Microsoft COCO: Common Objects in Context

Tsung-Yi Lin Michael Maire Serge Belongie Lubomir Bourdev Ross Girshick
James Hays Pietro Perona Deva Ramanan C. Lawrence Zitnick Piotr Dollár

Abstract—We present a new dataset with the goal of advancing the state-of-the-art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding. This is achieved by gathering images of complex everyday scenes containing common objects in their natural context. Objects are labeled using per-instance segmentations to aid in precise object localization. Our dataset contains photos of 91 objects types that would be easily recognizable by a 4 year old. With a total of 2.5 million labeled instances in 328k images, the creation of our dataset drew upon extensive crowd worker involvement via novel user interfaces for category detection, instance spotting and instance segmentation. We present a detailed statistical analysis of the dataset in comparison to PASCAL, ImageNet, and SUN. Finally, we provide baseline performance analysis for bounding box and segmentation detection results using a Deformable Parts Model.

1 Introduction

One of the primary goals of computer vision is the understanding of visual scenes. Scene understanding involves numerous tasks including recognizing what objects are present, localizing the objects in 2D and 3D, determining the objects' and scene's attributes, characterizing relationships between objects and providing a semantic description of the scene. The current object classification and detection datasets [1], [2], [3], [4] help us explore the first challenges related to scene understanding. For instance the ImageNet dataset [1], which contains an unprecedented number of images, has recently enabled breakthroughs in both object classification and detection research [5], [6], [7]. The community has also created datasets containing object attributes [8], scene attributes [9], keypoints [10], and 3D scene information [11]. This leads us to the obvious question: what datasets will best continue our advance towards our ultimate goal of scene understanding?

We introduce a new large-scale dataset that addresses three core research problems in scene understanding: detecting non-iconic views (or non-canonical perspectives [12]) of objects, contextual reasoning between objects and the precise 2D localization of objects. For many categories of objects, there exists an iconic view. For example, when performing a web-based image search for the object category "bike," the top-ranked retrieved examples appear in profile, unobstructed near the center of a neatly composed photo. We posit that current recognition systems perform fairly well on iconic views, but struggle to recognize objects otherwise – in the background, partially occluded, amid clutter [13] – reflecting the composition of actual everyday scenes. We verify this experimentally; when evaluated on everyday scenes, models trained on our data perform better than those trained with prior datasets. A challenge is finding natural images that contain multiple objects. The identity of many objects can only be resolved using context, due to small size or ambiguous appearance in the image. To push research in contextual reasoning, images depicting scenes [3] rather than objects in isolation are necessary. Finally, we argue that detailed spatial understanding of object layout will be a core component of scene analysis. An object's spatial location can be defined coarsely using a bounding box [2] or with a precise pixel-level segmentation [14], [15], [16]. As we demonstrate, to measure either kind of localization performance it is essential for the dataset to have every instance of every object

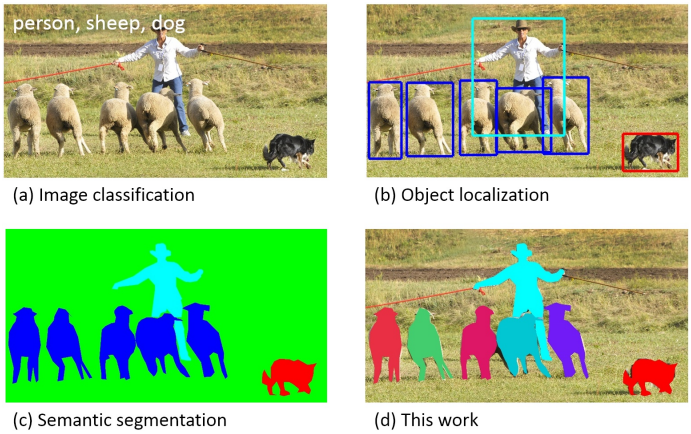


Fig. 1: While previous object recognition datasets have focused on (a) image classification, (b) object bounding box localization or (c) semantic pixel-level segmentation, we focus on (d) segmenting individual object instances. We introduce a large, richly-annotated dataset comprised of images depicting complex everyday scenes of common objects in their natural context.

- *T.Y. Lin and S. Belongie are with Cornell NYC Tech and the Cornell Computer Science Department.*
- *M. Maire is with the Toyota Technological Institute at Chicago.*
- *L. Bourdev and P. Dollár are with Facebook AI Research. The majority of this work was performed while P. Dollár was with Microsoft Research.*
- *R. Girshick and C. L. Zitnick are with Microsoft Research, Redmond.*
- *J. Hays is with Brown University.*
- *P. Perona is with the California Institute of Technology.*
- *D. Ramanan is with the University of California at Irvine.*

category labeled and fully segmented. Our dataset is unique in its annotation of instance-level segmentation masks, Fig. 1.

To create a large-scale dataset that accomplishes these three goals we employed a novel pipeline for gathering data with extensive use of Amazon Mechanical Turk. First and most importantly, we harvested a large set of images containing contextual relationships and non-iconic object views. We accomplished this using a surprisingly simple yet effective technique that queries for pairs of objects in conjunction with images retrieved via scene-based queries [17], [3]. Next, each image was labeled as containing particular object categories using a hierarchical labeling approach [18]. For each category found, the individual instances were labeled, verified, and finally segmented. Given the inherent ambiguity of labeling, each of these stages has numerous tradeoffs that we explored in detail.

The Microsoft Common Objects in COntext (MS COCO) dataset contains 91 common object categories with 82 of them having more than 5,000 labeled instances, Fig. 6. In total the dataset has 2,500,000 labeled instances in 328,000 images. In contrast to the popular ImageNet dataset [1], COCO has fewer categories but more instances per category. This can aid in learning detailed object models capable of precise 2D localization. The dataset is also significantly larger in number of instances per category than the PASCAL VOC [2] and SUN [3] datasets. Additionally, a critical distinction between our dataset and others is the number of labeled instances per image which may aid in learning contextual information, Fig. 5. MS COCO contains considerably more object instances per image (7.7) as compared to ImageNet (3.0) and PASCAL (2.3). In contrast, the SUN dataset, which contains significant contextual information, has over 17 objects and "stuff" per image but considerably fewer object instances overall.

An abridged version of this work appeared in [19].

2 Related Work

Throughout the history of computer vision research datasets have played a critical role. They not only provide a means to train and evaluate algorithms, they drive research in new and more challenging directions. The creation of ground truth stereo and optical flow datasets [20], [21] helped stimulate a flood of interest in these areas. The early evolution of object recognition datasets [22], [23], [24] facilitated the direct comparison of hundreds of image recognition algorithms while simultaneously pushing the field towards more complex problems. Recently, the ImageNet dataset [1] containing millions of images has enabled breakthroughs in both object classification and detection research using a new class of deep learning algorithms [5], [6], [7].

Datasets related to object recognition can be roughly split into three groups: those that primarily address object classification, object detection and semantic scene labeling. We address each in turn.

Image Classification The task of object classification requires binary labels indicating whether objects are present in an image; see Fig. 1(a). Early datasets of this type comprised images containing a single object with blank backgrounds, such as the MNIST handwritten digits [25] or COIL household objects [26]. Caltech 101 [22] and Caltech 256 [23] marked the transition to more realistic object images retrieved from the internet while also increasing the number of object categories to 101 and 256, respectively. Popular datasets in the machine learning community due to the larger number of training examples, CIFAR-10 and CIFAR-100 [27] offered 10 and 100 categories from a dataset of tiny 32×32 images [28]. While these datasets contained up to 60,000 images and hundreds of categories, they still only captured a small fraction of our visual world.

Recently, ImageNet [1] made a striking departure from the incremental increase in dataset sizes. They proposed the creation of a dataset containing 22k categories with 500-1000 images each. Unlike previous datasets containing entry-level categories [29], such as "dog" or "chair," like [28], ImageNet used the WordNet Hierarchy [30] to obtain both entry-level and fine-grained [31] categories. Currently, the ImageNet dataset contains over 14 million labeled images and has enabled significant advances in image classification [5], [6], [7].

Object detection Detecting an object entails both stating that an object belonging to a specified class is present, and localizing it in the image. The location of an object is typically represented by a bounding box, Fig. 1(b). Early algorithms focused on face detection [32] using various ad hoc datasets. Later, more realistic and challenging face detection datasets were created [33]. Another popular challenge is the detection of pedestrians for which several datasets have been created [24], [4]. The Caltech Pedestrian Dataset [4] contains 350,000 labeled instances with bounding boxes.

For the detection of basic object categories, a multi-year effort from 2005 to 2012 was devoted to the creation and maintenance of a series of benchmark datasets that were widely adopted. The PASCAL VOC [2] datasets contained 20 object categories spread over 11,000 images. Over 27,000 object instance bounding boxes were labeled, of which almost 7,000 had detailed segmentations. Recently, a detection challenge has been created from 200 object categories using a subset of 400,000 images from ImageNet [34]. An impressive 350,000 objects have been labeled using bounding boxes.

