LVIS: A Dataset for Large Vocabulary Instance Segmentation

Agrim Gupta Piotr Dollár Ross Girshick
Facebook AI Research (FAIR)

Abstract

Progress on object detection is enabled by datasets that focus the research community's attention on open challenges. This process led us from simple images to complex scenes and from bounding boxes to segmentation masks. In this work, we introduce LVIS *(pronounced 'el-vis'): a new dataset for Large Vocabulary Instance Segmentation. We plan to collect 2.2 million high-quality instance segmentation masks for over 1000 entry-level object categories in 164k images. Due to the Zipfian distribution of categories in natural images,* LVIS *naturally has a long tail of categories with few training samples. Given that state-of-the-art deep learning methods for object detection perform poorly in the low-sample regime, we believe that our dataset poses an important and exciting new scientific challenge.* LVIS *is available at* `http://www.lvisdataset.org`.

1. Introduction

A central goal of computer vision is to endow algorithms with the ability to intelligently describe images. Object detection is a canonical image description task; it is intuitively appealing, useful in applications, and straightforward to benchmark in existing settings. The accuracy of object detectors has improved dramatically and new capabilities, such as predicting segmentation masks and 3D representations, have been developed. There are now exciting opportunities to push these methods towards new goals.

Today, rigorous evaluation of general purpose object detectors is mostly performed in the few category regime (*e.g.* 80) or when there are a large number of training examples per category (*e.g.* 100 to 1000+). There is now an opportunity to enable research in the setting where there are a large number of categories *and* where per-category data is sometimes scarce. *The long tail of rare categories is inescapable; annotating more images simply uncovers previously unseen, rare categories (see Fig. 9 and [29, 25, 24, 27])*. Efficiently learning from few examples is a significant open problem in machine learning and computer vision, making this opportunity one of the most exciting from a scientific and practical perspective. But to open this area to empirical study, a suitable, high-quality dataset and benchmark are required.

Figure 1. **Example annotations.** We present **LVIS**, a new dataset for benchmarking Large Vocabulary Instance Segmentation in the 1000+ category regime with a challenging long tail of rare objects.

We aim to enable this kind of research by designing and collecting **LVIS** (pronounced 'el-vis')—a new benchmark dataset for research on Large Vocabulary Instance Segmentation. We are collecting instance segmentation masks for more than 1000 entry-level object categories (see Fig. 1). When completed, we plan for our dataset to contain 164k images and 2.2 million *high-quality* instance masks.[1] Our annotation pipeline starts from a set of images that were collected without prior knowledge of the categories that will be labeled in them. We engage annotators in an iterative object spotting process that uncovers the long tail of categories that naturally appears in the images and avoids using machine learning algorithms to automate data labeling.

We designed a crowdsourced annotation pipeline that enables the collection of our large-scale dataset while also yielding high-quality segmentation masks. Quality is important for future research because relatively coarse masks, such as those in the COCO dataset [18], limit the ability to differentiate algorithm-predicted mask quality beyond a certain, coarse point. When compared to expert annotators, our segmentation masks have higher overlap and boundary

[1] We plan to annotate the 164k images in COCO 2017 (we have permission to label `test2017`). 2.2M is a projection from current data.

consistency than both COCO and ADE20K [28].

To build our dataset, we adopt an *evaluation-first design principle*. This principle states that we should first determine exactly how to perform quantitative evaluation and only then design and build a dataset collection pipeline to gather the data entailed by the evaluation. We select our benchmark task to be COCO-style instance segmentation and we use the same COCO-style average precision (AP) metric that averages over categories and different mask intersection over union (IoU) thresholds [19]. Task and metric continuity with COCO reduces barriers to entry.

Buried within this seemingly innocuous task choice are immediate technical challenges: How do we fairly evaluate detectors when one object can reasonably be labeled with multiple categories (see Fig. 2)? How do we make the annotation workload feasible when labeling 164k images with segmented objects from over 1000 categories?

The essential design choice resolving these challenges is to build a *federated dataset*: a single dataset that is formed by the union of a large number of smaller constituent datasets, each of which looks exactly like a traditional object detection dataset for a single category. Each small dataset provides the essential guarantee of *exhaustive annotations* for a single category—*all instances of that category are annotated*. Multiple constituent datasets may overlap and thus a single object within an image can be labeled with multiple categories. Furthermore, since the exhaustive annotation guarantee only holds within each small dataset, we do not require the entire federated dataset to be exhaustively annotated with all categories, which dramatically reduces the annotation workload. Crucially, at test time the membership of each image with respect to the constituent datasets is not known by the algorithm and thus it must make predictions as if all categories will be evaluated. The evaluation oracle evaluates each category fairly on its constituent dataset.

In the remainder of this paper, we summarize how our dataset and benchmark relate to prior work, provide details on the evaluation protocol, describe how we collected data, and then discuss results of the analysis of this data.

