_{t,c},
\]
where \( T = \{0.5, 0.55, ..., 0.95\} \) denotes the predefined 10 IoU thresholds, \( C_d \) denotes categories in the dataset \( d \), and \( \text{AP}_{t,c} \) 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. \( \text{AP}_{50} \) and \( \text{AP}_{75} \) denote AP at single IoU thresholds of 0.5 and 0.75, respectively. \( \text{AP}_S \), \( \text{AP}_M \), and \( \text{AP}_L \) are variants of CAP, where target objects are limited to small (area \( \leq 32^2 \)), medium (\( 32^2 \leq \text{area} \leq 96^2 \)), and large (\( 96^2 \leq \text{area} \)) 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 
\[
m\text{CAP} = \frac{1}{|D|} \sum_{d \in D} \text{CAP}_d.
\]
Since USB adopts the three datasets described in Sec. 3.3, 
\[
m\text{CAP} = (\text{CAP}_{\text{COCO}} + \text{CAP}_{\text{WOD}} + \text{CAP}_{\text{M109s}}) / 3.
\]
Similarly, we define five metrics from \( \text{AP}_{50} \), \( \text{AP}_{75} \), \( \text{AP}_S \), \( \text{AP}_M \), and \( \text{AP}_L \) by averaging them over the datasets.

The three COCO-style scale-wise metrics (\( \text{AP}_S \), \( \text{AP}_M \), and \( \text{AP}_L \)) 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 test 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 [22] 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]. 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 [44], Deformable Convolutional Networks (DCN) [44], and multi-scale training. (15) UniverseNet-20.08: A variant of UniverseNet designed around August 2020 with GFL [44], SyncBN [44], iBN [44], and the light use of DCN [44, 44]. See Supp. D for the details of the methods and architectures used in UniverseNets.

<table>
<thead>
<tr>
<th>Hyperparameters</th>
<th>COCO</th>
<th>WOD</th>
<th>M109s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning rate for multi-stage detectors</td>
<td>0.02</td>
<td>0.02</td>
<td>0.16</td>
</tr>
<tr>
<td>Learning rate for single-stage detectors</td>
<td>0.01</td>
<td>0.01</td>
<td>0.08</td>
</tr>
<tr>
<td>Test scale</td>
<td>1333×800</td>
<td>1248×832</td>
<td>1216×864</td>
</tr>
</tbody>
</table>

Table 5: Default hyperparameters. See Supp. E for exceptions.
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 [1] 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.

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.
**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 [25].

**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 [4] 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


[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.