Since the detection of many objects such as sunglasses, cellphones or chairs is highly dependent on contextual information, it is important that detection datasets contain objects in their natural environments. In our dataset we strive to collect images rich in contextual information. The use of bounding boxes also limits the accuracy for which detection algorithms may be evaluated. We propose the use of fully segmented instances to enable more accurate detector evaluation.

(a) Iconic object images (b) Iconic scene images (c) Non-iconic images

Fig. 2: Example of (a) iconic object images, (b) iconic scene images, and (c) non-iconic images.

Semantic scene labeling The task of labeling semantic objects in a scene requires that each pixel of an image be labeled as belonging to a category, such as sky, chair, floor, street, etc. In contrast to the detection task, individual instances of objects do not need to be segmented, Fig. 1(c). This enables the labeling of objects for which individual instances are hard to define, such as grass, streets, or walls. Datasets exist for both indoor [11] and outdoor [35], [14] scenes. Some datasets also include depth information [11]. Similar to semantic scene labeling, our goal is to measure the pixel-wise accuracy of object labels. However, we also aim to distinguish between individual instances of an object, which requires a solid understanding of each object's extent.

A novel dataset that combines many of the properties of both object detection and semantic scene labeling datasets is the SUN dataset [3] for scene understanding. SUN contains 908 scene categories from the WordNet dictionary [30] with segmented objects. The 3,819 object categories span those common to object detection datasets (person, chair, car) and to semantic scene labeling (wall, sky, floor). Since the dataset was collected by finding images depicting various scene types, the number of instances per object category exhibits the long tail phenomenon. That is, a few categories have a large number of instances (wall: 20,213, window: 16,080, chair: 7,971) while most have a relatively modest number of instances (boat: 349, airplane: 179, floor lamp: 276). In our dataset, we ensure that each object category has a significant number of instances, Fig. 5.

Other vision datasets Datasets have spurred the advancement of numerous fields in computer vision. Some notable datasets include the Middlebury datasets for stereo vision [20], multi-view stereo [36] and optical flow [21]. The Berkeley Segmentation Data Set (BSDS500) [37] has been used extensively to evaluate both segmentation and edge detection algorithms. Datasets have also been created to recognize both scene [9] and object attributes [8], [38]. Indeed, numerous areas of vision have benefited from challenging datasets that helped catalyze progress.

3 Image Collection

We next describe how the object categories and candidate images are selected.

3.1 Common Object Categories

The selection of object categories is a non-trivial exercise. The categories must form a representative set of all categories, be relevant to practical applications and occur with high enough frequency to enable the collection of a large dataset. Other important decisions are whether to include both "thing" and "stuff" categories [39] and whether fine-grained [31], [1] and object-part categories should be included. "Thing" categories include objects for which individual instances may be easily labeled (person, chair, car) where "stuff" categories include materials and objects with no clear boundaries (sky, street, grass). Since we are primarily interested in precise localization of object instances, we decided to only include "thing" categories and not "stuff." However, since "stuff" categories can provide significant contextual information, we believe the future labeling of "stuff" categories would be beneficial.

The specificity of object categories can vary significantly. For instance, a dog could be a member of the "mammal", "dog", or "German shepherd" categories. To enable the practical collection of a significant number of instances per category, we chose to limit our dataset to entry-level categories, i.e. category labels that are commonly used by humans when describing objects (dog, chair, person). It is also possible that some object categories may be parts of other object categories. For instance, a face may be part of a person. We anticipate the inclusion of object-part categories (face, hands, wheels) would be beneficial for many real-world applications.

We used several sources to collect entry-level object categories of "things." We first compiled a list of categories by combining categories from PASCAL VOC [2] and a subset of the 1200 most frequently used words that denote visually identifiable objects [40]. To further augment our set of candidate categories, several children ranging in ages from 4 to 8 were asked to name every

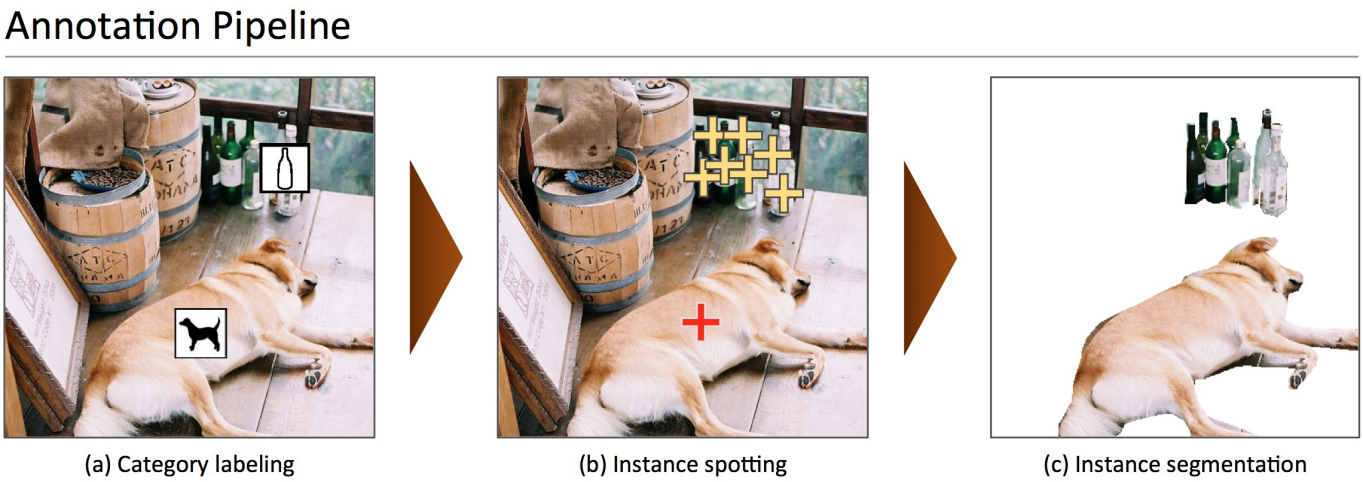


Fig. 3: Our annotation pipeline is split into 3 primary tasks: (a) labeling the categories present in the image (§4.1), (b) locating and marking all instances of the labeled categories (§4.2), and (c) segmenting each object instance (§4.3).

object they see in indoor and outdoor environments. The final 272 candidates may be found in the appendix. Finally, the co-authors voted on a 1 to 5 scale for each category taking into account how commonly they occur, their usefulness for practical applications, and their diversity relative to other categories. The final selection of categories attempts to pick categories with high votes, while keeping the number of categories per supercategory (animals, vehicles, furniture, etc.) balanced. Categories for which obtaining a large number of instances (greater than 5,000) was difficult were also removed. To ensure backwards compatibility all categories from PASCAL VOC [2] are also included. Our final list of 91 proposed categories is in Fig. 5(a).

3.2 Non-iconic Image Collection

Given the list of object categories, our next goal was to collect a set of candidate images. We may roughly group images into three types, Fig. 2: iconic-object images [41], iconic-scene images [3] and non-iconic images. Typical iconic-object images have a single large object in a canonical perspective centered in the image, Fig. 2(a). Iconic-scene images are shot from canonical viewpoints and commonly lack people, Fig. 2(b). Iconic images have the benefit that they may be easily found by directly searching for specific categories using Google or Bing image search. While iconic images generally provide high quality object instances, they can lack important contextual information and non-canonical viewpoints.

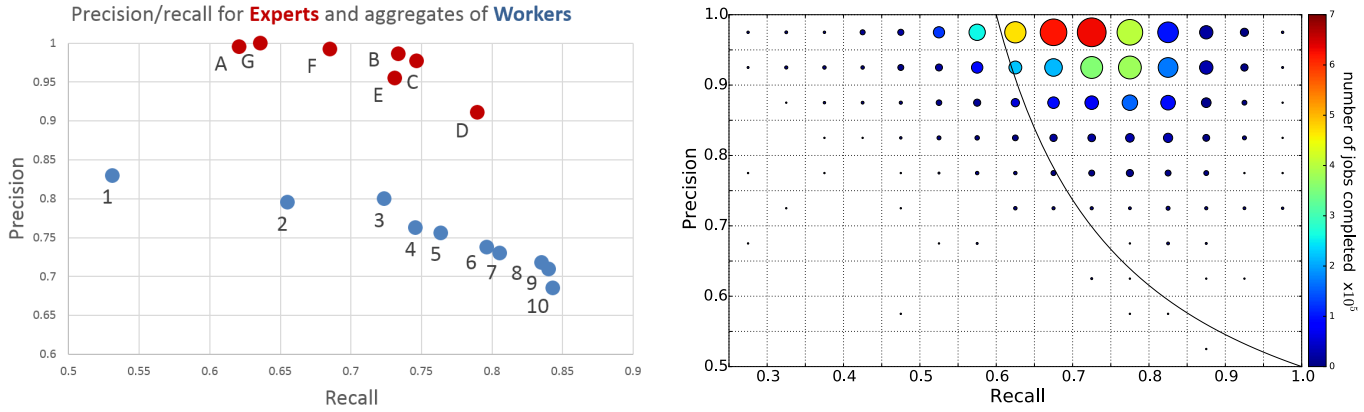
Our goal was to collect a dataset such that a majority of images are non-iconic, Fig. 2(c). It has been shown that datasets containing more non-iconic images are better at generalizing [42]. We collected non-iconic images using two strategies. First as popularized by PASCAL VOC [2], we collected images from Flickr which tends to have fewer iconic images. Flickr contains photos uploaded by amateur photographers with searchable metadata and keywords. Second, we did not search for object categories in isolation. A search for "dog" will tend to return iconic images of large, centered dogs. However, if we searched for pairwise combinations of object categories, such as "dog + car" we found many more non-iconic images. Surprisingly, these images typically do not just contain the two categories specified in the search, but numerous other categories as well. To further supplement our dataset we also searched for scene/object category pairs, see the appendix. We downloaded at most 5 photos taken by a single photographer within a short time window. In the rare cases in which enough images could not be found, we searched for single categories and performed an explicit filtering stage to remove iconic images. The result is a collection of 328,000 images with rich contextual relationships between objects as shown in Figs. 2(c) and 6.