Dataset Timeline. We report detailed analysis on a 5000 image subset that we have annotated twice. We are working with challenge organizers from the COCO dataset committee and hope to run the first LVIS challenge at the 2019 COCO workshop, likely at ICCV. We anticipate that LVIS annotation collection will be completed by this time.

1.1. Related Datasets

Datasets shape the technical problems researchers study and consequently the path of scientific discovery [17]. We owe much of our current success in image recognition to pioneering datasets such as MNIST [16], BSDS [20], Caltech 101 [6], PASCAL VOC [5], ImageNet [23], and

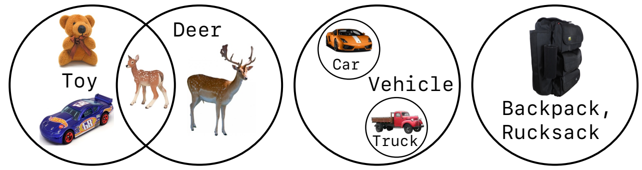


Figure 2. **Category relationships from left to right:** non-disjoint category pairs may be in *partially overlapping*, *parent-child*, or *equivalent (synonym)* relationships. Fair evaluation of object detectors must take into account these relationships and the fact that a single object may have multiple valid category labels.

COCO [18]. These datasets enabled the development of algorithms that detect edges, perform large-scale image classification, and localize objects by bounding boxes and segmentation masks. They were also used in the discovery of important ideas, such as Convolutional Networks [15, 13], Residual Networks [10], and Batch Normalization [11].

LVIS is inspired by these and other related datasets, including those focused on street scenes (Cityscapes [3] and Mapillary [22]) and pedestrians (Caltech Pedestrians [4]). We review the most closely related datasets below.

COCO [18] is the most popular instance segmentation benchmark for common objects. It contains 80 categories that are pairwise distinct. There are a total of 118k training images, 5k validation images, and 41k test images. All 80 categories are exhaustively annotated in all images (ignoring annotation errors), leading to approximately 1.2 million instance segmentation masks. To establish continuity with COCO, we adopt the same instance segmentation task and AP metric, and we are also annotating all images from the COCO 2017 dataset. All 80 COCO categories can be mapped into our dataset. In addition to representing an order of magnitude more categories than COCO, our annotation pipeline leads to higher-quality segmentation masks that more closely follow object boundaries (see §4).

ADE20K [28] is an ambitious effort to annotate almost every pixel in 25k images with object instance, 'stuff', and part segmentations. The dataset includes approximately 3000 named objects, stuff regions, and parts. Notably, ADE20K was annotated by a *single expert annotator*, which increases consistency but also limits dataset size. Due to the relatively small number of annotated images, most of the categories do not have enough data to allow for both training and evaluation. Consequently, the instance segmentation benchmark associated with ADE20K evaluates algorithms on the 100 most frequent categories. In contrast, our goal is to enable benchmarking of *large vocabulary* instance segmentation methods.

iNaturalist [26] contains nearly 900k images annotated with bounding boxes for an astonishing 5000 plant and animal species. Similar to our goals, iNaturalist emphasizes

Figure 3. **Example annotations** from our dataset. For clarity, we show one category per image.

the importance of benchmarking classification and detection in the few example regime. Unlike our effort, iNaturalist does not include segmentation masks and is focussed on a different image and *fine-grained* category distribution; our category distribution emphasizes entry-level categories.

Open Images v4 [14] is a large dataset of 1.9M images. The detection portion of the dataset includes 15M bounding boxes labeled with 600 object categories. The associated benchmark evaluates the 500 most frequent categories, all of which have over 100 training samples (>70% of them have over 1000 training samples). Thus, unlike our benchmark, low-shot learning is not integral to Open Images. Also different from our dataset is a reliance on machine learning algorithms to select which images will be annotated by using classifiers for the target categories. Our data collection process, in contrast, involves no machine learning algorithms (see §4.1 and Fig. 5). With release v4, developed concurrently with our work, Open Images has used a federated dataset design for their object detection task.

2. Dataset Design

We followed an *evaluation-first design principle*: prior to any data collection, we precisely defined what task would be performed and how it would be evaluated. This principle is important because there are technical challenges that arise when evaluating detectors on a large vocabulary dataset that do not occur when there are few categories. These must be resolved first, because they have profound implications for the structure of the dataset, as we discuss next.

2.1. Task and Evaluation Overview

Task and Metric. Our dataset benchmark is the instance segmentation task: given a fixed, known set of categories, design an algorithm that when presented with a previously unseen image will output a segmentation mask for each instance of each category that appears in the image along with the category label and a confidence score. Given the output of an algorithm over a set of images, we compute mask average precision (AP) using the definition and implementation from the COCO dataset [19] (for more detail see §2.3).

