Large Scale Image Annotations on Amazon Mechanical Turk

Subhransu Maji

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2011-79
http://www.eecs.berkeley.edu/Pubs/TechRpts/2011/EECS-2011-79.html

July 1, 2011

Large Scale Image Annotations on Amazon Mechanical Turk

Subhransu Maji
Computer Science Division
University of California, Berkeley
smaji@cs.berkeley.edu

Abstract

We describe our experience with collecting roughly $250,000$ image annotations on Amazon Mechanical Turk (AMT). The annotations we collected range from location of keypoints and figure ground masks of various object categories, 3D pose estimates of head and torsos of people in images and attributes like gender, race, type of hair, etc. We describe the setup and strategies we adopted to automatically approve and reject the annotations, which becomes important for large scale annotations. These annotations were used to train algorithms for detection, segmentation, pose estimation, action recognition and attribute recognition of people in images.

1 Introduction

Collecting annotations in a cost effective manner has become possible due to emergence of efficient marketplaces like AMT (Amazon Mechanical Turk) [1]. There are three ingredients for constructing a HIT (Human Intelligence Task) which "workers" on AMT can complete :

1. **User Interface.** This is the front-end which enables the user to do the task inside their web browsers. Some of our tasks required users to draw the boundaries or mark the locations of various keypoints of objects. All our GUIs (Graphical User Interfaces) were written in Java/JavaScript + HTML.

2. **Instructions.** Contains the task description, with examples of completed task as well as GUI usage instructions.

3. **Verification.** A method to approve/reject the HITs. This becomes important for large scale annotations as this step also has to be done automatically. One can have a task done by multiple workers followed by outlier rejection or a secondary HIT to verify the results to automatically select the right answers. We adopt the first approach for all our tasks.

An interesting aspect of collecting annotations on AMT is that we can measure the inherent hardness of these tasks. Many of these tasks don't require specific training and human performance of even a casual annotator is quite good. The agreement between

various workers on a given problem can give a sense of the hardness of the task and provides an upperbound on the performance one might expect from an automatic system. In the 3D pose estimation problem we see that the humans are not perfect with an average error of 6° across views.

We describe the three ingredients, i.e. the interface, instructions and verification method for the each of the tasks we set up on AMT in the next few sections.

2 Figure-ground Masks of Objects

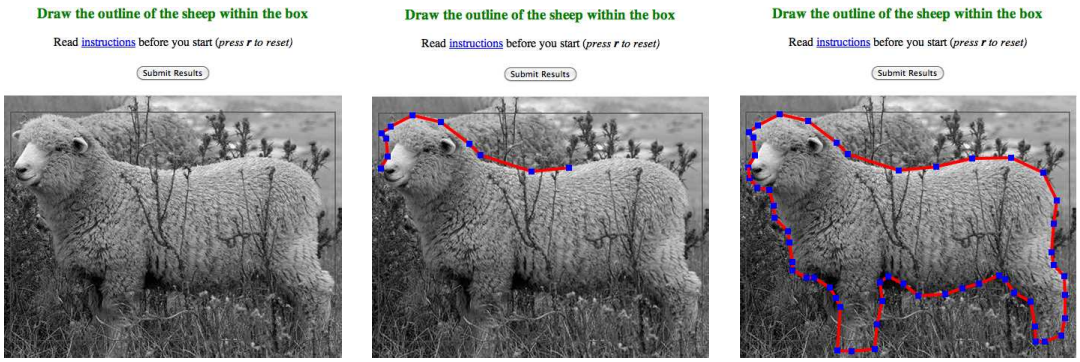
We wanted to collect figure ground masks for all the object categories. We focus on the categories in the PASCAL VOC dataset [2]. The PASCAL VOC 2010 dataset has $23,374$ objects in the training set from 20 categories. The statistics of the dataset are show in the Table 2.

Interface & Instructions. Our interface was as simple polygon outline tool which allows the user to draw a closed polygon and then move the vertices around to adjust the polygon. The advantage of this interface is that it is quite simple and intuitive to use. On the other hand, it only allows the user to draw one closed polygon which does not work well for objects with holes. An alternate interface was one which allows the user to paint the pixels belonging to the figure. This interface is too time consuming if done at the pixel level and too inaccurate on the boundaries if done at a superpixel (or a coarser quantization) level.