4 Image Annotation

We next describe how we annotated our image collection. Due to our desire to label over 2.5 million object instances, the design of a cost efficient yet high quality annotation pipeline was critical. The annotation pipeline is outlined in Fig. 3. For all crowdsourcing tasks we used workers on Amazon's Mechanical Turk (AMT). Our user interfaces are described in detail in the appendix. Note that, since the original version of this work [19], we have taken a number of steps to further improve the quality of the annotations. In particular, we have increased the number of annotators for the category labeling and instance spotting stages to eight. We also added a stage to verify the instance segmentations.

4.1 Category Labeling

The first task in annotating our dataset is determining which object categories are present in each image, Fig. 3(a). Since we have 91 categories and a large number of images, asking workers to answer 91 binary classification questions per image would be prohibitively expensive. Instead, we used a hierarchical approach [18].



(a) (b)

Fig. 4: Worker precision and recall for the category labeling task. (a) The union of multiple AMT workers (blue) has better recall than any expert (red). Ground truth was computed using majority vote of the experts. (b) Shows the number of workers (circle size) and average number of jobs per worker (circle color) for each precision/recall range. Most workers have high precision; such workers generally also complete more jobs. For this plot ground truth for each worker is the *union* of responses from all other AMT workers. See §4.4 for details.

We group the object categories into 11 super-categories (see the appendix). For a given image, a worker was presented with each group of categories in turn and asked to indicate whether any instances exist for that super-category. This greatly reduces the time needed to classify the various categories. For example, a worker may easily determine no animals are present in the image without having to specifically look for cats, dogs, etc. If a worker determines instances from the super-category (animal) are present, for each subordinate category (dog, cat, etc.) present, the worker must drag the category's icon onto the image over one instance of the category. The placement of these icons is critical for the following stage. We emphasize that only a single instance of each category needs to be annotated in this stage. To ensure high recall, 8 workers were asked to label each image. A category is considered present if any worker indicated the category; false positives are handled in subsequent stages. A detailed analysis of performance is presented in §4.4. This stage took ∼20k worker hours to complete.

4.2 Instance Spotting

In the next stage all instances of the object categories in an image were labeled, Fig. 3(b). In the previous stage each worker labeled one instance of a category, but multiple object instances may exist. Therefore, for each image, a worker was asked to place a cross on top of each instance of a specific category found in the previous stage. To boost recall, the location of the instance found by a worker in the previous stage was shown to the current worker. Such priming helped workers quickly find an initial instance upon first seeing the image. The workers could also use a magnifying glass to find small instances. Each worker was asked to label at most 10 instances of a given category per image. Each image was labeled by 8 workers for a total of ∼10k worker hours.

4.3 Instance Segmentation

Our final stage is the laborious task of segmenting each object instance, Fig. 3(c). For this stage we modified the excellent user interface developed by Bell et al. [16] for image segmentation. Our interface asks the worker to segment an object instance specified by a worker in the previous stage. If other instances have already been segmented in the image, those segmentations are shown to the worker. A worker may also indicate there are no object instances of the given category in the image (implying a false positive label from the previous stage) or that all object instances are already segmented.

Segmenting 2,500,000 object instances is an extremely time consuming task requiring over 22 worker hours per 1,000 segmentations. To minimize cost we only had a single worker segment each instance. However, when first completing the task, most workers produced only coarse instance outlines. As a consequence, we required all workers to complete a training task for each object category. The training task required workers to segment an object instance. Workers could not complete the task until their segmentation adequately matched the ground truth. The use of a training task vastly improved the quality of the workers (approximately 1 in 3 workers passed the training stage) and resulting segmentations. Example segmentations may be viewed in Fig. 6.

While the training task filtered out most bad workers, we also performed an explicit verification step on each segmented instance to ensure good quality. Multiple workers (3 to 5) were asked to judge each segmentation and indicate whether it matched the instance well or not. Segmentations of insufficient quality were discarded and the corresponding instances added back to the pool of unsegmented objects. Finally, some approved workers consistently produced poor segmentations; all work obtained from such workers was discarded.

For images containing 10 object instances or fewer of a given category, every instance was individually segmented (note that in some images up to 15 instances were segmented). Occasionally the number of instances is drastically higher; for example, consider a dense crowd of people or a truckload of bananas. In such cases, many instances of the same category may be tightly grouped together and distinguishing individual instances is difficult. After 10-15 instances of a category were segmented in an image, the remaining instances were marked as "crowds" using a single (possibly multi-part) segment. For the purpose of evaluation, areas marked as crowds will be ignored and not affect a detector's score. Details are given in the appendix.

4.4 Annotation Performance Analysis

We analyzed crowd worker quality on the category labeling task by comparing to dedicated expert workers, see Fig. 4(a). We compared precision and recall of seven expert workers (co-authors of the paper) with the results obtained by taking the union of one to ten AMT workers. Ground truth was computed using majority vote of the experts. For this task recall is of primary importance as false positives could be removed in later stages. Fig. 4(a) shows that the union of 8 AMT workers, the same number as was used to collect our labels, achieved greater recall than any of the expert workers. Note that worker recall saturates at around 9-10 AMT workers.

Object category presence is often ambiguous. Indeed as Fig. 4(a) indicates, even dedicated experts often disagree on object presence, e.g. due to inherent ambiguity in the image or disagreement about category definitions. For any unambiguous examples having a probability of over 50% of being annotated, the probability all 8 annotators missing such a case is at most $.5^8 \approx .004$. Additionally, by observing how recall increased as we added annotators, we estimate that in practice over 99% of all object categories not later rejected as false positives are detected given 8 annotators. Note that a similar analysis may be done for instance spotting in which 8 annotators were also used.

Finally, Fig. 4(b) re-examines precision and recall of AMT workers on category labeling on a much larger set of images. The number of workers (circle size) and average number of jobs per worker (circle color) is shown for each precision/recall range. Unlike in Fig. 4(a), we used a leave-one-out evaluation procedure where a category was considered present if *any* of the remaining workers named the category. Therefore, overall worker precision is substantially higher. Workers who completed the most jobs also have the highest precision; all jobs from workers below the black line were rejected.

4.5 Caption Annotation

We added five written caption descriptions to each image in MS COCO. A full description of the caption statistics and how they were gathered will be provided shortly in a separate publication.

5 Dataset Statistics

Next, we analyze the properties of the Microsoft Common Objects in COntext (MS COCO) dataset in comparison to several other popular datasets. These include ImageNet [1], PASCAL VOC 2012 [2], and SUN [3]. Each of these datasets varies significantly in size, list of labeled categories and types of images. ImageNet was created to capture a large number of object categories, many of which are fine-grained. SUN focuses on labeling scene types and the objects that commonly occur in them. Finally, PASCAL VOC's primary application is object detection in natural images. MS COCO is designed for the detection and segmentation of objects occurring in their natural context.

The number of instances per category for all 91 categories is shown in Fig. 5(a). A summary of the datasets showing the number of object categories and the number of instances per category is shown in Fig. 5(d). While MS COCO has fewer categories than ImageNet and SUN, it has more instances per category which we hypothesize will be useful for learning complex models capable of precise localization. In comparison to PASCAL VOC, MS COCO has both more categories and instances.

An important property of our dataset is we strive to find non-iconic images containing objects in their natural context. The amount of contextual information present in an image can be estimated by examining the average number of object categories and instances per image, Fig. 5(b, c). For ImageNet we plot the object detection validation set, since the training data only has a single object labeled. On average our dataset contains 3.5 categories and 7.7 instances per image. In comparison ImageNet and PASCAL VOC both have less than 2 categories and 3 instances per image on average. Another interesting observation is only 10% of the images in MS COCO have only one category per image, in comparison, over 60% of images contain a single object category in ImageNet and PASCAL VOC. As expected, the SUN dataset has the most contextual information since it is scene-based and uses an unrestricted set of categories.

Finally, we analyze the average size of objects in the datasets. Generally smaller objects are harder to recognize and require more contextual reasoning to recognize. As shown in Fig. 5(e), the average sizes of objects is smaller for both MS COCO and SUN.

