Crowdsourcing Annotations for Visual Object Detection

Hao Su, Jia Deng, Li Fei-Fei
Computer Science Department, Stanford University

Abstract

A large number of images with ground truth object bounding boxes are critical for learning object detectors, which is a fundamental task in compute vision. In this paper, we study strategies to crowd-source bounding box annotations. The core challenge of building such a system is to effectively control the data quality with minimal cost. Our key observation is that drawing a bounding box is significantly more difficult and time consuming than giving answers to multiple choice questions. Thus quality control through additional verification tasks is more cost effective than consensus based algorithms. In particular, we present a system that consists of three simple sub-tasks — a drawing task, a quality verification task and a coverage verification task. Experimental results demonstrate that our system is scalable, accurate, and cost-effective.

1 Introduction

Object detection is one of the fundamental tasks of visual recognition. Given an input image, an object detector outputs a bounding box wherever an object of interest exists. To learn a good detector, it is necessary to have a large number of training images with ground truth annotations in the form of bounding boxes, i.e. tight rectangles around the object of interest. Indeed, state of the art detection systems (Viola and Jones 2004; Felzenszwalb et al. 2010) have relied on accurate bounding box annotations. Although it is possible to use weaker supervision, e.g. binary labels of object presence, it substantially increases the difficulty of learning.

In this paper, we study strategies to crowd-source bounding box annotations. Our goal is to build a system that is fully automated, highly accurate, and cost-effective. Given a collection of images where the object of interest has been verified to exist, for each image the system collects a tight bounding box for every instance of the object. Specifically, we have the following two requirements.

- **Quality.** Each bounding box needs to be tight, i.e. the smallest among all bounding boxes that contain the object. This would greatly facilitate the learning algorithms for the object detector by giving better alignment of the object instances;

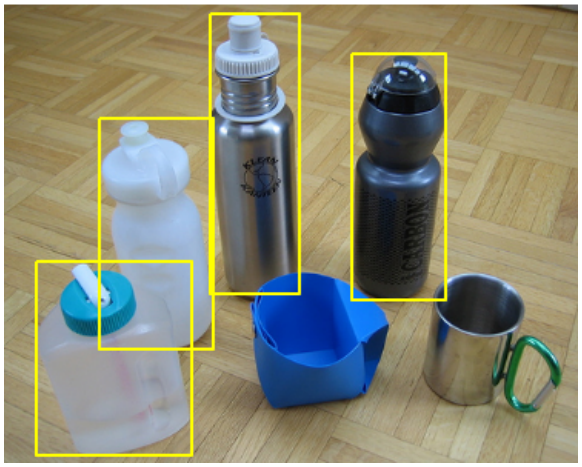


Figure 1: An example of bounding box annotations for the "bottle" category.

- **Coverage.** Every object instance needs to have a bounding box. This is important for detection because it tells the learning algorithms with certainty what is *not* the object.

Figure 1 shows examples of bounding box annotations that meet both the quality and coverage requirements.

The core challenge of building such a system is how to achieve both high quality and complete coverage in a cost-effective way, i.e. minimizing cost while guaranteeing quality. A basic quality control strategy is majority voting—collecting answers from multiple human subjects and taking the consensus. This approach has been successfully applied to image annotation tasks such as verifying the presence of objects or attributes (Deng et al. 2009; Sorokin and Forsyth 2008). However, drawing bounding box is significantly more time consuming than giving answers to multiple-choice questions about presence of objects. Thus instead of depending on the consensus of multiple workers, we propose to control quality by *having one worker draw the bounding box and another worker verify the quality of the bounding box.* Similarly, to guarantee coverage, we can ask a third worker to verify whether all object instances have bounding boxes. This leads to the following workflow that consists of three simple sub-tasks.

- **Drawing.** A worker draws one bounding box around one instance of the given image.
- **Quality verification.** A second worker verifies whether a bounding box is correctly drawn.
- **Coverage verification.** A third worker verifies whether all object instances have bounding boxes.

In this workflow, both verification tasks serve to control the quality of the drawing task. Meanwhile, since they both require only binary answers, their own quality can be controlled by well-proven techniques such as majority voting.

