Generation of Ground Truth for Object Detection While Playing an Online Game: Productive Gaming or Recreational Working?

Isaak Kavasidis, Concetto Spampinato, Daniela Giordano
University of Catania
V.le A. Doria 6, Catania, Italy
{kavasidis,cspampin,dgiordan}@dieei.unict.it

Abstract

In this paper we present a flash game that aims at generating easily ground truth for testing object detection algorithms. Flash the Fish is an online game where the user is shown videos from underwater environments and has to take photos of fish by clicking on them. The initial ground truth is provided by object detection algorithms and, subsequent, cluster analysis and user evaluation techniques, allow for the generation of ground truth based on the weighted combination of these "photos". Evaluation of the platform and comparison of the obtained results against a hand drawn ground truth confirmed that reliable ground truth generation is not necessarily a cumbersome task both in terms of effort and time needed.

1. Introduction

Testing the performance of image processing algorithms is a necessary evil. Necessary, because the development and fine-tuning of image processing algorithms is heavily influenced by the results of intermediate performance evaluations. These evaluations are usually made against annotated datasets, better known as ground truth. Evil, because gathering ground truth data is not a trivial task. In fact, the process of generating high quality ground truth presents several limitations mainly due to the human effort and working time needed.

There exist tools that integrate semi-automatic methods aiming to alleviate the burden that comes with image and video annotation. These methods require minimal user intervention and for this reason they offer a more user-friendly experience in gathering object information, such as bounding boxes, contours, recognition classes and associations to other appearances of the same object in previous or following frames of the video.

Nevertheless, the manual annotation of these data, even when such tools are available, still requires a lot of concentration especially when the quality of the video is too low or in the presence of crowded scenes, making the identification of objects of interest very difficult. In all cases, both the identification and the subsequent annotation of the objects are time consuming, tedious and error-prone tasks, since it requires the user to be focused for the total duration of the process.

To overcome all these difficulties, in this paper we propose an online game, named *Flash the Fish*, for generating large-scale object detection ground truth for underwater video segments. The idea behind the game is to engage users to play a funny game by simply clicking on fish through the game levels. By using this mechanism, no *a priori* knowledge is required for the users who must only "take photos" of fish, providing an increasing dataset of annotations which can be used for detection algorithm evaluation. Moreover, since the accuracy of the obtained ground truth is not always high enough, a cluster analysis module is used to offer more reliable results.

The main scientific contributions are to 1) show how playing a game and integrating crowdsourcing and user quality control methods constitutes a valid solution for easily creating reliable image and video annotations, and 2) provide an experimental tool in order to demonstrate that such approach accomplishes its purpose.

The remainder of the paper is organized as follows: Section 2 describes existing tools for ground truth generation, together with their strengths and limitations. Section 3, instead, describes the *Flash the Fish* game and its advantages over classical video annotation applications, while, Section 4 discusses the ability of the game to generate accurate annotations by assessing its performance and comparing it to hand-drawn ground truth data. Finally, concluding remarks and future developments are given.

2. Related Works

The majority of the existing applications for ground truth generation are "ad-hoc" tools created by isolated

978-0-7695-4990-3/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPRW.2013.105

research groups, and as such, they are designed to fulfill specific requirements. Examples of such applications include ViPER-GT [6], GTVT [2], GTTool [10], ODViS [9], which, however, lack any data sharing capabilities and they cannot be used for generating large scale ground truth datasets. Moreover, the majority of these use incompatible file formats, which does not permit integration of the produced annotations. All these needs combined with the rapid growth of the Internet have favored in the last years the expansion of web-based collaborative tools, which take advantage of the efforts of large groups of people to collect reliable ground truths. Additionally in [16] the authors demonstrated the utility of "crowdsourcing" to non-expert humans, the task of collecting large annotated datasets. An illustrious example is LabelMe [15], a web-based and crowdsourcing-enabled platform designed to collect user annotations in still images. However, the main disadvantage of LabelMe is the lack of intelligent mechanisms for quality control and integration of user annotations and the lack of automatic tools that aim to make the annotation process less cumbersome. Moreover, LabelMe is thought specifically for still images, although a video based version has been proposed [18] that, however, is not as successful and flexible as the image based version.