6 Dataset Splits

To accommodate a faster release schedule, we split the MS COCO dataset into two roughly equal parts. The first half of the dataset was released in 2014, the second half will be released in 2015. The 2014 release contains 82,783 training, 40,504 validation, and 40,775 testing images (approximately $\frac{1}{2}$ train, $\frac{1}{4}$ val, and $\frac{1}{4}$ test). There are nearly 270k segmented people and a total of 886k segmented object instances in the 2014 train+val data alone. The cumulative 2015 release will contain a total of 165,482 train, 81,208 val, and 81,434 test images.

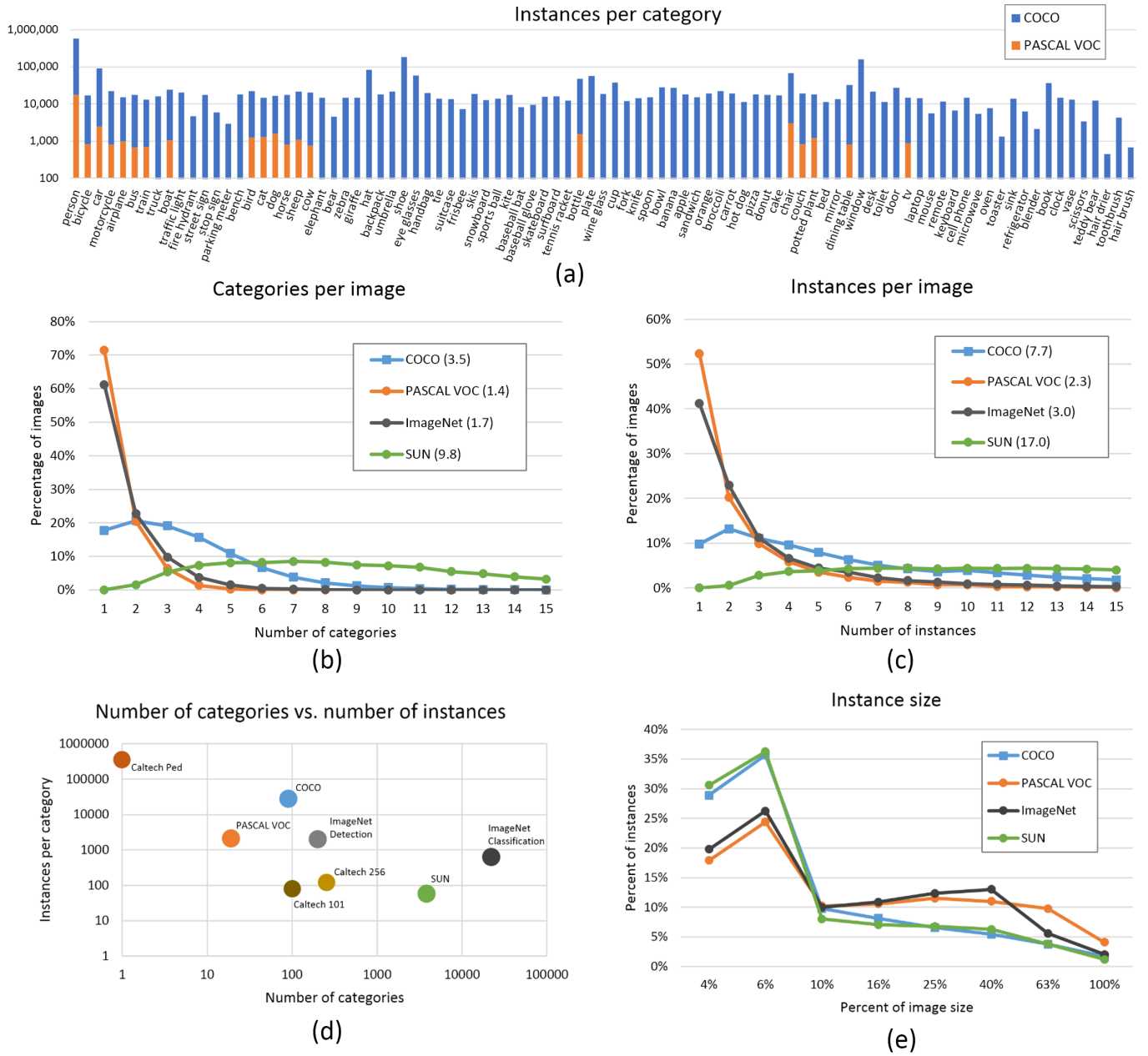


Fig. 5: (a) Number of annotated instances per category for MS COCO and PASCAL VOC. (b,c) Number of annotated categories and annotated instances, respectively, per image for MS COCO, ImageNet Detection, PASCAL VOC and SUN (average number of categories and instances are shown in parentheses). (d) Number of categories vs. the number of instances per category for a number of popular object recognition datasets. (e) The distribution of instance sizes for the MS COCO, ImageNet Detection, PASCAL VOC and SUN datasets.

We took care to minimize the chance of near-duplicate images existing across splits by explicitly removing near duplicates (detected with [43]) and grouping images by photographer and date taken.

Following established protocol, annotations for train and validation data will be released, but not for test. We are currently finalizing the evaluation server for automatic evaluation on the test set. A full discussion of evaluation metrics will be added once the evaluation server is complete.

Note that we have limited the 2014 release to a subset of 80 categories. We did not collect segmentations for the following 11 categories: hat, shoe, eyeglasses (too many instances), mirror, window, door, street sign (ambiguous and difficult to label), plate, desk (due to confusion with bowl and dining table, respectively) and blender, hair brush (too few instances). We may add segmentations for some of these categories in the cumulative 2015 release.

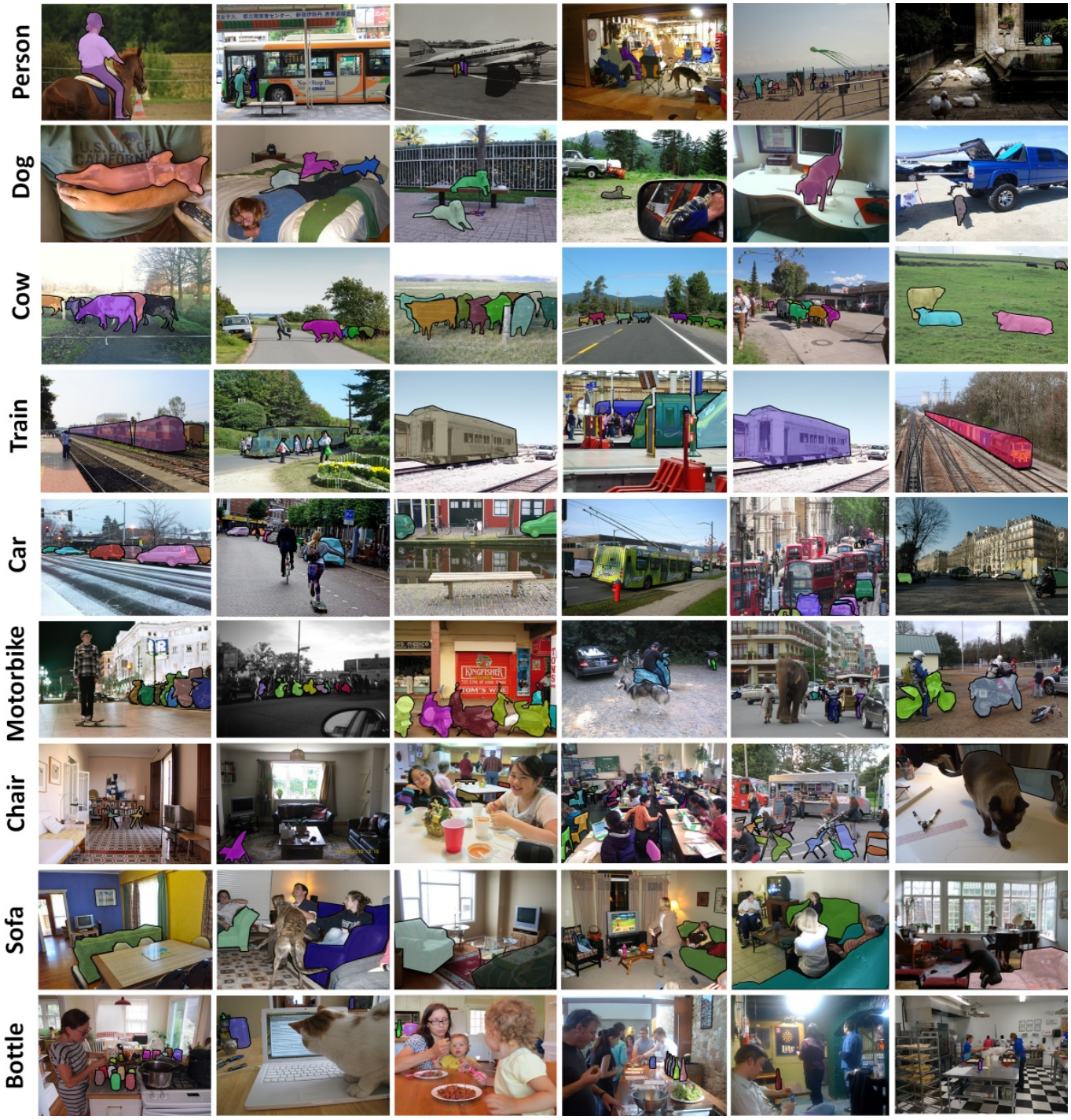


Fig. 6: Samples of annotated images in the MS COCO dataset.