In the rest of the paper, we first show how to effectively design and implement the sub-tasks, including how to guarantee quality for the verification tasks themselves (Section 3). We then empirically evaluate the performance of our system and validate our design choices (Section 4). Experiments show that our system is fully automated, cost-effective, and produces high quality annotations. The system has been deployed to collect bounding boxes for more than 1 million images of the ImageNet dataset (Deng et al. 2009).

2 Related Work

The emergence of crowd-sourcing platforms, such as Amazon Mechanical Turk (AMT), has made it possible to collect image annotations in very large scale (Deng et al. 2009). The issue of how to effectively leverage the crowd has received increasing attention (Sorokin and Forsyth 2008; Whitehill et al. 2009; Welinder et al. 2010; P. Welinder 2010). However, most of the research in crowd-sourcing image annotations has been focusing on obtaining multi-choice answers such as those used in object categorization. There has been no in-depth study of crowd-sourcing approaches for collecting object bounding boxes. In (P. Welinder 2010), collecting bounding boxes is considered in a general framework. However, their approach essentially depends on the consensus of multiple workers. As we will demonstrate empirically, it is sub-optimal in terms of annotation cost.

Our approach of annotating bounding boxes is similar to the "grading" strategy mentioned as a general framework in (Sorokin and Forsyth 2008), but to our knowledge we are the first to study it in the context of bounding box annotations.

Another line of work studies how to incorporate computer vision techniques to optimize the human labeling. In (Vittayakorn and Hays 2011), human annotations can be approximately evaluated by machines through learning a scoring function based on visual cues. Active learning (Vijayanarasimhan and Grauman 2011) concerns determining which image to label next (as opposed to a random pick) in a way that better benefits the learning of visual models. Online learning techniques have also been explored to make the human labeling interactive (Branson, Perona, and Belongie 2011). These approaches are orthogonal to the problem we study here, as human annotation is still an indispensable component.

3 Approach

In this section, we describe our system in detail.

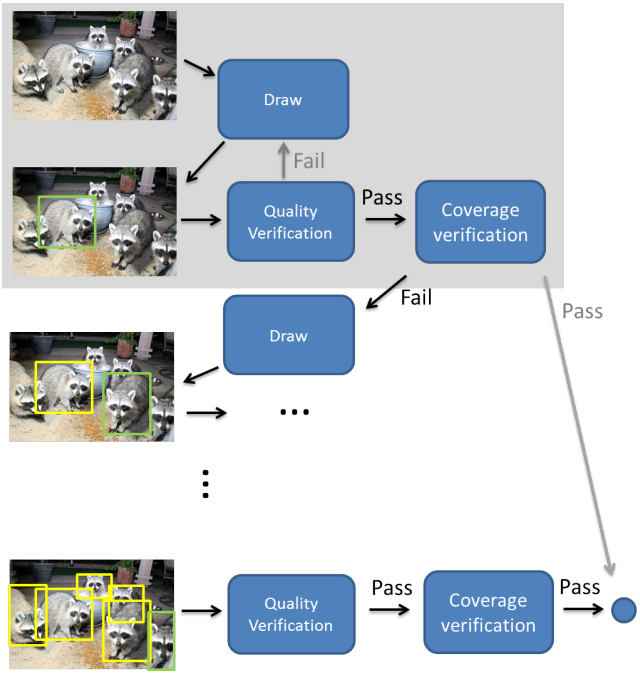


Figure 2: The work flow of our system. There are three sub-tasks, drawing, quality verification, and coverage verification.

3.1 Work Flow

The system starts with an image where the presence of the object has been verified. Take the "raccoon" category as an example (see Figure 2).

The drawing task asks the worker to draw a bounding box around *one* instance of raccoon. Once a bounding box is drawn, it is then passed to a quality verification task.

In the quality verification task, a second worker evaluates the quality of the newly drawn bounding box. Good bounding boxes are registered in the database. Bad bounding boxes are rejected and a new drawing task is generated.

The coverage verification task requests a third worker to check whether there are still instances of raccoon not covered by a bounding box. If everyone is covered, the annotation of the image is marked as complete. Otherwise, the system launches a new drawing task to solicit another bounding box over an uncovered raccoon.

The procedure repeats until every raccoon is covered by a bounding box.