Many of these shortcomings are dealt with in PerLa [11], another web-based platform for creating cooperatively object detection, tracking and recognition ground truth for lengthy videos which also integrates crowdsourcing methods for annotation integration.

Nevertheless, these applications do not respond at one important need: generate cheap, mid to high quality, annotations in the least amount of time possible. Several approaches have tried to deal with this problem, by generating ground truth data automatically, [7, 4], but the results are unreliable and of low quality, leaving crowdsourcing-based methods as the only viable alternative.

For crowdsourcing to be effective two conditions must be satisfied: workers' motivation and quality control. For the former, the most immediate and attractive way, is paying the workers for their work [14, 5]. Another valid solution for workers motivation is personal amusement [17], such as in the case of the ESP, Peekaboom and KissKissBan games [1, 8] which exploit players' agreement (random pairing two players and let them guess each other's labels) to collect ground truth data.

The latter condition, the quality control one, has been tackled with different strategies that can be summarized [13] as: Task Redundancy (ask multiple users to annotate the same data), User Reputation and Ground Truth Seeding (i.e. coupling ground truth with test data).

When both of these conditions are satisfied, large scale datasets can be built, but the process might be very expensive and the results often contain low quality annotations since workers (even if payed) are not as motivated as researchers.

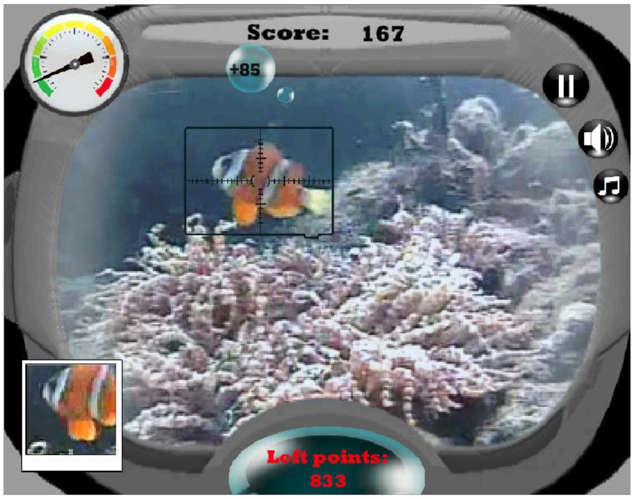

Figure 1. The game's interface. On the top left, the manometer shows the oxygen remaining (time) before the level ends. On top, the acquired points and on the top right three button controls to pause the game, mute the music and the sound effects, respectively, can be seen. On bottom left, the last took photo is shown and on bottom the points needed to advance to the next level are shown. Finally, the central area shows the video and the camera's shutter, which is centered on the mouse's pointer.

3. On-line Game for Ground Truth Generation

Flash the Fish exploits the amusement strategy to generate large scale object detection ground truth.

Playing with the game is simple: the user is presented a segment of an underwater video and she has to take photos of the fish, by clicking on them (Fig. 1) gaining as many points as possible. The user needs to gather a certain score to advance to the next game levels. Each "photo" contributes in estimating the presence or absence of fish at the corresponding point in the video.

The game consists of 7 different levels of progressively increasing difficulty. Every time a game session starts, a list with 7 random selected video segments, taken from our repository that contains more than 600.000 10-minute videos, is generated. The first level serves the role of assessing the skills of the player (see next section) and has an initial frame rate of 5 FPS and the time available is 35 seconds. At each successive level the frame rate of the video segment is increased by one, while the time available is reduced by 2 seconds, to a maximum of 11 FPS and a minimum of 23 seconds at the seventh and last level. The game can be found at `http://f4k-db.ing.unict.it/`.

In order to make the game more appealing, we adopted a scoring system that rewards users according to the quality of their annotations. In other words, the more precise the user is, the more points she earns and climbs up the final classification. Of course, in order to be able to assign scores, it is necessary that each video segment comes with a reference ground truth. If, for the specific video, there exists a hand-made ground truth, it will be used. Otherwise, if the video is not a new one (i.e. several players have already played it, meaning that several annotations exist), the reference ground truth is given by the combination of all the existing annotations (see Section 3.2). If, instead, the video is a new one (i.e. no one has played a session with this video yet) then the detection algorithm's [3] output is used as reference ground truth.