Figure 1 shows our interface for marking the outer boundaries of the objects. We provide instructions and sample segmentations to describe the task to the user. Below that is area to display the image to be segmented. To avoid confusion when there are multiple overlapping objects in the image, we draw a bounding box to indicate which object we are interested in.

The interface is written in Javascript + HTML5 entirely. It uses the "canvas" tag [3] which is currently supported in the latest Internet Explorer, Firefox and Safari browsers. We did not have any users complaining that the interface was not working properly for them. This switch was partly motivated by the difficulties we had in porting our keypoint labeling tool (next section) written in JAVA to various browsers. At the time of writing the "canvas" tag was only partly supported on IE, in particular they had no support for displaying text. We would like to port the keypoint labeling tool to Javascript + HTML5 in the future once text is supported by them.

Verification. We collect 5 independent annotations per object. For approving the HITs automatically we compute the pairwise overlap between the masks of an object and find the one which overlaps the maximum with everyone else. We consider all masks which overlap with this mask greater than a threshold. The threshold is chosen manually based on how flexible the object category is. For example for rigid objects like bottles and tv-monitors the we choose a threshold of 0.75 while for less rigid objects like cats and dogs we choose a lower threshold of 0.65 or even lower. In general the



(a) (b) (c)

Figure 1: The user interface for annotating the outer boundary of the objects. (a) The user sees this inside the Amazon Mechanical Turk environment. (b) Partial annotation by the user. (c) The user closes the polygon and edits the boundary if needed and clicks the submit button.

quality of segmentations submitted by the users are pretty good and only about 10% of the submitted hits were rejected. Figure 2 shows the distribution of submit times for the aeroplane and cat categories. Figure 3 shows some of the submitted results by the workers. Figure 4 shows some outliers which are rejected automatically by our algorithm.

3 Keypoint Annotation of Objects

Our goal was to mark the locations of various joints in images of objects. The first challenge is deciding which keypoints to use. This is fairly straightforward for other animal categories where one can base them on anatomical parts, but becomes more complicated for categories, such as a chair, a boat and an airplane, whose examples have large structural variations. There are chairs with four legs or one stem and a wide base. Some chairs have armrests, and others don't. Military airplanes look very different from commercial ones, and sail boats have little in common with cruise ships. We decided to split the categories into a few common subcategories and provide separate keypoints for each subcategory. This allows us to train separate poselets for the pointed front of a military airplane, the round tip of a commercial airliner and the propeller blades of a propeller plane.

The second challenge is that some categories do not have a principal orientation, which makes it difficult to assign keypoints in the reference frame of the object. For example, it is clear what the front left leg is in the case of a horse, but what is the front left leg of a table? Other categories have round parts and thus have no extrema points, such as the base of a bottle or a potted plant. Our solution in these cases is to introduce view-dependent keypoints. For example, we have a keypoint for the bottom left corner of a bottle, and we define the front left leg of a table based on the current camera view.

Category	Number of Objects	Reward (cents)	Submit Time (seconds)
Aeroplane	738	2	77/59
Bicycle	614	2	87/69
Bird	971	1	72/57
Boat	687	1	47/36
Bottle	1014	1	47/36
Bus	498	1	55/41
Car	1774	1	55/42
Cat	1132	1	70/57
Chair	1890	1	60/44
Cow	464	2	70/58
Diningtable	468	1	50/36
Dog	1416	1	70/57
Horse	621	2	95/77
Motorbike	611	2	80/65
Person	7296	1	55/43
Pottedplant	821	1	65/50
Sheep	701	2	67/50
Sofa	451	1	65/51
Train	524	1	59/46
Tvmonitor	683	1	32/25
Total	23374		

Table 1: Statistics of PASCAL VOC 2010 trainval set. For each image we collected 5 independent annotations. We paid them either 1 or 2 cents based on the how complex we thought the boundaries of the class were as shown in the Reward column. This is more or less also reflected in the mean/median submit time of the HITs shown in the last column.