It is worth noting that the sub-tasks are designed following two principles. First, the tasks are made as simple as possible. For example, instead of asking the worker to draw all bounding boxes on the same image, we ask the worker to draw only one. This reduces the complexity of the task. Second, each task has a fixed and predictable amount of work. For example, assuming that the input images are clean (object presence is correctly verified) and the coverage verification tasks give correct results, the amount of work of the drawing task is always that of providing exactly one bounding box.

3.2 Drawing Task

The drawing task consists of a batch of images. In each image it has been assured that there exists at least one object instance not covered by a bounding box. For each image, we ask the worker to draw one bounding box around one object instance that does not have a bounding box yet. Although it is an intuitive task, we find it important to make precise the requirements and make sure that the worker understands them. To this end, we mandate a training phase for all new workers.

Worker Training The training consists of reading a set of instructions and then passing a qualification test.

The instructions are composed by a set of rules:

Rule 1: Include all visible part and draw as tightly as possible.

Rule 2: If there are multiple instances, include only ONE (any one).

Rule 3: Do not draw on an instance that already has a bounding box. Draw on a new instance.

Rule 4: If you cannot find the required object, or every instance already has a bounding box, check the check box.

Each rule is illustrated with real examples. Figure 3 shows the instructions for Rule 1 to 3.

We then ask the worker to pass a qualification test that includes a small set of test images. These test images are chosen so that she cannot pass it without correctly understanding the rules. The worker receives instant feedback if she draws the bounding box incorrectly. For example, Figure 4 shows what happens when the bounding box is close but not exactly correct. Note that we set a rather high standard for getting tight bounding boxes. We provide three types of feedback messages targeting at common mistakes: 1) the bounding box is not sufficiently tight, 2) the object selected is not the solicited object, 3) the object selected already have a bounding box. We note that the instant feedbacks have effectively improved the learning speed of annotators.

Workers who have completed the training phase can then start to work on real images. Previously drawn bounding boxes are displayed in a different color. The worker clicks the mouse to select the upper-left corner and then drag the mouse to draw a rectangle over an uncovered object instance. She can further refine it by adjusting the four corners. The drawing interface also provides links to Wikipedia and Google such that the worker can look up the definition of the object (see Figure 5).

3.3 Quality Verification Task

In the quality verification task, a worker is given a batch of bounding boxes and is asked to examine the quality of each of them. We show only one bounding box in each image so that workers can be focused.

Worker Training Similar to the drawing task, training is required. A new worker is first shown instructions describing what a good bounding box means:

Rule 1: A good bounding box must include an instance of the required object.

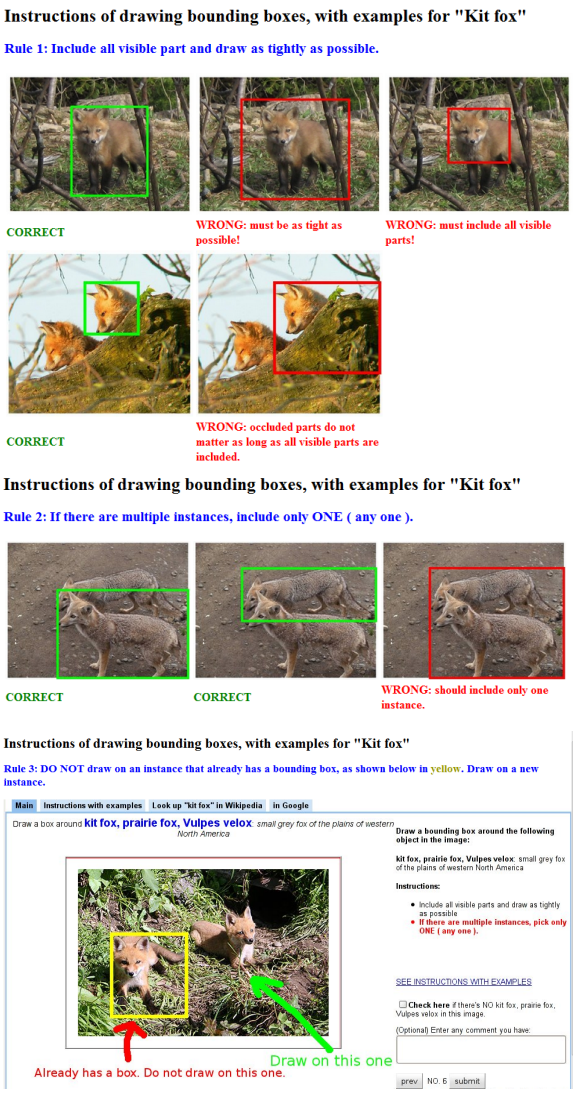


Figure 3: Instructions (Rule 1 to 3) for the drawing task.