Having a reference ground truth, it is possible to compare the annotations provided by the users against it. For each object in the reference ground truth a 2D Gaussian Distribution is placed, centered on the object's bounding box center. If a player clicks on this point, she gains the maximum score bonus she can get, while the bonus awarded is reduced as the clicked point gets more distant from the center.

In order to make sense of the data produced by this game, we had to deal with two important issues: 1) assess the quality of the users (see Section 3.1) and 2) combine the different annotations to derive a single "best" representation of the objects (see Section 3.2).

3.1. Assess the quality of the users

The contribution of each user playing the game cannot be equal. In fact, there exist casual players that dedicate a little time playing, achieving, usually, low scores and on the other extreme, hardcore players can be found. Assessing user quality is of key importance for generating a ground truth based on the weighted contribution of users. The weight is the quality score itself, meaning that the higher a player's score is, the more influential her annotations will be in determining the final ground truth.

To estimate user quality we resort to the ground truth seeding technique, i.e. the first level of the game always contains a video for which a hand-drawn ground truth (G_{GT}) already exists. When the first level of the game ends, the acquired data (GT_u) of the user u is compared to the G_{GT}. Each submitted ground truth starts with a quality score (S_{GT}) of 1 and the number of False Positives (FP_u, a location where the user clicked but fish does not exist), False Negatives (FN_u, a location where the user did not click but fish does exist) and True Positives (TP_u, a location where the user clicked and fish does exist) are determined.

While a TP_u does not decrease the quality of the ground truth and a FP_u decreases it always, a FN_u is more complicated because it can occur for two reasons: 1) the user did not click on it at all, because she was not fast enough, or 2) because, at the same time, she was clicking on another object. In the former case, if the user was not fast enough to click, S_{GT} is decremented by N_{ft}/N_d, where N_d and N_{f_t} are the objects contained, respectively, in the G_{GT} and in the frame f_t. If the user was clicking other objects at the time that FN_u occurred, is determined by seeking for objects in frame f_t. If at least one such object exists, and it was shot by the user, no action is taken. Conversely, the user's quality score is decremented as before.

Summarizing the score of each submitted ground truth is given by:

$$S_{GT} = 1 - \frac{1}{N_d} \sum_{N_d} N_{false} \quad (1)$$

where

$$N_{false} = \begin{cases} 0, \text{if } Click \text{ is a } TP_u \text{or } (FN_u \text{ and } \exists\, TP_u \in Frame) \\ 1, \text{if } Click \text{ is a } FP_u \text{ or } (FN_u \text{ and } \nexists TP_u \in Frame) \end{cases}$$

If this is the first ground truth created by the user, her quality score is equal to S_{GT}. If, instead, previous assessments already exist, the quality score of the user is determined by:

$$S_u = \frac{1}{N_{Tot}} \sum_{i=1}^{U_{GT}} S_{GT_i} \times N_{GO_i} \quad (2)$$

where N_{Tot} is the number of objects in all the ground truths of the user, U_{GT} is the set of her ground truths, S_{GT_i} is the quality of i^{th} ground truth, given by (1), and N_{GO_i} is the number of objects in it.

3.2. Build the ground truth objects

Once the users obtain a quality score, their annotations can be integrated in order to build the best ground truth representations. In order to identify the locations that users clicked the most, we apply iteratively an unsupervised clustering algorithm. In particular, initially, a K-means analysis is applied with a predefined number of clusters (set to 10 or to the number of fish in the existing ground truth, if it contains more). The clustering result is further refined by iterating through each point (clicked by the user) and determining whether it fits well in the assigned cluster or not, by calculating the euclidean distance from the cluster's centroid. If such distance is over than a threshold T, it means that the point does not fit well into that cluster and it is removed from it. Afterwards, the euclidean distance of the removed point from the centroid of the other clusters is calculated. If a more suitable cluster (distance less than T) is found the point is marked as confirmed and it will be included in the next iteration. On the contrary, if no appropriate cluster exists, the point in question is excluded from successive iterations.