Evaluation Challenges. Datasets like PASCAL VOC and COCO use manually selected categories that are *pairwise disjoint*: when annotating a *car*, there's never any question if the object is instead a *potted plant* or a *sofa*. When increasing the number of categories, it is inevitable that other types of pairwise relationships will occur: (1) partially overlapping visual concepts; (2) parent-child relationships; and (3) perfect synonyms. See Fig. 2 for examples.

If these relations are not properly addressed, then the evaluation protocol will be unfair. For example, most *toys* are not *deer* and most *deer* are not *toys*, but a *toy deer* is both—if a detector outputs *deer* and the object is only labeled *toy*, the detection will be marked as wrong. Likewise, if a car is only labeled *vehicle*, and the algorithm outputs *car*, it will be incorrectly judged to be wrong. Or, if an object is only labeled *backpack* and the algorithm outputs the synonym *rucksack*, it will be incorrectly penalized. Providing a fair benchmark is important for accurately reflecting algorithm performance.

These problems occur when the ground-truth annotations are missing one or more true labels for an object. If an algorithm happens to predict one of these correct, *but missing* labels, it will be unfairly penalized. Now, if all objects are exhaustively and correctly labeled with all categories, then the problem is trivially solved. But correctly and exhaustively labeling 164k images each with 1000 categories is undesirable: *it forces a binary judgement deciding if each category should be applied to each object; there will be many cases of genuine ambiguity, inter-annotator disagreement, and the annotation workload will be very large*. Given these drawbacks, we describe our solution next.

2.2. Federated Datasets

Our key observation is that the desired evaluation protocol does not require us to exhaustively annotate all images with all categories. What is required instead is that for each category c there must exist disjoint subsets of the entire dataset $\mathcal{D}$ for which the following guarantees hold:

Positive set: there exists a subset of images $\mathcal{P}_c \subseteq \mathcal{D}$ such that all instances of c in these images are segmented. In other words, $\mathcal{P}_c$ is exhaustively annotated for category c.

Negative set: there exists a subset of images $\mathcal{N}_c \subseteq \mathcal{D}$ such that no instance of c appears in any of these images.

Given these two subsets for a category c, $\mathcal{P}_c \cup \mathcal{N}_c$ can be used to perform standard COCO-style AP evaluation for c. We only judge the algorithm on a category c in the subset of images in which c has been exhaustively annotated; if a detector reports a detection of category c on an image $i \notin \mathcal{P}_c \cup \mathcal{N}_c$, the detection is *not* evaluated.

By collecting the per-category sets into a single dataset, $\mathcal{D} = \cup_c(\mathcal{P}_c \cup \mathcal{N}_c)$, we arrive at the concept of a *federated dataset*. A federated dataset is a dataset that is formed by the union of smaller constituent datasets, each of which looks exactly like a traditional object detection dataset for a single category. *By not annotating all images with all categories, freedom is created to design an annotation process that avoids ambiguous cases and collects annotations only if there is sufficient inter-annotator agreement. At the same time, the workload can be dramatically reduced.*

Finally, we note that positive set and negative set membership on the test split is not disclosed and therefore algorithms have no side information about what categories will be evaluated in each image. An algorithm thus must make its best prediction for *all* categories in each test image.

Reduced Workload. Federated dataset design allows us to make $|\mathcal{P}_c \cup \mathcal{N}_c| \ll |\mathcal{D}|, \forall c$. This choice dramatically reduces the workload and allows us to undersample the most frequent categories in order to avoid wasting annotation resources on them (*e.g. person* accounts for 30% of COCO). Of our estimated 2.2 million instances, likely no single category will account for more than ~3% of the total instances.

2.3. Evaluation Details

The evaluation API only returns the overall category-averaged AP, not per-category APs. We do this because: (1) it avoids leaking which categories are present in the test set;[2] (2) given that tail categories are rare, there will be few examples for evaluation in some cases, which makes per-category AP unstable; (3) by averaging over a large number of categories, the overall category-averaged AP has lower variance, making it a robust metric for ranking algorithms.

Non-Exhaustive Annotations. We also collect an image-level boolean label, e_i^c, indicating if image $i \in \mathcal{P}_c$ is exhaustively annotated for category c. In most cases (91%), this flag is true, indicating that the annotations are indeed exhaustive. In the remaining cases, there is at least one instance in the image that is not annotated. Missing annotations often occur in 'crowd' cases in which there are a large number of instances and delineating them is difficult. During evaluation, we do not count false positives for category c on images i that have e_i^c set to false. We do measure recall on these images: the detector is expected to predict accurate segmentation masks for the labeled instances. Our strategy differs from other datasets that use a small maximum number of instances per image, per category (10-15) together with 'crowd regions' (COCO) or use a special 'group of c' label to represent 5 or more instances (Open Images). Our annotation pipeline (§3) attempts to collect segmentations for *all* instances in an image, regardless of count, and then checks if the labeling is in fact exhaustive. See Fig. 3.