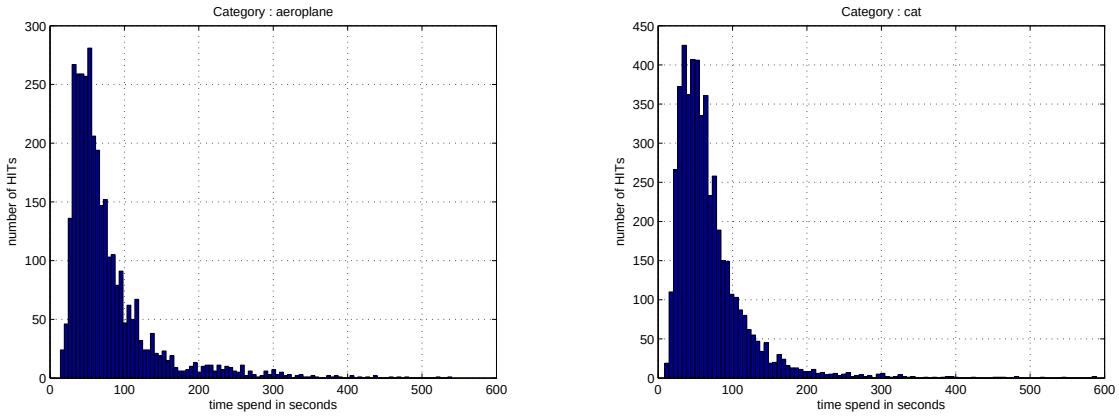


Figure 2: Histogram of submit times for the aeroplane (left) and cat (right) categories.

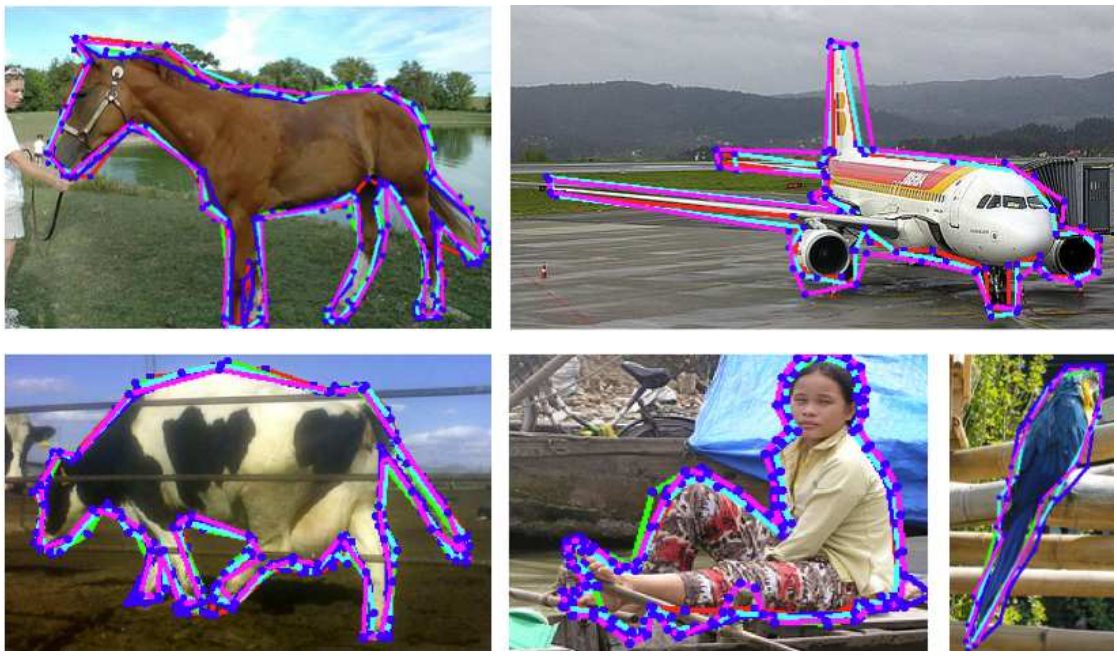


Figure 3: Example results submitted by workers.