Rule 2: A good bounding box must include all visible parts and be as tight as possible.

Rule 3: If there are multiple instances, a good bounding box must include only ONE (any one).

Next, the worker must pass a qualification test where she rates some test images that are known to have good and bad bounding boxes.

Quality Control Workers who successfully finish the training can start to verify the quality of bounding boxes from the drawing tasks (see Figure 6). However, the *quality* of these quality verification tasks also needs to be controlled. For instance, a spammer might rate every bounding box as bad or good without examining their qualities. To minimize cost, we adopt a strategy of embedding "gold standard".

For each task, a fraction of validation images that are known to have good or bad images are planted into the batch. A worker's submission is accepted if and only if the worker

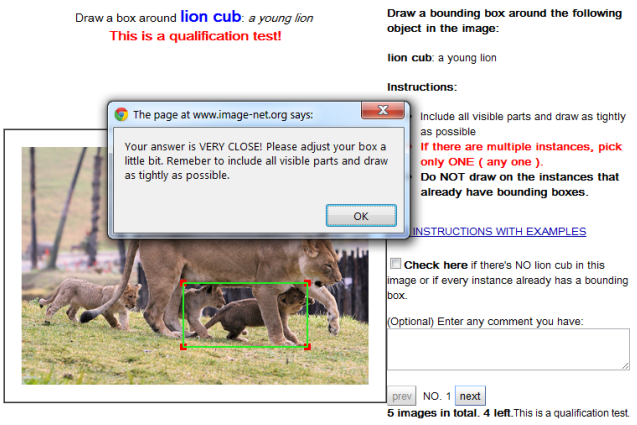


Figure 4: Qualification test for drawing task training. The worker is asked to draw a bounding box around a young lion. The worker draws the green rectangle, but it is not tight enough—the paw is outside the bounding box. The user is thus prompted by the system to refine the bounding box.

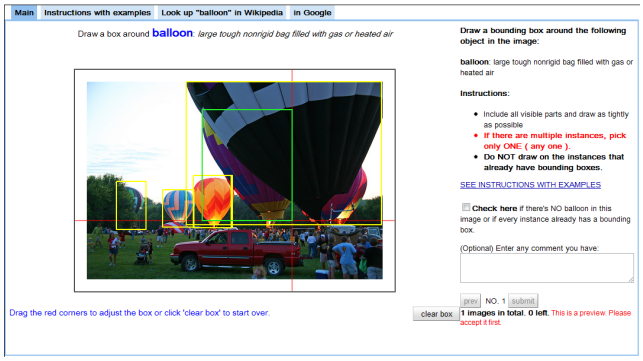


Figure 5: The interface of the drawing task. The worker clicks and drags the mouse to draw a new bounding box (the green bounding box). At the moment, the mouse is at the intersection of the horizontal line and the vertical line. The yellow boxes are existing bounding boxes.

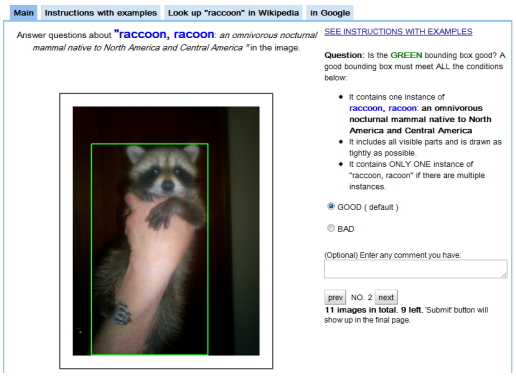
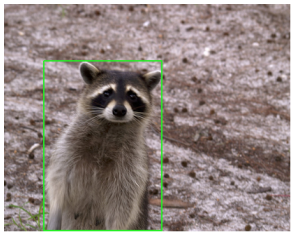


Figure 6: The interface of the quality verification task.

performs well on these validation images.