Hierarchy. During evaluation, we treat all categories the same; we do nothing special in the case of hierarchical relationships. To perform best, for each detected object o, the detector should output the most specific correct category as well as all more general categories, *e.g.*, a canoe should be labeled both *canoe* and *boat*. The detected object o in image i will be evaluated with respect to all labeled positive categories $\{c \mid i \in \mathcal{P}_c\}$, which may be any subset of categories between the most specific and the most general.

Synonyms. A federated dataset that separates synonyms into different categories is valid, but is unnecessarily fragmented (see Fig. 2, right). We avoid splitting synonyms into separate categories with WordNet [21]. Specifically, in LVIS each category c is a WordNet *synset*—a word sense specified by a set of synonyms and a definition.

[2] It's possible that the categories present in the validation and test sets may be a strict subset of those in the training set; we use the standard COCO 2017 validation and test splits and cannot guarantee that all categories present in the training data are also present in validation and test.

Stage 1: Object spotting, one instance per category | Stage 2: Exhaustive instance marking of each category | Stage 3 & 4 (back and forth): Segmentation and verification | Stage 5: Exhaustive annotation verification | Stage 6: Negative labels

Figure 4. Our **annotation pipeline** comprises six stages. **Stage 1: Object Spotting** elicits annotators to mark a single instance of many different categories per image. This stage is iterative and causes annotators to discover a long tail of categories. **Stage 2: Exhaustive Instance Marking** extends the stage 1 annotations to cover all instances of each spotted category. Here we show additional instances of *book*. **Stages 3 and 4: Instance Segmentation and Verification** are repeated back and forth until ~99% of all segmentations pass a quality check. **Stage 5: Exhaustive Annotations Verification** checks that all instances are in fact segmented and flags categories that are missing one or more instances. **Stage 6: Negative Labels** are assigned by verifying that a subset of categories do not appear in the image.

3. Dataset Construction

In this section we provide an overview of our annotation pipeline. User interface examples are in the supplement.[3]

3.1. Annotation Pipeline

Fig. 4 illustrates our annotation pipeline by showing the output of each stage, which we describe below. For now, assume that we have a fixed category vocabulary $\mathcal{V}$. We will describe how the vocabulary was collected in §3.2.

Object Spotting, Stage 1. The goals of the object spotting stage are to: (1) generate the positive set, $\mathcal{P}_c$, for each category $c \in \mathcal{V}$ and (2) elicit vocabulary recall such that many different object categories are included in the dataset.

Object spotting is an iterative process in which each image is visited a variable number of times. On the first visit, an annotator is asked to mark one object with a point and to name it with a category $c \in \mathcal{V}$ using an *autocomplete* text input. On each subsequent visit, all previously spotted objects are displayed and an annotator is asked to mark an object of a previously unmarked category or to skip the image if no more categories in $\mathcal{V}$ can be spotted. When an image has been skipped 3 times, it will no longer be visited. The autocomplete is performed against the set of all synonyms, presented with their definitions; we internally map the selected word to its synset/category to resolve synonyms.

Obvious and salient objects are spotted early in this iterative process. As an image is visited more, less obvious objects are spotted, including incidental, non-salient ones. We run the spotting stage twice, and for each image we retain categories that were spotted in both runs. *Thus two people must independently agree on a name in order for it to be included in the dataset; this increases naming consistency.*

To summarize the output of stage 1: for each category in the vocabulary, we have a (possibly empty) set of images in which one object of that category is marked per image. This defines an initial positive set, $\mathcal{P}_c$, for each category c.

[3] See an extended version of this work on arXiv (under preparation).

Exhaustive Instance Marking, Stage 2. The goals this stage are to: (1) verify stage 1 annotations and (2) take each image $i \in \mathcal{P}_c$ and mark *all* instances of c in i with a point.

In this stage, (i, c) pairs from stage 1 are each sent to 5 annotators. They are asked to perform two steps. First, they are shown the definition of category c and asked to verify if it describes the spotted object. Second, if it matches, then the annotators are asked to mark all other instances of the same category. If it does not match, there is no second step. To prevent frequent categories from dominating the dataset and to reduce the overall workload, we subsample frequent categories such that no positive set exceeds more than 1% of the images in the dataset.

To ensure annotation quality, we embed a 'gold set' within the pool of work. These are cases for which we know the correct ground-truth. We use the gold set to automatically evaluate the work quality of each annotator so that we can direct work towards more reliable annotators. We use 5 annotators per (i, c) pair to help ensure instance-level recall.

To summarize, from stage 2 we have exhaustive instance spotting for each image $i \in \mathcal{P}_c$ for each category $c \in \mathcal{V}$.

Instance Segmentation, Stage 3. The goals of the instance segmentation stage are to: (1) verify the category for each marked object from stage 2 and (2) upgrade each marked object from a point annotation to a full segmentation mask.

To do this, each pair (i, o) of image i and marked object instance o is presented to one annotator who is asked to verify that the category label for o is correct and if it is correct, to draw a *detailed* segmentation mask for it (*e.g.* see Fig. 3).