7 Algorithmic Analysis

Bounding-box detection For the following experiments we take a subset of 55,000 images from our dataset[1] and obtain tight-fitting bounding boxes from the annotated segmentation masks. We evaluate models tested on both MS COCO and PASCAL, see Table 1. We evaluate two different models. **DPMv5-P**: the latest implementation of [44] (release 5 [45]) trained on PASCAL VOC 2012. **DPMv5-C**: the same implementation trained on COCO (5000 positive and 10000 negative images). We use the default parameter settings for training COCO models.

If we compare the average performance of DPMv5-P on PASCAL VOC and MS COCO, we find that average performance on MS COCO drops by nearly a *factor of 2*, suggesting that MS COCO does include more difficult (non-iconic) images of objects that are partially occluded, amid clutter, etc. We notice a similar drop in performance

1. These preliminary experiments were performed before our final split of the dataset intro train, val, and test. Baselines on the actual test set will be added once the evaluation server is complete.

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	moto	person	plant	sheep	sofa	train	tv	avg.
DPMv5-P	**45.6**	49.0	11.0	**11.6**	**27.2**	50.5	**43.1**	**23.6**	**17.2**	23.2	**10.7**	**20.5**	42.5	**44.5**	**41.3**	**8.7**	**29.0**	**18.7**	**40.0**	34.5	**29.6**
DPMv5-C	43.7	**50.1**	**11.8**	2.4	21.4	**60.1**	35.6	16.0	11.4	**24.8**	5.3	9.4	**44.5**	41.0	35.8	6.3	28.3	13.3	38.8	**36.2**	26.8
DPMv5-P	35.1	17.9	3.7	2.3	**7**	45.4	**18.3**	8.6	**6.3**	17	4.8	**5.8**	35.3	25.4	**17.5**	4.1	**14.5**	9.6	31.7	27.9	16.9
DPMv5-C	**36.9**	**20.2**	**5.7**	**3.5**	6.6	**50.3**	16.1	**12.8**	4.5	**19.0**	**9.6**	4.0	**38.2**	**29.9**	15.9	**6.7**	13.8	**10.4**	**39.2**	**37.9**	**19.1**

TABLE 1: **Top**: Detection performance evaluated on **PASCAL VOC 2012**. DPMv5-P is the performance reported by Girshick et al. in VOC release 5. DPMv5-C uses the same implementation, but is trained with MS COCO. **Bottom**: Performance evaluated on **MS COCO** for DPM models trained with PASCAL VOC 2012 (DPMv5-P) and MS COCO (DPMv5-C). For DPMv5-C we used 5000 positive and 10000 negative training examples. While MS COCO is considerably more challenging than PASCAL, use of more training data coupled with more sophisticated approaches [5], [6], [7] should improve performance substantially.

for the model trained on MS COCO (DPMv5-C).

The effect on detection performance of training on PASCAL VOC or MS COCO may be analyzed by comparing DPMv5-P and DPMv5-C. They use the same implementation with different sources of training data. Table 1 shows DPMv5-C still outperforms DPMv5-P in 6 out of 20 categories when testing on PASCAL VOC. In some categories (e.g., dog, cat, people), models trained on MS COCO perform worse, while on others (e.g., bus, tv, horse), models trained on our data are better.

Consistent with past observations [46], we find that including difficult (non-iconic) images during training may not always help. Such examples may act as noise and pollute the learned model if the model is not rich enough to capture such appearance variability. Our dataset allows for the exploration of such issues.

Torralba and Efros [42] proposed a metric to measure cross-dataset generalization which computes the 'performance drop' for models that train on one dataset and test on another. The performance difference of the DPMv5-P models across the two datasets is 12.7 AP while the DPMv5-C models only have 7.7 AP difference. Moreover, overall performance is much lower on MS COCO. These observations support two hypotheses: 1) MS COCO is significantly more difficult than PASCAL VOC and 2) models trained on MS COCO can generalize better to easier datasets such as PASCAL VOC given more training data. To gain insight into the differences between the datasets, see the appendix for visualizations of person and chair examples from the two datasets.

Generating segmentations from detections We now describe a simple method for generating object bounding boxes and segmentation masks, following prior work that produces segmentations from object detections [47], [48], [49], [50]. We learn aspect-specific pixel-level segmentation masks for different categories. These are readily learned by averaging together segmentation masks from aligned training instances. We learn different masks corresponding to the different mixtures in our DPM detector. Sample masks are visualized in Fig. 7.

Detection evaluated by segmentation Segmentation is a challenging task even assuming a detector reports correct results as it requires fine localization of object part boundaries. To decouple segmentation evaluation from detection correctness, we benchmark segmentation quality using only correct detections. Specifically, given that the detector reports a correct bounding box, how well does the predicted segmentation of that object match the ground truth segmentation? As criterion for correct detection, we impose the standard requirement that intersection over union between predicted and ground truth boxes is at least 0.5. We then measure the intersection over union of the predicted and ground truth segmentation masks, see Fig. 8. To establish a baseline for our dataset, we project learned DPM part masks onto the image to create segmentation masks. Fig. 9 shows results of this segmentation baseline for the DPM learned on the 20 PASCAL categories and tested on our dataset.

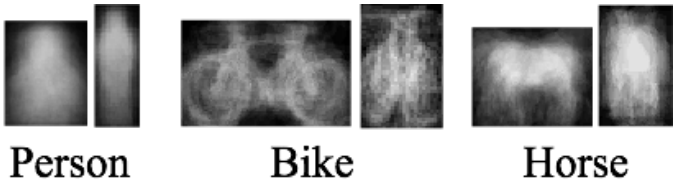


Fig. 7: We visualize our mixture-specific shape masks. We paste thresholded shape masks on each candidate detection to generate candidate segments.

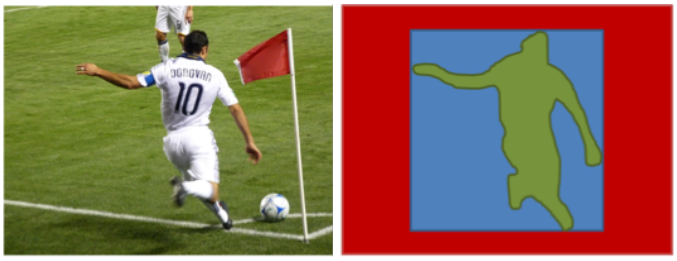


Fig. 8: Evaluating instance detections with segmentation masks versus bounding boxes. Bounding boxes are a particularly crude approximation for articulated objects; in this case, the majority of the pixels in the (**blue**) tight-fitting bounding-box do not lie on the object. Our (**green**) instance-level segmentation masks allows for a more accurate measure of object detection and localization.

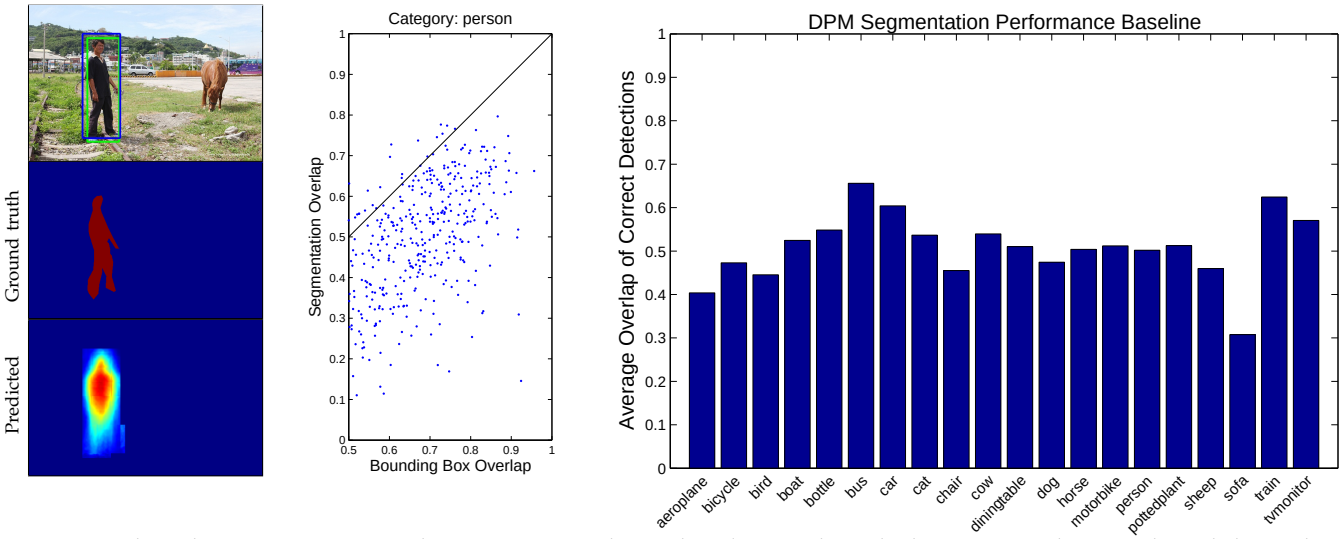


Fig. 9: A predicted segmentation might not recover object detail even though detection and ground truth bounding boxes overlap well (left). Sampling from the person category illustrates that predicting segmentations from top-down projection of DPM part masks is difficult even for correct detections (center). Average segmentation overlap measured on MS COCO for the 20 PASCAL VOC categories demonstrates the difficulty of the problem (right).