The validation images are used to prevent the two types of mistakes that a worker may make. The first type mistake is to rate a good bounding box as bad, while the second type is to rate a bad bounding box as good. To measure how frequently a worker makes the first (second) type mistake, we need validation images with good (bad) bounding boxes (see Figure 3.3). Since validation images with bad bounding boxes can be generated by perturbing good bounding boxes, the problem is reduced to obtaining bounding boxes that are assured to be good.



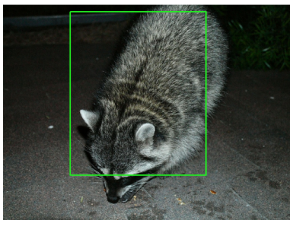


Figure 7: Left: validation image with a good bounding box. Right: validation image with a bad bounding box.

We collect the good validation bounding boxes via majority voting. More specifically, given a set of images containing a specific object, the system first samples a small subset and acquires their bounding boxes from the drawing tasks. Next, these bounding boxes are rated by multiple workers and those with strong consensus are selected as the "gold standard".

Note that the cost of obtaining these validation images is small, because only a small number of images are needed for each category and their annotations are collected only once. Other than those used for validation, each bounding box only needs to be rated by one worker who performs well on the validation bounding boxes.

3.4 Coverage Verification Task

The coverage verification task displays all bounding boxes collected so far for an image and asks a worker whether every instance of the object has a bounding box. Each task consists of a batch of images that contain the same object and the task is assigned to one annotator. Figure 8 shows the interface for the coverage verification task.

Similar to the drawing task and quality verification task, training is also required. It includes reading instructions with illustrations and passing a qualification test.

Quality Control We implement quality control in a way similar to the quality verification task. We need to create two types of validation images, one that are completely covered and one that are not. The first type can be generated by majority voting and the second by removing a subset of bounding boxes from the first type.

4 Experiments

We deploy our system on Amazon Mechanical Turk (AMT) and evaluate it in terms of quality and cost.

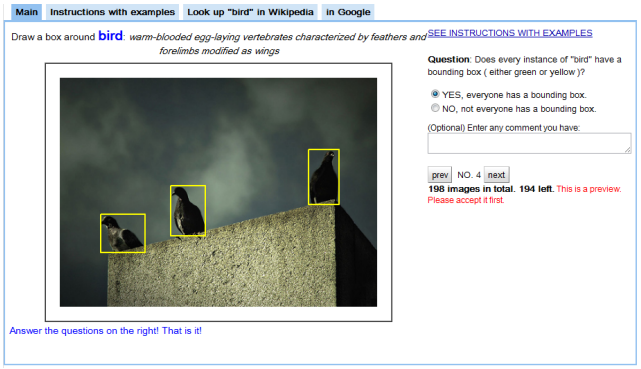


Figure 8: The interface of the coverage verification task

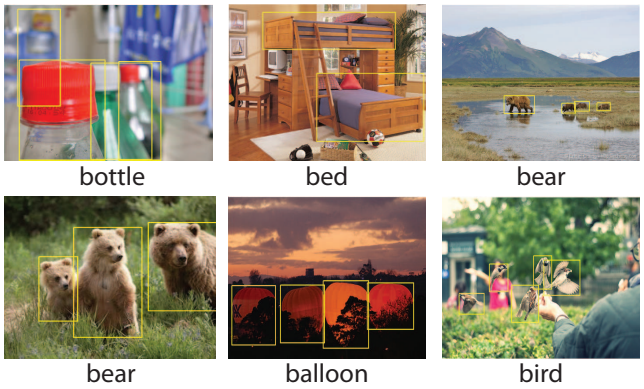


Figure 9: Examples of accurate bounding boxes produced by our system.

Figure 10: Examples of typical errors of the drawing task. Left: basketball hoop. The bounding box is not tight. Middle: bed. The bounding box does not include one leg of the bed. Right: bottle. The upper part of the bounding box is actually not part of the bottle.