We use a training task to establish our quality standards. Annotator quality is assessed with a gold set and by tracking their average vertex count per polygon. We use these metrics to assign work to reliable annotators.

In sum, from stage 3 we have for each image and spotted instance pair one segmentation mask (if it is not rejected).

Segment Verification, Stage 4. The goal of the segment verification stage is to verify the quality of the segmentation masks from stage 3. We show each segmentation to up to 5 annotators and ask them to rate its quality using a rubric. If two or more annotators reject the mask, then we requeue the instance for stage 3 segmentation. Thus we only accept a segmentation if 4 annotators agree it is high-quality. Unreliable workers from stage 3 are not invited to judge segmentations in stage 4; we also use rejections rates from this stage to monitor annotator reliability. We iterate between stages 3 & 4 a total of four times, each time only re-annotating rejected instances.

To summarize the output of stage 4 (after iterating back and forth with stage 3): we have a high-quality segmentation mask for >99% of all marked objects.

Full Recall Verification, Stage 5. The full recall verification stage finalizes the positive sets. The goal is to find images $i \in \mathcal{P}_c$ where c is not exhaustively annotated. We do this by asking annotators if there are any unsegmented instances of category c in i. We ask up to 5 annotators and require at least 4 to agree that annotation is exhaustive. As soon as two believe it is not, we mark the exhaustive annotation flag e_i^c as false. We use a gold set to maintain quality.

To summarize the output of stage 5: we have a boolean flag e_i^c for each image $i \in \mathcal{P}_c$ indicating if category c is exhaustively annotated in image i. This finalizes the positive sets along with their instance segmentation annotations.

Negative Sets, Stage 6. The final stage of the pipeline is to collect a negative set $\mathcal{N}_c$ for each category c in the vocabulary. We do this by randomly sampling images $i \in \mathcal{D} \setminus \mathcal{P}_c$, where $\mathcal{D}$ is all images in the dataset. For each sampled image i, we ask up to 5 annotators if category c appears in image i. If any one annotator reports that it does, we reject the image. Otherwise i is added to $\mathcal{N}_c$. We sample until the negative set $\mathcal{N}_c$ reaches a target size of 1% of the images in the dataset. We use a gold set to maintain quality.

To summarize, from stage 6 we have a negative image set $\mathcal{N}_c$ for each category $c \in \mathcal{V}$ such that the category does not appear in any of the images in $\mathcal{N}_c$.

3.2. Vocabulary Construction

We construct the vocabulary $\mathcal{V}$ with an iterative process that starts from a large super-vocabulary and uses the object spotting process (stage 1) to winnow it down. We start from 8.8k synsets that were selected from WordNet by removing some obvious cases (*e.g.* proper nouns) and then finding the intersection with highly concrete common nouns [2]. This yields a high-recall set of concrete, and thus likely visual, entry-level synsets. We then apply object spotting to 10k COCO images with autocomplete against this super-vocabulary. This yields a reduced vocabulary with which we repeat the process once more. Finally, we perform minor manual editing. For more details, see the supplement.[3] The resulting vocabulary contains 1723 synsets—the upper bound on the number of categories that can appear in LVIS.

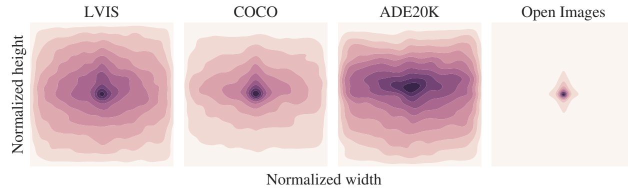


Figure 5. Distribution of object centers in normalized image coordinates for four datasets. Objects in LVIS, COCO, and ADE20K are well distributed (objects in LVIS are slightly less centered than in COCO and slightly more centered than in ADE20K). On the other hand, Open Images exhibits a strong center bias.

4. Dataset Analysis

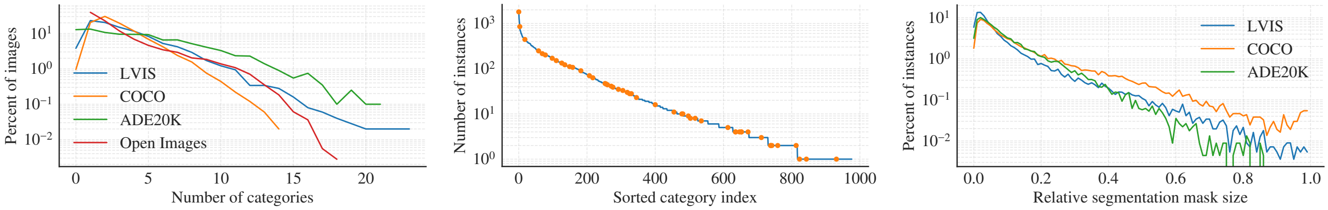
For analysis, we have annotated 5000 images (the COCO `val2017` split) twice using the proposed pipeline. We begin by discussing general dataset statistics next before proceeding to an analysis of annotation consistency in §4.2 and an analysis of the evaluation protocol in §4.3.