8 Discussion

We introduced a new dataset for detecting and segmenting objects found in everyday life in their natural environments. Utilizing over 70,000 worker hours, a vast collection of object instances was gathered, annotated and organized to drive the advancement of object detection and segmentation algorithms. Emphasis was placed on finding non-iconic images of objects in natural environments and varied viewpoints. Dataset statistics indicate the images contain rich contextual information with many objects present per image.

There are several promising directions for future annotations on our dataset. We currently only label "things", but labeling "stuff" may also provide significant contextual information that may be useful for detection. Many object detection algorithms benefit from additional annotations, such as the amount an instance is occluded [4] or the location of keypoints on the object [10]. Finally, our dataset could provide a good benchmark for other types of labels, including scene types [3], attributes [9], [8] and full sentence written descriptions [51]. We are actively exploring adding various such annotations.

To download and learn more about MS COCO please see the project website[2]. MS COCO will evolve and grow over time; up to date information is available online.

Acknowledgments Funding for all crowd worker tasks was provided by Microsoft. P.P. and D.R. were supported by ONR MURI Grant N00014-10-1-0933. We would like to thank all members of the community who provided valuable feedback throughout the process of defining and collecting the dataset.

2. http://mscoco.org/

Appendix Overview

In the appendix, we provide detailed descriptions of the AMT user interfaces and the full list of 272 candidate categories (from which our final 91 were selected) and 40 scene categories (used for scene-object queries).

Appendix I: User Interfaces

We describe and visualize our user interfaces for collecting non-iconic images, category labeling, instance spotting, instance segmentation, segmentation verification and finally crowd labeling.

Non-iconic Image Collection Flickr provides a rich image collection associated with text captions. However, captions might be inaccurate and images may be iconic. To construct a high-quality set of non-iconic images, we first collected candidate images by searching for pairs of object categories, or pairs of object and scene categories. We then created an AMT filtering task that allowed users to remove invalid or iconic images from a grid of 128 candidates, Fig. 10. We found the choice of instructions to be crucial, and so provided users with examples of iconic and non-iconic images. Some categories rarely co-occurred with others. In such cases, we collected candidates using only the object category as the search term, but apply a similar filtering step, Fig. 10(b).

Category Labeling Fig. 12(a) shows our interface for category labeling. We designed the labeling task to encourage workers to annotate all categories present in the image. Workers annotate categories by dragging and dropping icons from the bottom category panel onto a corresponding object instance. Only a single instance of each object category needs to be annotated in the image. We group icons by the super-categories from Fig. 11, allowing workers to quickly skip categories that are unlikely to be present.

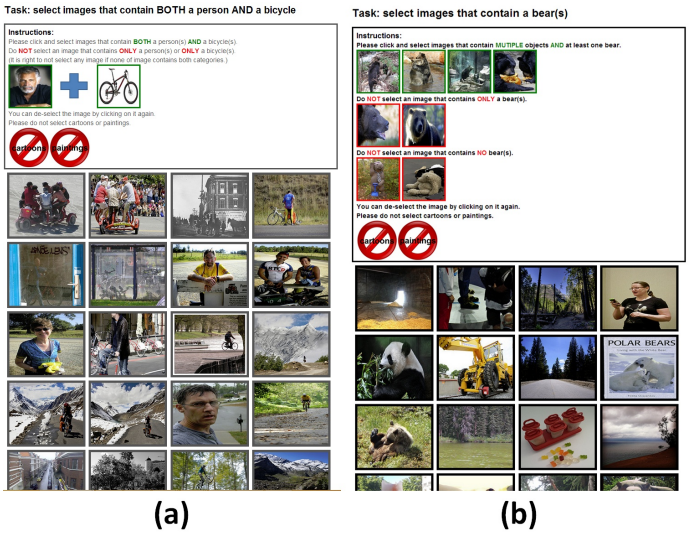


Fig. 10: User interfaces for non-iconic image collection. (a) Interface for selecting non-iconic images containing pairs of objects. (b) Interface for selecting non-iconic images for categories that rarely co-occurred with others.

Instance Spotting Fig. 12(b) depicts our interface for labeling all instances of a given category. The interface is initialized with a blinking icon specifying a single instance obtained from the previous category-labeling stage. Workers are then asked to spot and click on up to 10 total instances of the given category, placing a single cross anywhere within the region of each instance. In order to spot small objects, we found it crucial to include a "magnifying glass" feature that doubles the resolution of a worker's currently selected region.

Instance Segmentation Fig. 12(c) shows our user interface for instance segmentation. We modified source code from the OpenSurfaces project [16], which defines a single AMT task for segmenting multiple regions of a homogenous material in real-scenes. In our case, we define a single task for segmenting a single object instance labeled from the previous annotation stage. To aid the segmentation process, we added a visualization of the object category icon to remind workers of the category to be segmented. Crucially, we also added zoom-in functionality to allow for efficient annotation of small objects and curved boundaries. In the previous annotation stage, to ensure high coverage of all object instances, we used multiple workers to label all instances per image. We would like to segment *all* such object instances, but instance annotations across different workers may refer to different or redundant instances. To resolve this correspondence ambiguity, we sequentially post AMT segmentation tasks, ignoring instance annotations that are already covered by an existing segmentation mask.

Segmentation Verification Fig. 12(d) shows our user interface for segmentation verification. Due to the time consuming nature of the previous task, each object instance is segmented only once. The purpose of the verification stage is therefore to ensure that each segmented instance from the previous stage is of sufficiently high quality. Workers are shown a grid of 64 segmentations and asked to select poor quality segmentations. Four of the 64 segmentation are known to be bad; a worker must identify 3 of the 4 known bad segmentations to complete the task. Each segmentation is initially shown to 3 annotators. If any of the annotators indicates the segmentation is bad, it is shown to 2 additional workers. At this point, any segmentation that doesn't receive at least 4 of 5 favorable votes is discarded and the corresponding instance added back to the pool of unsegmented objects. Examples of borderline cases that either passed (4/5 votes) or were rejected (3/5 votes) are shown in Fig. 15.

Crowd Labeling Fig. 12(e) shows our user interface for crowd labeling. As discussed, for images containing ten object instances or fewer of a given category, every object instance was individually segmented. In some images, however, the number of instances of a given category is much higher. In such cases crowd labeling provided a more efficient method for annotation. Rather than requiring workers to draw exact polygonal masks around each object instance, we allow workers to "paint" all pixels belonging to the category in question. Crowd labeling is similar to semantic segmentation as object instance are not individually identified. We emphasize that crowd labeling is only necessary for images containing more than ten object instances of a given category.

Appendix II: Object & Scene Categories

Our dataset contains 91 object categories (the 2014 release contains segmentation masks for 80 of these categories). We began with a list of frequent object categories taken from WordNet, LabelMe, SUN and other sources as well as categories derived from a free recall experiment with young children. The authors then voted on the resulting 272 categories with the aim of sampling a diverse and computationally challenging set of categories; see §3 for details. The list in Table 2 enumerates those 272 categories in descending order of votes. As discussed, the final selection of 91 categories attempts to pick categories with high votes, while keeping the number of categories per super-category (animals, vehicles, furniture, etc.) balanced.

As discussed in §3, in addition to using object-object queries to gather non-iconic images, object-scene queries also proved effective. For this task we selected a subset of 40 scene categories from the SUN dataset that frequently co-occurred with object categories of interest. Table 3 enumerates the 40 scene categories (evenly split between indoor and outdoor scenes).

REFERENCES

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.

[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *IJCV*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[3] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *CVPR*, 2010.

[4] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *PAMI*, vol. 34, 2012.

[5] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *NIPS*, 2012.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[7] P. Sermanet, D. Eigen, S. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," in *ICLR*, April 2014.

[8] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *CVPR*, 2009.

[9] G. Patterson and J. Hays, "SUN attribute database: Discovering, annotating, and recognizing scene attributes," in *CVPR*, 2012.

[10] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3D human pose annotations," in *ICCV*, 2009.

[11] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *ECCV*, 2012.

[12] S. Palmer, E. Rosch, and P. Chase, "Canonical perspective and the perception of objects," *Attention and performance IX*, vol. 1, p. 4, 1981.

[13] . Hoiem, D, Y. Chodpathumwan, and Q. Dai, "Diagnosing error in object detectors," in *ECCV*, 2012.

[14] G. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *PRL*, vol. 30, no. 2, pp. 88–97, 2009.