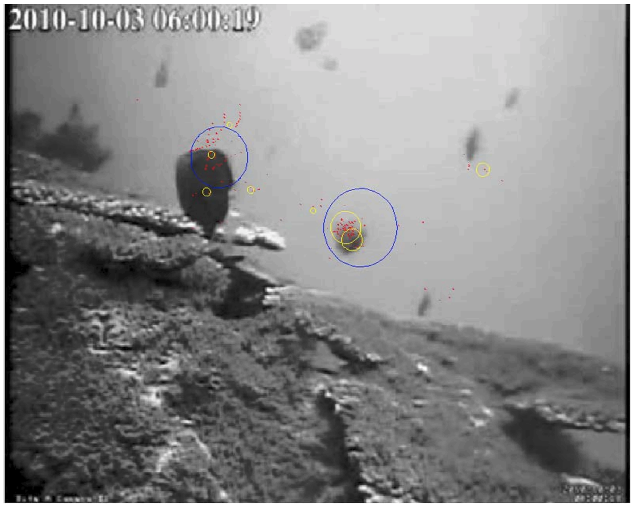

Figure 2. Clustering applied on the acquired data: Red dots are the locations clicked by the users. Yellow circles represent the result of the first clustering iteration, while the blue circles are the final result of the clustering method. The radius of each circle is equal to the sum of the quality scores of the users that made an annotation that belongs to that cluster, given by (3).

At each iteration, every cluster c is assigned a value that represents its significance, or radius, and is given by:

$$r_c = \frac{1}{N} \sum_{p}^{P_c} Q_{u,p} \tag{3}$$

where N is the total number of points in the current frame, P_c represents the points in that cluster and $Q_{u,p}$ is the quality of the user that created that point.

The algorithm stops when all the clusters have a value of $r_c > Tr$ (Tr empirically set to 0.4) or the initial maximum number of clusters is equal to zero. In case these conditions are not satisfied, the maximum cluster number is decreased by one and the algorithm proceeds with the next iteration.

The resulting clusters can be represented as heat maps, showing how the users' clicks are distributed over the scene.

When the algorithm execution ends, the obtained clusters are the objects of the "best" ground truth. In detail, each object is represented by the bounding box of the corresponding cluster.

Algorithm 1 shows the clustering algorithm, Fig. 2 shows an example output of the method described, where the 10 initial small clusters (in yellow) are reduced to 2 bigger ones (in blue) and Fig. 3 shows the heatmaps produced in a 4-frame sequence.

```
$MaxClusters = max(10, Count(ObjectsInGT))$;
$P = ClickedLocations$;
$C = Clustering(MaxClusters, P)$;
while ($MaxClusters > 0$) do
    $ClustersOK = True$;
    foreach k in C do
        $r_c = radius(k)$;
        if $r_c < T_r$ then
            $ClustersOK = False$;
            break;
        end
    end
    if $ClustersOK == True$ then
        Output: C
    end
    foreach p in P do
        $C_p$ = Centroid of cluster containing p;
        if $distance(p, C_p) > T$ then
            $Found = False$;
            foreach k in C do
                $C_k$ = Centroid of cluster k;
                if distance(p,$C_k$) $< T$ then
                    $Found = True$;
                    $break$;
                end
            end
            if $Found == False$ then
                remove p from P;
            end
        end
    end
    $MaxClusters = MaxClusters - 1$;
    $C = Clustering(MaxClusters, P)$;
end
```

Algorithm 1: The clustering Algorithm

4. Experimental Results

In order to test the accuracy and the efficiency of the tool, we compared the annotations generated from the game against a hand made accurate dataset. In particular, we used 7 hand-labeled videos (with frame rate and duration values as described in the previous section, for a total of 1568 frames) that contained 4140 annotated objects.

Clustering methods work better when applied to as bigger datasets as possible and for this reason we organized a Facebook event to gather users. To motivate the players we also offered a prize for the winner. For the event's duration (4 days), 80 users participated and played 1273 game sessions that resulted in 264316 annotations (Table 1).