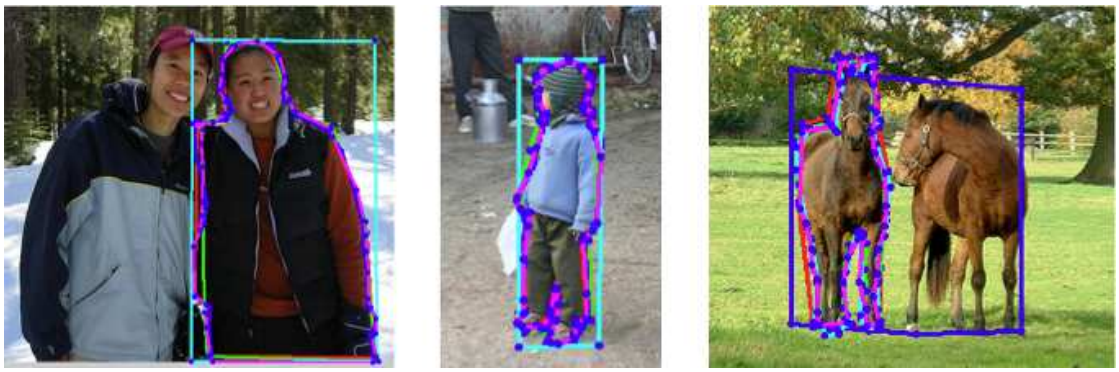


Figure 4: Outliers in the submitted boundaries by the workers on several images. These are automatically rejected as they do not have high overlap with the best answer.

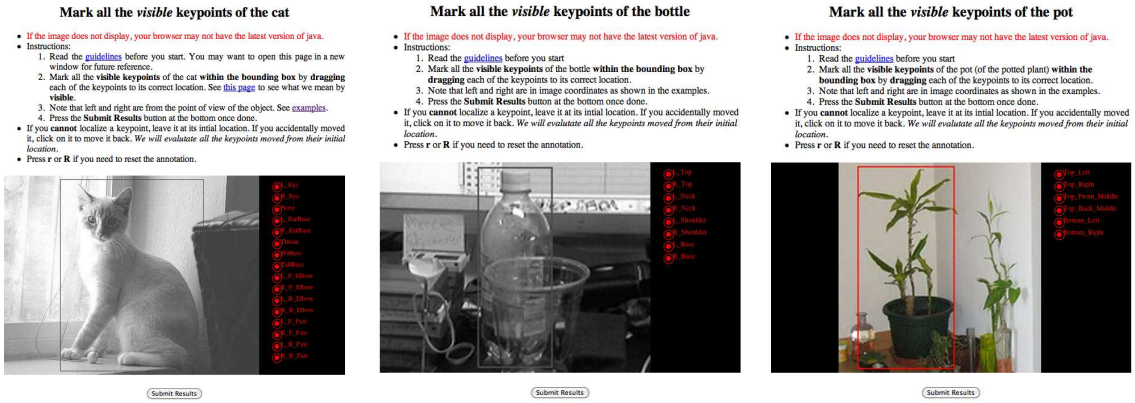
Class	# Keypoints	# Subcategories
Aeroplane	16	3
Bicycle	11	1
Bird	12	2
Boat	11	2
Bottle	8	1
Bus	8	1
Car	14	1
Cat	16	1
Chair	10	1
Cow	16	1
Dining table	8	1
Dog	16	1
Horse	19	1
Motorbike	10	1
Person	20	1
Potted plant	6	1
Sheep	16	1
Sofa	12	1
Train	7	1
TV monitor	8	1

Table 2: Class-specific variations in the keypoint annotations. **#Keypoints** is the number of keypoints and **#Subcategories** is the number of subcategories.

The number of keypoints and the subcategories are shown in Table 2.

Interface & Instructions. Figure 5 shows the interface we have for annotating the keypoints. Each user is shown an image within a bounding box and a list of keypoints. The user drags and drops these to their locations in the image. The user is instructed not to mark the points which are not visible due to occlusion, truncation etc. If a user accidentally moves a point then he/she can click on it to move it back to its initial position. Once the user is done he/she can press submit.

Verification. Each object was annotated by 5 independent users. We assume that a keypoint is visible if at least 2 annotators have marked its location. To determine the location of each keypoint, we find the closest pair of annotations and average all the annotations which are within a certain radius of them. We also get an estimate of the variance of keypoints and manually fix points which have large variance.