[15] B. Russell, A. Torralba, K. Murphy, and W. Freeman, "LabelMe: a database and web-based tool for image annotation," *IJCV*, vol. 77, no. 1-3, pp. 157–173, 2008.

[16] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "OpenSurfaces: A richly annotated catalog of surface appearance," *SIGGRAPH*, vol. 32, no. 4, 2013.

[17] V. Ordonez, G. Kulkarni, and T. Berg, "Im2text: Describing images using 1 million captioned photographs." in *NIPS*, 2011.

[18] J. Deng, O. Russakovsky, J. Krause, M. Bernstein, A. Berg, and L. Fei-Fei, "Scalable multi-label annotation," in *CHI*, 2014.

[19] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *ECCV*, 2014.

[20] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IJCV*, vol. 47, no. 1-3, pp. 7–42, 2002.

[21] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *IJCV*, vol. 92, no. 1, pp. 1–31, 2011.

[22] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *CVPR Workshop of Generative Model Based Vision (WGMBV)*, 2004.

[23] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Institute of Technology, Tech. Rep. 7694, 2007.

[24] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[25] Y. Lecun and C. Cortes, "The MNIST database of handwritten digits," 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[26] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (coil-20)," Columbia Universty, Tech. Rep., 1996.

[27] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Computer Science Department, University of Toronto, Tech. Rep*, 2009.

[28] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *PAMI*, vol. 30, no. 11, pp. 1958–1970, 2008.

[29] V. Ordonez, J. Deng, Y. Choi, A. Berg, and T. Berg, "From large scale image categorization to entry-level categories," in *ICCV*, 2013.

[30] C. Fellbaum, *WordNet: An electronic lexical database*. Blackwell Books, 1998.

[31] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," Caltech, Tech. Rep. CNS-TR-201, 2010.

[32] E. Hjelmås and B. Low, "Face detection: A survey," *CVIU*, vol. 83, no. 3, pp. 236–274, 2001.

[33] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[34] O. Russakovsky, J. Deng, Z. Huang, A. Berg, and L. Fei-Fei, "Detecting avocados to zucchinis: what have we done, and where are we going?" in *ICCV*, 2013.

[35] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *IJCV*, vol. 81, no. 1, pp. 2–23, 2009.

[36] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *CVPR*, 2006.

[37] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *PAMI*, vol. 33, no. 5, pp. 898–916, 2011.

[38] C. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *CVPR*, 2009.

[39] G. Heitz and D. Koller, "Learning spatial context: Using stuff to find things," in *ECCV*, 2008.

[40] R. Sitton, *Spelling Sourcebook*. Egger Publishing, 1996.

[41] T. Berg and A. Berg, "Finding iconic images," in *CVPR*, 2009.

[42] A. Torralba and A. Efros, "Unbiased look at dataset bias," in *CVPR*, 2011.

[43] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid, "Evaluation of gist descriptors for web-scale image search," in *CIVR*, 2009.

[44] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *PAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.

[45] R. Girshick, P. Felzenszwalb, and D. McAllester, "Discriminatively trained deformable part models, release 5," *PAMI*, 2012.

[46] X. Zhu, C. Vondrick, D. Ramanan, and C. Fowlkes, "Do we need more training data or better models for object detection?" in *BMVC*, 2012.

[47] T. Brox, L. Bourdev, S. Maji, and J. Malik, "Object segmentation by alignment of poselet activations to image contours," in *CVPR*, 2011.

[48] Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes, "Layered object models for image segmentation," *PAMI*, vol. 34, no. 9, pp. 1731–1743, 2012.

[49] D. Ramanan, "Using segmentation to verify object hypotheses," in *CVPR*, 2007.

[50] Q. Dai and D. Hoiem, "Learning to localize detected objects," in *CVPR*, 2012.

[51] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, "Collecting image annotations using Amazon's Mechanical Turk," in *NAACL Workshop*, 2010.

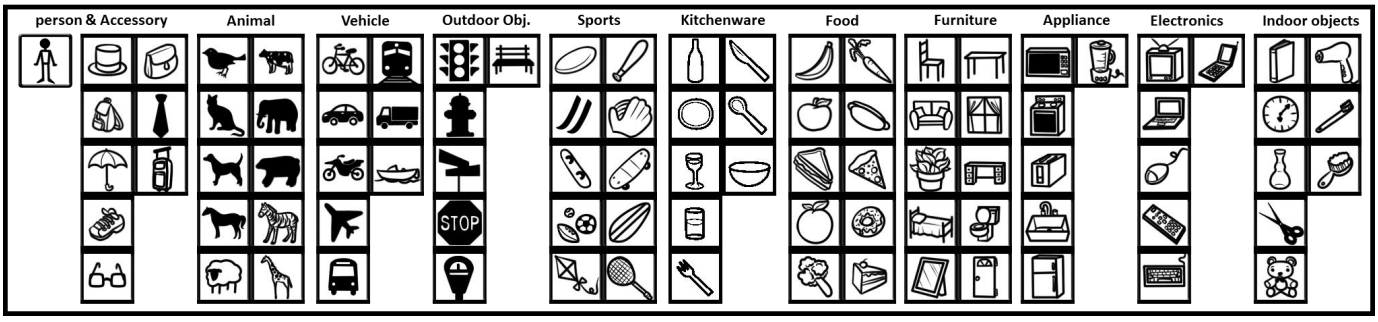
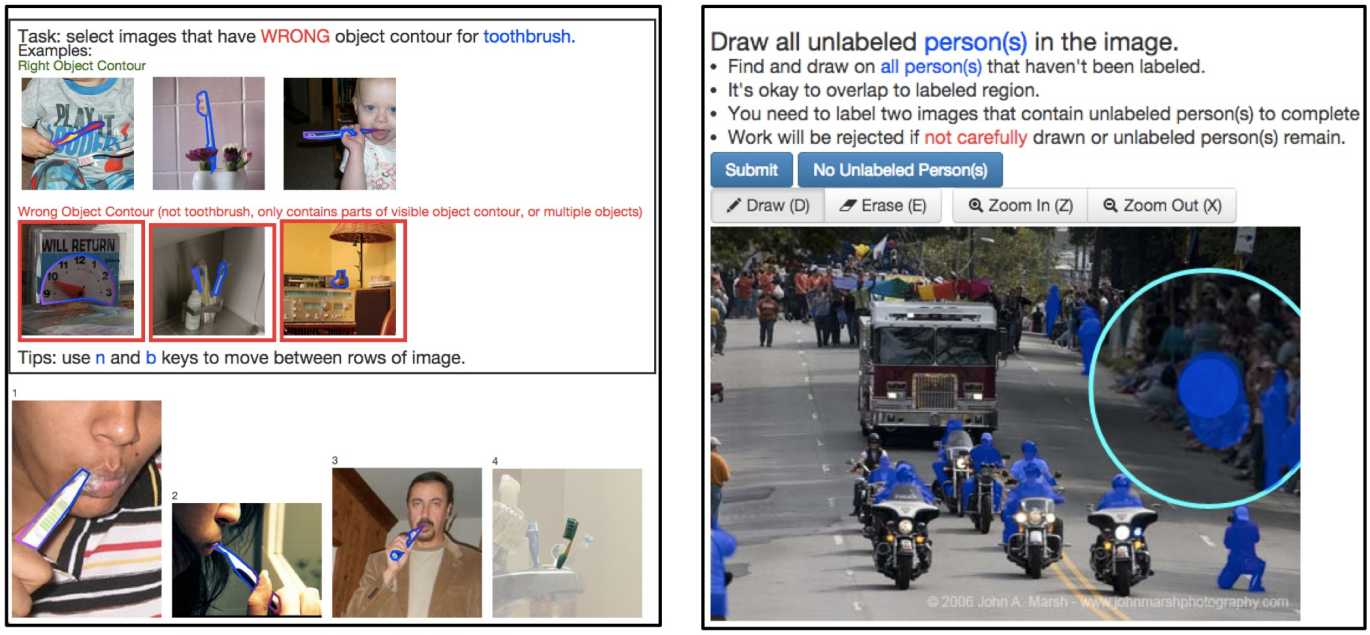