For determining whether an object (BB, the bounding box of the clusters calculated by the algorithm described in Section 3.2) is either a hit or a miss we calculate the overlap, or PASCAL score, between it and the objects (GT) in the same frame in the hand-drawn ground truth given by:

$$O_{score} = \frac{area(BB \cap GT)}{area(BB \cup GT)} \tag{4}$$

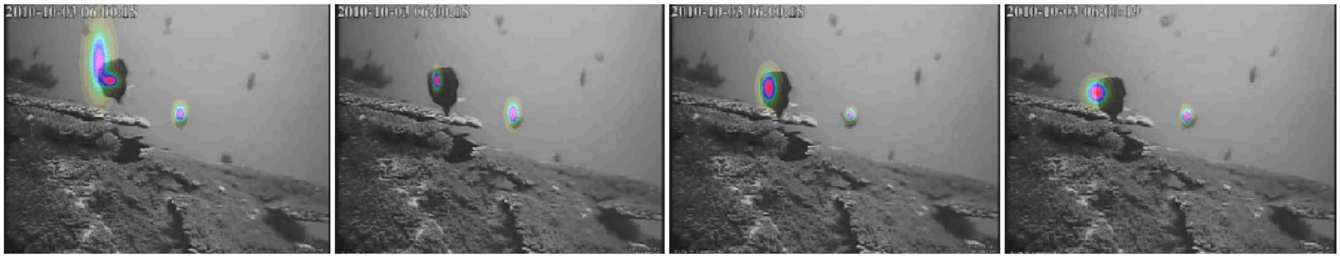

Figure 3. Heatmaps of two fish detected in an 4-frame sequence.

Level	Annotated objects	Acquired clicks	Number of users	Precision	Recall	F_1
1	722	71105	80	0.71	0.69	0.70
2	1847	70406	80	0.69	0.64	0.66
3	593	58528	69	0.70	0.64	0.67
4	251	47137	52	0.74	0.71	0.72
5	446	16276	46	0.57	0.51	0.54
6	104	522	19	0.31	0.21	0.25
7	177	342	18	0.26	0.09	0.13
Total	4140	264316	80	0.66	0.60	0.63

Table 1. The datasets we used for performance evaluation and the results obtained. The precision and recall values refer to the case where the annotations of all the users that played the corresponding level were used and the values in the last row are the weighted averages with respect to the number of ground truth objects.

If there exists at least one object where such value is greater than a threshold, empirically set equal to 0.7, then the cluster is considered a true positive; conversely, it is considered a false positive. A false negative is an object in the GT that has no corresponding BB.

The performance of the application was evaluated in terms of precision, recall and F_1 measure, given by:

$$Precision = \frac{TP}{TP + FP}, \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

and

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}, \tag{7}$$

respectively. The obtained results, together with the dataset used for testing the application, are shown in Table 1.

All the players, reached level two, but with the progressively increasing difficulty, a little less than 25% (19 out of 80) of them reached the last one. The absolutely best performance was achieved at the fourth level, where the precision, recall and F_1 values were 0.74, 0.71 and 0.72 respectively. This was due to the fact that it has the highest clicks/object ratio, which was about 187 at that level.

It should be noted that the turning point of the game is the fifth level where more than half players who played it could not advance through it (27 out of 46). As a consequence, it severely hampered the performance of the clustering method at the last two levels, where an inadequate number of annotations was collected (a little less than 2 clicks per object at the last level). In particular, the recall value at that level was too low, 0.09, because of the high number of false negatives (i.e. many objects in the GT did not have any clicks at all). Furthermore, we also analysed how the user quality influenced the method's performance. First, we excluded all the annotations made by the users who had a quality score lower than 0.7. We noted that this choice influenced minimally the precision, whereas it affected the recall. This result is explained by the fact that the true positives TP_u of the low-quality users were able to raise the r_c value over the threshold T_r (see Section 3.2), thus, resulting in a lower number of false negatives. Conversely, the false positives FP_u produced by the same users were not enough to create false positive clusters (Fig. 2) because this creation depends also on the user quality. More quantitatively, for level 1, the precision kept almost stable (0.69), whereas the recall dropped to 0.37. This also explains why the performance decreased drastically at the higher levels (Levels 6 and 7) when the number of players decreased. On the contrary, when we excluded the users with quality higher than 0.7, both precision and recall dropped to, respectively, 0.44 and 0.25.
The user quality allows us, therefore, to keep balanced precision and recall, whereas, the number of users serves to support the explorative nature of the game, i.e. more users play, higher is the probability (which also depends on the users' quality) to detect correclty objects.

5. Concluding Remarks

In this paper we presented *Flash the Fish*, a simple online game that aims at generating video annotations for object detection algorithms. The acquired annotations were then fed to a clustering module which refined the results, producing good quality ground truth and we are confident that the quality will increase as more and more users will play the game.