4.1. Dataset Statistics

Category Statistics. There are 977 categories present in the 5000 LVIS images. The category growth rate (see Fig. 9) indicates that the final dataset will have well over 1000 categories. On average, each image is annotated with 11.2 instances from 3.4 categories. The largest instances-per-image count is a remarkable 294. Fig. 6a shows the full categories-per-image distribution. LVIS's distribution has more spread than COCO's indicating that many images are labeled with more categories. The low-shot nature of our dataset can be seen in Fig. 6b, which plots the total number of instances for each category (in the 5000 images). The median value is 9, and while this number will be larger for the full image set, this statistic highlights the challenging long-tailed nature of our data.

Spatial Statistics. Our object spotting process (stage 1) encourages the inclusion of objects distributed throughout the image plane, not just the most salient foreground objects. The effect can be seen in Fig. 5 which shows object-center density plots. While objects in LVIS, COCO, and ADE20K are fairly well distributed, objects in Open Images exhibit a strong centered object bias possibly due to semi-automated annotation. The even distribution of object centers is an important characteristic for detection datasets and was a core motivating factor for the creation of COCO which emphasized *in context* detection. LVIS shares this property.

Scale Statistics. Objects in LVIS are also more likely to be small. Fig. 6c shows the relative size distribution of object masks: compared with COCO, LVIS objects tend to smaller and there are fewer large objects (*e.g.*, objects that occupy most of an image are ~10× less frequent). ADE20K has the fewest large objects overall and more medium ones.

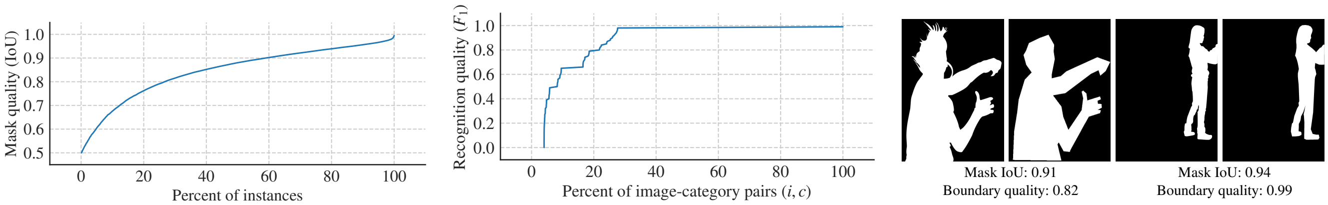


(a) Distribution of categories in images. LVIS has a heavier tail than COCO and Open Images. ADE20K has the most uniform distribution.

(b) The number of instances per category (on 5k images) reveals the long tail with few examples. Orange dots: categories in common with COCO.

(c) Relative segmentation mask size (square root of mask-area-divided-by-image-area) compared between LVIS, COCO, and ADE20K.

Figure 6. **Dataset statistics**. Best viewed digitally.



(a) LVIS segmentation quality measured by mask IoU between matched instances from two runs of our annotation pipeline. Masks from the runs are consistent with a dataset average IoU of 0.85.

(b) LVIS recognition quality measured by F_1 score given matched instances across two runs of our annotation pipeline. Category labeling is consistent with a dataset average F_1 score of 0.87.

(c) Illustration of mask IoU *vs*. boundary quality to provide intuition for interpreting Fig. 7a (left) and Tab. 1a (dataset annotations *vs*. expert annotators, below).

Figure 7. **Annotation consistency** using 5000 *doubly annotated* images from LVIS. Best viewed digitally.

dataset	comparison	mask IoU mean	mask IoU median	boundary quality mean	boundary quality median
COCO	dataset *vs*. experts	0.83 – 0.87	0.88 – 0.91	0.77 – 0.82	0.79 – 0.88
	expert 1 *vs*. expert 2	0.91 – 0.95	0.96 – 0.98	0.92 – 0.96	0.97 – 0.99
ADE20K	dataset *vs*. experts	0.84 – 0.88	0.90 – 0.93	0.83 – 0.87	0.84 – 0.92
	expert 1 *vs*. expert 2	0.90 – 0.94	0.95 – 0.97	0.90 – 0.95	0.99 – 1.00
LVIS	dataset *vs*. experts	**0.90 – 0.92**	**0.94 – 0.96**	**0.87 – 0.91**	**0.93 – 0.98**
	expert 1 *vs*. expert 2	0.93 – 0.96	0.96 – 0.98	0.91 – 0.96	0.97 – 1.00

(a) For each metric (mask IoU, boundary quality) and each statistic (mean, median), we show a bootstrapped 95% confidence interval. LVIS has the highest quality across all measures.