(a)*cat* (b)*bottle* (c)*pottedplant*

Figure 5: The user interface for annotating the keypoints of the objects. An image inside the bounding box is shown along with the list of keypoints on the right. The user can move the points and place them on their locations in the image or leave them at untouched if the point is not visible.

4 3D Pose of Humans

We construct a dataset of people annotated with the 3D pose of the head and torso. One can try to estimate the 3D pose from the 2D keypoints but this itself is nontrivial because of occlusions, truncations and variations of head/torso sizes across people. We asked users to estimate the rotations around X, Y and Z directly. Since we want to study pose estimation in a challenging setting, we collect images of people from the *validation* subset of Pascal VOC 2010 dataset but remove the person annotations which are marked difficult or truncated.

Interface & Instructions. The interface (Figure 6(a)) shows an image on the left and two gauge figures corresponding to the head and the torso on the right. They are asked to adjust the pose of the gauge figures corresponding to match the 3D pose of the shown person in the image.

Verification Each person was annotated by 4 different people for outlier rejection and estimate of variance. We manually verified the results and threw away the images where there was high disagreement between the annotators. These typically turned out to be images of low resolution or severely occluded ones. Our dataset has very few examples where the rotation along X and Z axes is high which is natural in everyday pictures of people. We collected a total of 1620 people annotations.

Figure 6(b) shows the human error in estimating the yaw across views of the head and torso. This is measured as the average of standard deviation of the annotations on a single image in the view range. The error is small when the person is facing front, back,

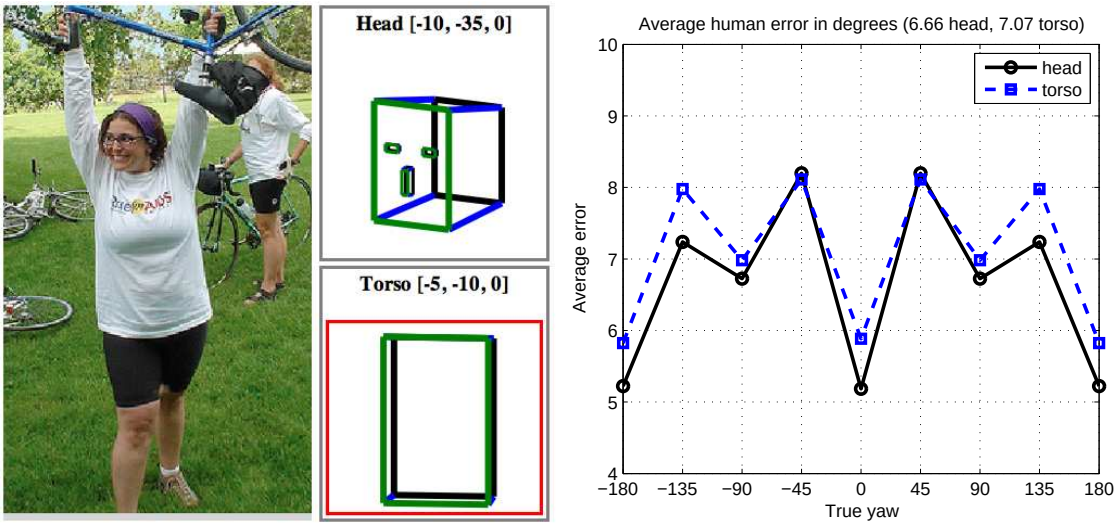


Figure 6: (a) The user interface for 3D pose annotation on AMT (b) Human error rate across views.

left or right whereas it is high when the person in facing somewhere in between. Overall the annotators are fairly consistent with one another with a median error of 6.66° for the head and 7.07° for the torso across views.

5 Attributes of People

About 9000 images of people were taken from the H3D and PASCAL VOC 2010 dataset for which we collect several attributes.