4.1 Dataset

We evaluate our system using images from ImageNet(Deng et al. 2009), an image database with over 20,000 categories and over 14 million images. We select 10 categories: balloon, bear, bed, bench, beach, bird, bookshelf, basketball hoop, bottle, and people. A subset of 200 images are randomly sampled from each category. The presence of the objects in the images is guaranteed by ImageNet[1]

4.2 Overall Quality

We evaluate the overall performance of our system by manually inspecting its end results.

On the image level, our evaluation shows that 97.9% images are completely covered with bounding boxes. For the remaining 2.1%, some bounding boxes are missing. However, these are all difficult cases—the size is too small, the boundary is blurry, or there is strong shadow.

On the bounding box level, 99.2% of all bounding boxes are accurate (the bounding boxes are visibly tight). The remaining 0.8% are somewhat off. No bounding boxes are found to have less than 50% overlap with ground truth. Figure 9 shows examples of accurate bounding boxes and Figure 10 shows typical bounding boxes that are somewhat off.

Our evaluation demonstrates that our system produces highly accurate bounding boxes.

4.3 Overall Cost

In this experiment, we show that our system design is highly cost-effective. We measure cost by the amount of time that workers spend. Figure 11 plots the histogram of time cost per bounding box for different tasks among the workers. Table 1gives the means and medians.

Table 1 and Figure 11 shows that the drawing task takes more than twice as long to finish as a quality verification task or a coverage verification task. This difference is not surprising given that both verification tasks require only binary answers. There are two implications: 1) The drawing task costs twice or more than either of the verification tasks; 2) our system design is significantly more efficient than a naive majority-voting system—for any majority-voting approach, a minimum of two drawing tasks are needed to reach consensus and an additional coverage verification task is necessary, too.

How does the cost of our approach compare to consensus based ones? Based on Table 1, our system costs an average of 88.0 seconds of worker time for each bounding box whereas a consensus based approach would cost at least $50.8 \times 2 + 15.3 = 116.9$ seconds. In other words, the consensus based methods are at least 32.8% more expensive than ours. Note that the time measured in Table 1 includes the time for loading an image through the Internet, which typically takes 1 to 5 seconds. With the image loading time improved, our saving can be even more significant. In addition, this analysis is assuming that the market price is determined by the mean of the worker time. However, the histogram in Figure 11 shows that there are a significant minority of workers that take an excessively long time to finish a task (possibly due to switching to other tasks or taking breaks in the middle of a task). This makes the median time a better proxy of the market price, in which case the consensus based approaches would be even more costly (38.9% more expensive).

4.4 Analysis of Quality Control

In this section, we discuss the effect of our quality control for each task.

[1]According to (Deng et al. 2009), An average of 99.7% accuracy is achieved on over 10,000 classes.

Task Name	Time per b.box	
	Median	Mean
Drawing	25.5s	50.8s
Quality Verification	9.0s	21.9s
Coverage Verification	7.8s	15.3s
Total	42.4s	88.0s

Table 1: Time per bounding box spent by an worker for each type of task. The results are estimated from 106 drawing tasks, 237 quality verification tasks and 169 coverage verification tasks.

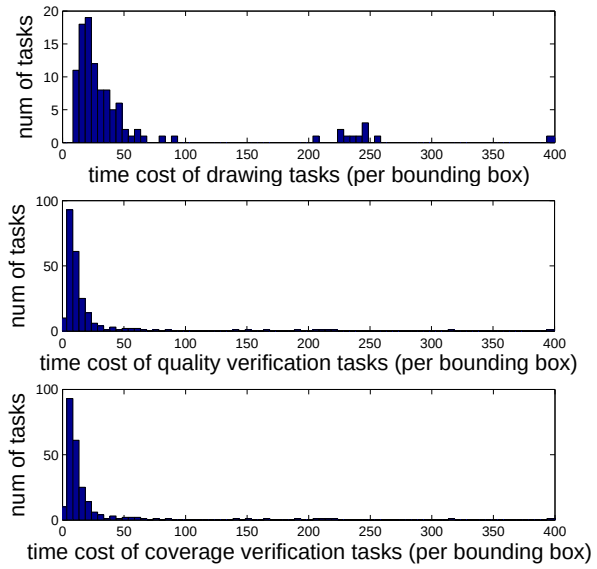


Figure 11: Histogram of time cost for different tasks among the workers (per bounding box).