dataset	annotation source	boundary complexity mean	boundary complexity median
COCO	dataset	5.59 – 6.04	5.13 – 5.51
	experts	6.94 – 7.84	5.86 – 6.80
ADE20K	dataset	6.00 – 6.84	4.79 – 5.31
	experts	6.34 – 7.43	4.83 – 5.53
LVIS	dataset	**6.35 – 7.07**	**5.44 – 6.00**
	experts	7.13 – 8.48	5.91 – 6.82

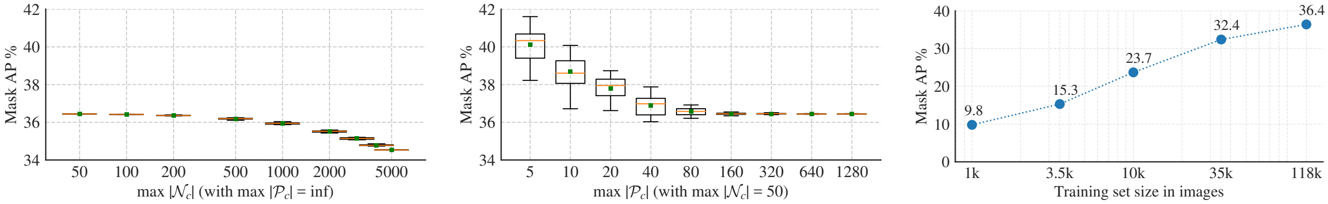
(b) Comparison of annotation complexity. Boundary complexity is perimeter divided by square root area [1].

Table 1. Annotation **quality and complexity** relative to experts.

4.2. Annotation Consistency

Annotation Pipeline Repeatability. A repeatable annotation pipeline implies that the process generating the ground-truth data is not overly random and therefore may be learned. To understand repeatability, we annotated the 5000 images twice: after completing object spotting (stage 1), we have initial positive sets $\mathcal{P}_c$ for each category c; we then execute stages 2 through 5 (exhaustive instance marking through full recall verification) twice in order to yield doubly annotated positive sets. To compare them, we compute a matching between them for each image and category pair. We find a matching that maximizes the total mask intersection over union (IoU) summed over the matched pairs and then discard any matches with IoU < 0.5. Given these matches we compute the dataset average mask IoU (0.85) and the dataset average F_1 score (0.87). Intuitively, these quantities describe 'segmentation quality' and 'recognition quality' [12]. The cumulative distributions of these metrics (Fig. 7a and 7b) show that even though matches are established based on a low IoU threshold (0.5), matched masks tend to have much higher IoU. The results show that roughly 50% of matched instances have IoU greater than 90% and roughly 75% of the image-category pairs have a perfect F_1 score. Taken together, these metrics are a strong indication that our pipeline has a large degree of repeatability.

Comparison with Expert Annotators. To measure segmentation quality, we randomly selected 100 instances with mask area greater than 32^2 pixels from LVIS, COCO, and ADE20K. We presented these instances (indicated by bounding box and category) to two independent expert annotators and asked them to segment each object using professional image editing tools. We compare dataset annotations to expert annotations using mask IoU and boundary quality (boundary F [20]) in Tab. 1a. The results (bootstrapped 95% confidence intervals) show that our masks are high-quality, surpassing COCO and ADE20K on both measures (see Fig. 7c for intuition). At the same time, the objects in LVIS have more complex boundaries [1] (Tab. 1b).



(a) Given fixed detections, we show how AP varies with $\max |\mathcal{N}_c|$, the max number of negative images per category used in evaluation.

(b) With the same detections from Fig. 8a and $\max |\mathcal{N}_c| = 50$, we show how AP varies as we vary $\max |\mathcal{P}_c|$, the max positive set size.

(c) Low-shot detection is an open problem: training Mask R-CNN on 1k images decreases COCO `val2017` mask AP from 36% to 10%.

Figure 8. **Detection experiments** using COCO and 5000 annotated images from LVIS. Best viewed digitally.

Mask R-CNN	test anno.	box AP	mask AP
R-50-FPN	COCO	38.2	34.1
model id: 35859007	LVIS	38.8	34.4
R-101-FPN	COCO	40.6	36.0
model id: 35861858	LVIS	40.9	36.0
X-101-64x4d-FPN	COCO	47.8	41.2
model id: 37129812	LVIS	48.6	41.7

Table 2. COCO-trained Mask R-CNN evaluated on LVIS annotations. Both annotations yield similar AP values.

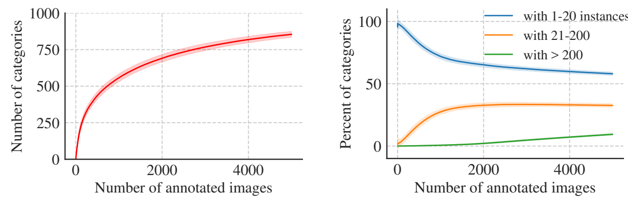


Figure 9. (Left) As more images are annotated, new categories are discovered. (Right) Consequently, the percentage of low-shot categories (blue curve) remains large, decreasing slowly.