While, in its current form, the game generates ground truth for object detection, as the number of annotations in-

creases, it should be interesting to assess whether it is possible to derive the exact object shapes from the heatmaps. While a preliminary analysis demonstrated that this can be possible, a very large dataset should be considered. To accomplish that, we are going 1) to integrate advanced voting methods [12] and 2) to use the clicked points as initial seeds for object segmentation approaches.
Bonus levels that permit the creation of annotations for testing object tracking and object classification methods are already implemented and are currently under testing.
One interesting observation that we made during the revision of the datasets is that many clicks should fit better in successive frames from the one that they were acquired. This happens due to the fact that the time that passes from the moment the eye catches the visual stimulus (fish moving) to the moment of the reaction (mouse movement and click) is not negligible and it should be taken into account. For this reason we are currently developing a module that analyses the reflexes of each user independently by controlling how well the clicks fit with the "best" ground truth and, eventually, introduces a delay.

Given the effectiveness of this game, we aim at creating an open platform, where researchers can upload their videos, to be used in the game, and the generated annotations will be publicly available.

References

[1] L. v. Ahn. Games with a purpose. *Computer*, 39(6):92–94, June 2006.

[2] A. Ambardekar, M. Nicolescu, and S. Dascalu. Ground truth verification tool (GTVT) for video surveillance systems. In *Proceedings of the 2009 Second International Conferences on Advances in Computer-Human Interactions*, ACHI '09, pages 354–359, 2009.

[3] O. Barnich and M. V. Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, pages 1709–1724, 2011.

[4] M. Bertini, A. Del Bimbo, and C. Torniai. Automatic video annotation using ontologies extended with visual information. In *Proceedings of the 13th annual ACM international conference on Multimedia*, MULTIMEDIA '05, pages 395–398, 2005.

[5] L. Biewald. Massive multiplayer human computation for fun, money, and survival. In *Proceedings of the 11th international conference on Current Trends in Web Engineering*, ICWE'11, pages 171–176, 2012.

[6] D. Doerman and D. Mihalcik. Tools and techniques for video performance evaluation. In *Proceedings of 15th International Conference on Pattern Recognition.*, volume 4, pages 167–170, 2000.

[7] P. Heroux, E. Barbu, S. Adam, and E. Trupin. Automatic ground-truth generation for document image analysis and understanding. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 01*, ICDAR '07, pages 476–480, 2007.

[8] C.-J. Ho, T.-H. Chang, J.-C. Lee, J. Y.-j. Hsu, and K.-T. Chen. Kisskissban: a competitive human computation game for image annotation. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 11–14. ACM, 2009.

[9] C. Jaynes, S. Webb, R. Steele, and Q. Xiong. An open development environment for evaluation of video surveillance systems. In *PETS02*, pages 32–39, 2002.

[10] I. Kavasidis, S. Palazzo, R. Di Salvo, D. Giordano, and C. Spampinato. A semi-automatic tool for detection and tracking ground truth generation in videos. In *Proceedings of the 1st International Workshop on Visual Interfaces for Ground Truth Collection in Computer Vision Applications*, VIGTA '12, pages 6:1–6:5, 2012.

[11] I. Kavasidis, S. Palazzo, R. Di Salvo, D. Giordano, and C. Spampinato. An innovative web-based collaborative platform for video annotation. *Multimedia Tools and Applications*, pages 1–20, 2013.

[12] X. Li, B. Aldridge, J. Rees, and R. Fisher. Estimating the ground truth from multiple individual segmentations with application to skin lesion segmentation. In *Proc. Medical Image Understanding and Analysis Conference, UK*, pages 101–106, 2010.

[13] A. J. Quinn and B. B. Bederson. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 1403–1412, 2011.

[14] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier. Collecting image annotations using amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 139–147, 2010.

[15] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *Int. J. Comput. Vision*, 77(1-3):157–173, May 2008.

[16] A. Sorokin and D. Forsyth. Utility data annotation with Amazon Mechanical Turk. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, June 2008.

[17] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, pages 319–326, New York, NY, USA, 2004. ACM.

[18] J. Yuen, B. C. Russell, C. Liu, and A. Torralba. Labelme video: Building a video database with human annotations. In *ICCV'09*, pages 1451–1458, 2009.