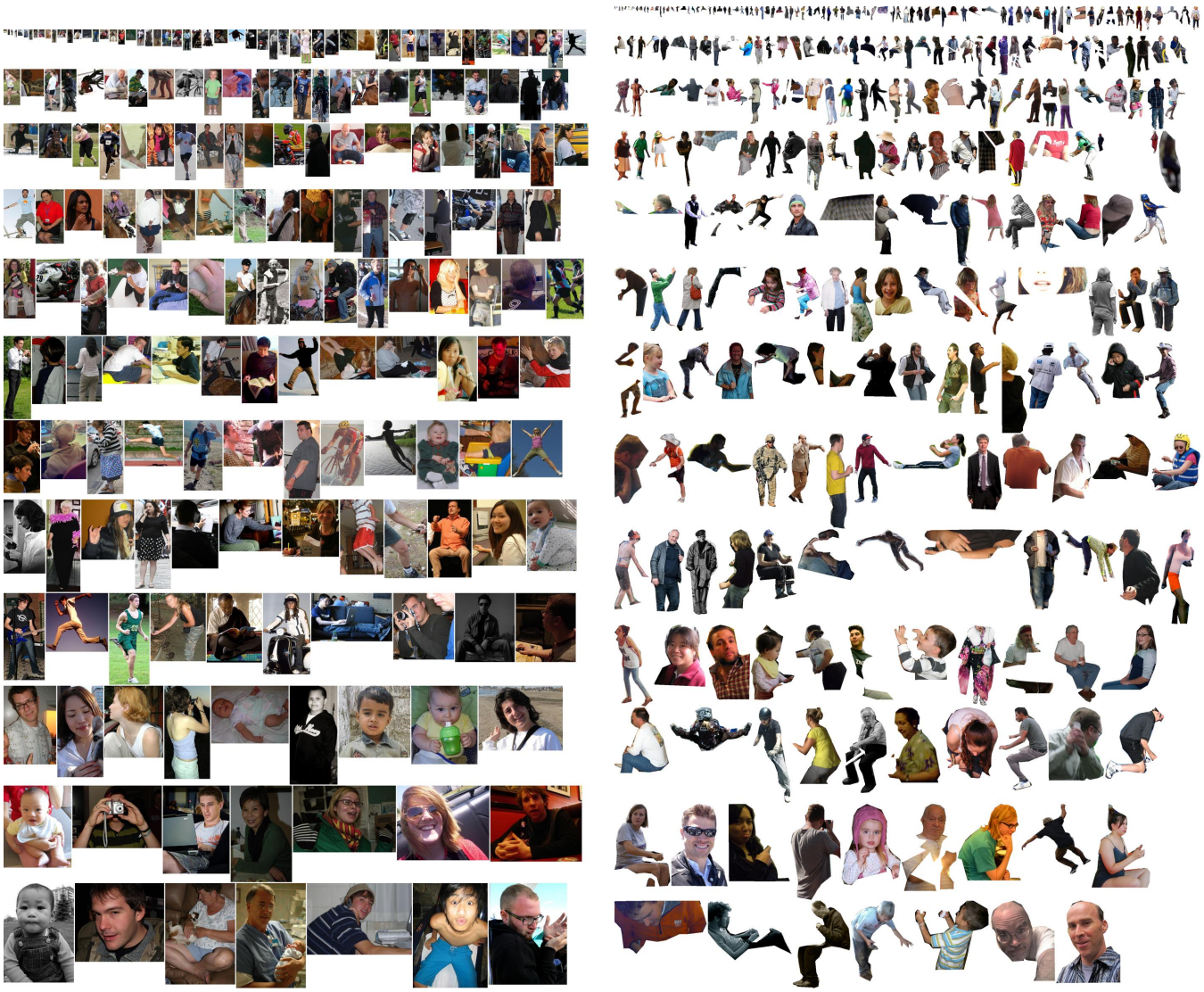
Fig. 11: Icons of 91 categories in the MS COCO dataset grouped by 11 super-categories. We use these icons in our annotation pipeline to help workers quickly reference the indicated object category.

(a) Category Labeling (b) Instance Spotting (c) Instance Segmentation



(d) Segmentation Verification (e) Crowd Labeling

Fig. 12: User interfaces for collecting instance annotations, see text for details.



(a) PASCAL VOC. (b) MS COCO.

Fig. 13: Random person instances from PASCAL VOC and MS COCO. At most one instance is sampled per image.

person	**bicycle**	**car**	**motorcycle**	**bird**	**cat**	**dog**	**horse**	**sheep**	**bottle**
chair	**couch**	**potted plant**	**tv**	**cow**	**airplane**	**hat***	license plate	**bed**	**laptop**
fridge	**microwave**	**sink**	**oven**	**toaster**	**bus**	**train**	**mirror***	**dining table**	**elephant**
banana	bread	**toilet**	**book**	**boat**	**plate***	**cell phone**	**mouse**	**remote**	**clock**
face	hand	**apple**	**keyboard**	**backpack**	steering wheel	**wine glass**	chicken	**zebra**	**shoe***
eye	mouth	**scissors**	**truck**	**traffic light**	**eyeglasses***	**cup**	**blender***	**hair drier**	wheel
street sign*	**umbrella**	**door***	**fire hydrant**	**bowl**	teapot	**fork**	**knife**	**spoon**	**bear**
headlights	**window***	**desk***	computer	**refrigerator**	**pizza**	squirrel	duck	**frisbee**	guitar
nose	**teddy bear**	**tie**	**stop sign**	**surfboard**	**sandwich**	pen/pencil	**kite**	**orange**	**toothbrush**
printer	pans	head	**sports ball**	**broccoli**	**suitcase**	**carrot**	chandelier	**parking meter**	fish
handbag	**hot dog**	stapler	basketball hoop	**donut**	**vase**	**baseball bat**	**baseball glove**	**giraffe**	jacket
skis	**snowboard**	table lamp	egg	door handle	power outlet	hair	tiger	table	coffee table
skateboard	helicopter	tomato	tree	bunny	pillow	**tennis racket**	**cake**	feet	**bench**
chopping board	washer	lion	monkey	**hair brush***	light switch	arms	legs	house	cheese
goat	magazine	key	picture frame	cupcake	fan (ceil/floor)	frogs	rabbit	owl	scarf
ears	home phone	pig	strawberries	pumpkin	van	kangaroo	rhinoceros	sailboat	deer
playing cards	towel	hyppo	can	dollar bill	doll	soup	meat	window	muffins
tire	necklace	tablet	corn	ladder	pineapple	candle	desktop	carpet	cookie
toy cars	bracelet	bat	balloon	gloves	milk	pants	wheelchair	building	bacon
box	platypus	pancake	cabinet	whale	dryer	torso	lizard	shirt	shorts
pasta	grapes	shark	swan	fingers	towel	side table	gate	beans	flip flops
moon	road/street	fountain	fax machine	bat	hot air balloon	cereal	seahorse	rocket	cabinets
basketball	telephone	movie (disc)	football	goose	long sleeve shirt	short sleeve shirt	raft	rooster	copier
radio	fences	goal net	toys	engine	soccer ball	field goal posts	socks	tennis net	seats
elbows	aardvark	dinosaur	unicycle	honey	legos	fly	roof	baseball	mat
ipad	iphone	hoop	hen	back	table cloth	soccer nets	turkey	pajamas	underpants
goldfish	robot	crusher	animal crackers	basketball court	horn	firefly	armpits	nectar	super hero costume
jetpack	robots								

TABLE 2: Candidate category list (272). **Bold**: selected categories (91). **Bold***: omitted categories in 2014 release (11).

(a) PASCAL VOC. (b) MS COCO.

Fig. 14: Random chair instances from PASCAL VOC and MS COCO. At most one instance is sampled per image.

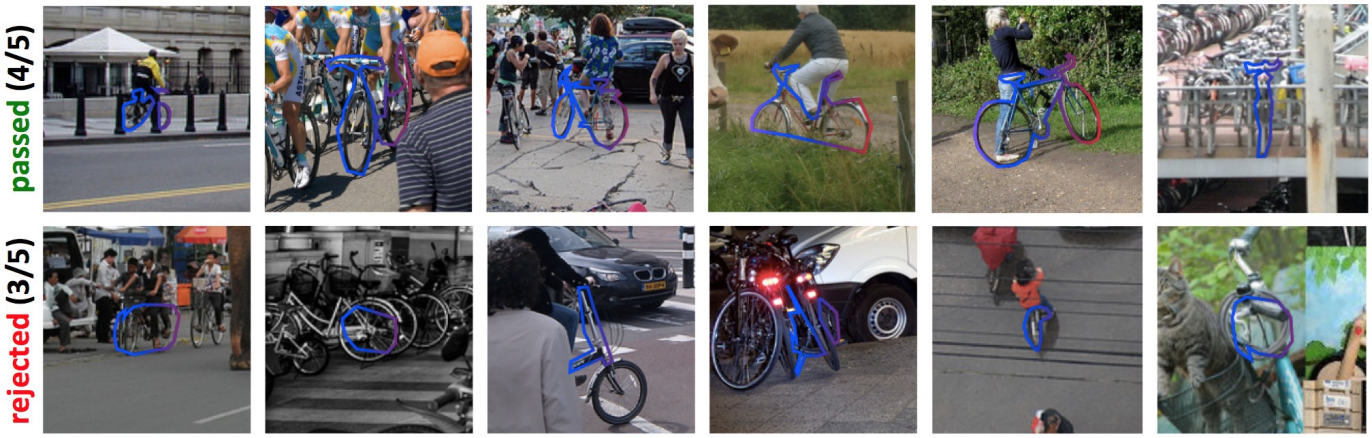


Fig. 15: Examples of borderline segmentations that passed (top) or were rejected (bottom) in the verification stage.

library	church	office	restaurant	kitchen	living room	bathroom	factory	campus	bedroom
child's room	dining room	auditorium	shop	home	hotel	classroom	cafeteria	hospital room	food court
street	park	beach	river	village	valley	market	harbor	yard	parking lot
lighthouse	railway	playground	swimming pool	forest	gas station	garden	farm	mountain	plaza

TABLE 3: Scene category list.