4.3. Evaluation Protocol

COCO Detectors on LVIS. To validate our annotations and federated dataset design we downloaded three Mask R-CNN [9] models from the Detectron Model Zoo [7] and evaluated them on LVIS annotations for the categories in COCO. Tab. 2 shows that both box AP and mask AP are close between our annotations and the original ones from COCO for all models, which span a wide AP range. This result validates our annotations and evaluation protocol: even though LVIS uses a federated dataset design with sparse annotations, the quantitative outcome closely reproduces the 'gold standard' results from dense COCO annotations.

Federated Dataset Simulations. For insight into how AP changes with positive and negative sets sizes $|\mathcal{P}_c|$ and $|\mathcal{N}_c|$, we randomly sample smaller evaluation sets from COCO `val2017` and recompute AP. To plot quartiles and min-max ranges, we re-test each setting 20 times. In Fig. 8a we use all positive instances for evaluation, but vary $\max |\mathcal{N}_c|$ between 50 and 5k. AP decreases somewhat (~2 points) as we increase the number of negative images as the ratio of negative to positive examples grows with fixed $|\mathcal{P}_c|$ and increasing $|\mathcal{N}_c|$. Next, in Fig 8b we set $\max |\mathcal{N}_c| = 50$ and vary $|\mathcal{P}_c|$. We observe that even with a small positive set size of 80, AP is similar to the baseline with low variance. With smaller positive sets (down to 5) variance increases, but the AP gap from 1st to 3rd quartile remains below 2 points. These simulations together with COCO detectors tested on LVIS (Tab. 2) indicate that including smaller evaluation sets for each category is viable for evaluation.

Low-Shot Detection. To validate the claim that low-shot detection is a challenging open problem, we trained Mask R-CNN on random subsets of COCO `train2017` ranging from 1k to 118k images. For each subset, we optimized the learning rate schedule and weight decay by grid search. Results on `val2017` are shown in Fig. 8c. At 1k images, mask AP drops from 36.4% (full dataset) to 9.8% (1k subset). In the 1k subset, 89% of the categories have more than 20 training instances, while the low-shot literature typically considers $\ll 20$ examples per category [8]. We estimate that roughly 50% of the categories in LVIS will have < 20 training instances, see Fig. 9 (right), discussed next.

Low-Shot Category Statistics. Fig. 9 (left) shows the category growth curve as a function of image count in the dataset (up to 977 categories in 5k images). Extrapolating the trajectory, our final dataset should include well over 1k categories (upper bounded by the vocabulary size, 1723). *Note that the low-shot nature of* LVIS *is largely independent of the scale of the dataset, Fig. 9 (right).* That is, even as the number of annotated images increases, new categories will be added that have few labeled examples.

5. Conclusion

We introduced LVIS, a new dataset designed to enable, for the first time, the rigorous study of instance segmentation algorithms that can recognize a large vocabulary of object categories (>1000) and must do so using methods that can cope with the open problem of low-shot learning. While LVIS emphasizes learning from few examples, the dataset is not small: it will span 164k images and label ~2.2 million object instances. Each object instance is segmented with a high-quality mask that surpasses the annotation quality of related datasets. We plan to establish LVIS as a benchmark challenge that we hope will lead to exciting new object detection, segmentation, and low-shot learning algorithms.

References

[1] Fred Attneave and Malcolm D Arnoult. The quantitative study of shape and pattern perception. *Psychological bulletin*, 1956. 7

[2] Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 2014. 6

[3] 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 *CVPR*, 2016. 2

[4] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *TPAMI*, 2012. 2

[5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010. 2

[6] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *TPAMI*, 2006. 2

[7] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. https://github.com/facebookresearch/detectron, 2018. 8

[8] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, 2017. 8

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 8

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 2

[12] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 7

[13] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[14] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018. 3

[15] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1989. 2

[16] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/, 1998. 2

[17] Marc Liberman. Reproducible research and the common task method. Simmons Foundation Lecture https://www.simonsfoundation.org/lecture/reproducible-research-and-the-common-task-method/, 2015. 2

[18] 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. 1, 2

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. COCO detection evaluation. http://cocodataset.org/#detection-eval, Accessed Oct 30, 2018. 2, 3

[20] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001. 2, 7

[21] George Miller. *WordNet: An electronic lexical database*. MIT press, 1998. 4

[22] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 2

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 2

[24] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *IJCV*, 2008. 1

[25] Merrielle Spain and Pietro Perona. Measuring and predicting importance of objects in our visual world. Technical Report CNS-TR-2007-002, California Institute of Technology, 2007. 1

[26] 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 *CVPR*, 2018. 2

[27] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 1

[28] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *IJCV*, 2019. 2

[29] George Kingsley Zipf. *The psycho-biology of language: An introduction to dynamic philology*. Routledge, 2013. 1