Interface & Instructions. Each person is shown a set of images and asked to mark the attribute for each image (Figure 7). If the attribute is not clear because of occlusion, truncation, etc, the user was asked not to mark any option. As instructions the users were also shown examples for each attribute kind as shown in Figure 8. Table 3 shows the list of attributes we annotated. Instead of showing the entire person we only show the region of interest, for example, upper bodies for hairtype and gender and lower bodies for lower-clothes. We are able to do this using the keypoint annotations we obtained earlier on the same dataset. This makes is much more easier for the users to annotate them and there were many more images which were marked with some attribute compared to an earlier run we did using the entire person shown as the same sized images. We typically paid the workers 1 cent for marking 16 attributes per HIT.

Verification. We collected labels for all attributes on all annotations by five independent annotators. A label was considered as ground truth if at least 4 of the 5 annotators

Figure 7: User interface for marking the "lower-clothes" attribute

.

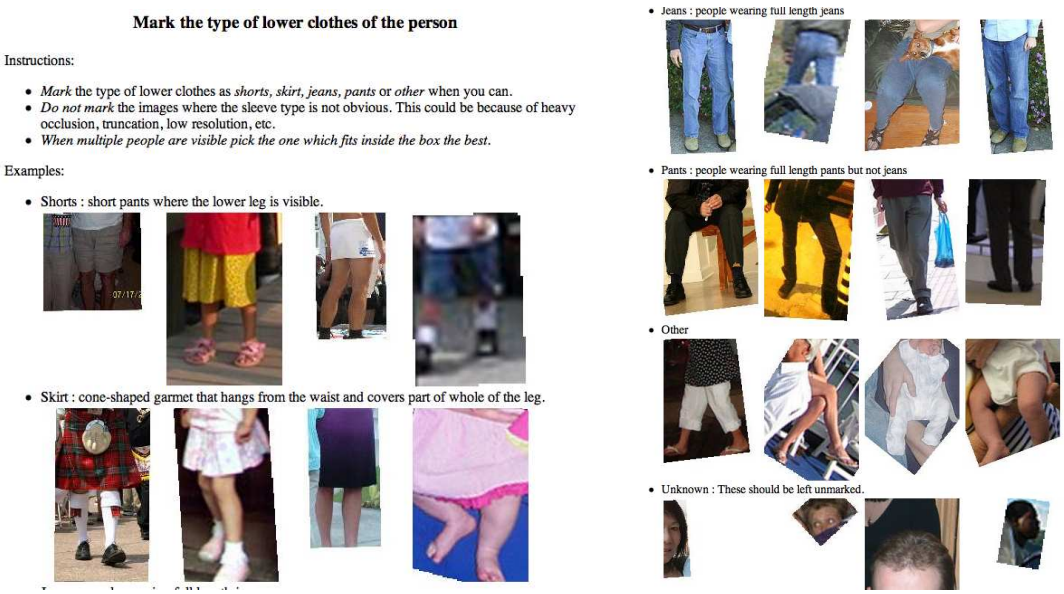


Figure 8: Instructions for marking the "lower-clothes" attribute. Examples of each choice like jeans, shorts etc are shown. The user is also shown examples which may be left unmarked.

Attribute	Choices
gender	male, female
race	white, black, asian, indian
age	baby(0-2), child(2-10), adult, old(65+)
hair-type	long, short, nohair
glasses	regular, sunglasses, no-glasses
shoes	barefoot, sneaker/shoe, sandal
sleeve-type	short-sleeve, long-sleeve, no-sleeve
upper-clothes	t-shirt, shirt, noclothes, bikini, tanktop, bikerwear, other
headgear	cap/hat, full-helmet, half-helmet, other, none
lower-clothes	shorts, skirt, jeans, pants, other
hair-color	black, blonde, white, no-hair, other

Table 3: List of attributes we annotated on Amazon Mechanical Turk.

agreed on the value of the label. We discarded 501 annotations in which less than two attributes were specified as ground truths which left us with 8035 images. We paid the workers who got at least half the marked annotations right.

References

[1] Amazon Mechanical Turk, http://www.mturk.com

[2] The PASCAL Visual Object Classes (VOC) Challenge Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. and Zisserman, A. *International Journal of Computer Vision, 88(2), 303-338, 2010.* http://pascallin.ecs.soton.ac.uk/challenges/VOC/

[3] The *canvas* element. http://www.w3.org/TR/html5/the-canvas-element.html#the-canvas-element