Quality control of the drawing task The quality of the drawing task is controlled by the quality verification task. The workers draw a total of 6861 bounding boxes on the 2000 images in our dataset. Out of the 6861 bounding boxes, 4267 are accepted in the quality verification task. *Thus the acceptance ratio of drawing tasks is 62.2%*. Examples of typical errors are shown in Figure 10.

Quality control of the quality verification tasks The quality of the quality verification task is controlled through "gold standard" constructed by "majority voting". As discussed in Section 3.3, the submissions of a worker is evaluated based on her performance on the validation images. In practice, we plant both validation images with good bounding boxes and those with bad ones. In particular, a good bounding box is generated by majority-voting if at least 3 workers vote it as good with no objection and a bad bounding box by perturbing known good ones. A submission of the quality verification task is accepted by our system if the worker does well on the validation bounding boxes. Out of 629 quality verification tasks submitted to AMT, 566 of them are accepted, which give an acceptance ratio of 89.9%. Further inspection shows that the typical errors are made by spammers who mark the bounding boxes as either all good or all bad.

Quality control of the coverage verification tasks Similar to the quality verification task, the quality of the coverage verification task is controlled by evaluating the worker's performance on validation images. Our data shows that out of 427 coverage verification tasks submitted to AMT, 406 of them are accepted. This gives an acceptance ratio of 95.0%, much higher than that of the drawing tasks. The typical errors are made by spammers who mark every image as complete or as incomplete.

Note that the acceptance ratios of the both qualification tasks (89.9% and 95%) are much higher than that of the drawing tasks (62.2%). This demonstrates that the drawing task is not only more time consuming but also much more *difficult* than the verification tasks.

Effectiveness of Worker Training Workers are trained when they work on a task for the first time. The training ensures that workers understand the annotation and verification instructions. In this section, we show that the training in the drawing task improves their work quality.

As a comparison experiment, we remove the worker training phase in drawing tasks and run the simplified system on the same set of data. Similar to Section 4.4, we measure the acceptance ratio of the drawing task (the percentage of bounding boxes that pass the quality verification task)[2]. Table 2 compares the acceptance ratios with and without the training phase. It shows that 4.2% more bounding boxes pass the quality verification, which is a significant improvement.

Our results on the quality control components demonstrate that 1) quality control is critical for our system to produce high quality data and 2) workers on AMT do better on simpler tasks such as answering binary questions in quality/coverage verification tasks. Both findings support our design of the sub-tasks.

	Without Training	With Training
Acceptance Ratio	58.0%	62.2%

Table 2: Effect of worker training in the drawing task.

5 Conclusion

In this paper, we have presented a system that collects bounding box annotations through crowd-sourcing. The work flow of the system consists of three sub-tasks, each with carefully designed quality control mechanisms. Experiments demonstrate that our system produces high quality data in a cost effective manner.

References

Branson, S.; Perona, P.; and Belongie, S. 2011. Strong supervision from weak annotation: Interactive training of deformable part models. In *ICCV*, 1832–1839.

[2]We excluded the results submitted by trained workers.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

Felzenszwalb, P. F.; Girshick, R. B.; McAllester, D. A.; and Ramanan, D. 2010. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* 32(9):1627–1645.

P. Welinder, P. P. 2010. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *CVPR*.

Sorokin, A., and Forsyth, D. 2008. Utility data annotation with amazon mechanical turk. *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* 51(c):1–8.

Vijayanarasimhan, S., and Grauman, K. 2011. Large-scale live active learning: Training object detectors with crawled data and crowds. In *CVPR*, 1449–1456.

Viola, P. A., and Jones, M. J. 2004. Robust real-time face detection. *International Journal of Computer Vision* 57(2):137–154.

Vittayakorn, S., and Hays, J. 2011. Quality assessment for crownsourced object annotations. In *Proceeding of British Mchine Vision Conference (BMVC)*.

Welinder, P.; Branson, S.; Belongie, S.; and Perona, P. 2010. The multidimensional wisdom of crowds. In *NIPS*, 2424–2432.

Whitehill, J.; Ruvolo, P.; Wu, T.; Bergsma, J.; and Movellan, J. R. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, 2035–2043.