CAPTCHA-based Image Labeling on the Soylent Grid

Peter Faymonville[1], [*] Kai Wang[1], John Miller[2], Serge Belongie[1] [†]
[1]Department of Computer Science and Engineering and [2]Calit2
University of California, San Diego
peter@faymonville.de, kaw006@cs.ucsd.edu, j8miller@soe.ucsd.edu,
belongie@cs.ucsd.edu

ABSTRACT

We introduce an open labeling platform for Computer Vision researchers based on CAPTCHAs, creating as a byproduct labeled image data sets while supporting web security. For the two different tasks of annotation and detection, we provide a security analysis and explore usability issues. We present the interfaces used by researchers, website administrators and users and experimental results obtained using the platform. Finally, we discuss system sustainability issues in the context of a broader "ecosystem" for the platform.

Categories and Subject Descriptors

K.6.5 [**Computing Milieux**]: Management of Computing and Information Systems—*Security and Protection*

1. INTRODUCTION

In today's web surfing experience, the task of solving a CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) has become ubiquitous, with application ranging from preventing comment spam to stopping automatic account creation. With the advent of ReCaptcha [8], the idea of using these tests to create labeled data sets was introduced. Our labeling tool on the *Soylent Grid* platform advances this idea in two ways, (1) by supporting more than one task and (2) by creating an open platform for all researchers to get their data sets labeled. Furthermore, we address the security and quality-of-service problems resulting from the openness. Given the unrelenting demand for labeled image data sets by the AI community, we assume for sake of argument the availability of a bottomless source of image data sets at our disposal.

Due to the variety of tasks and data sets, the system becomes harder to attack if the different tasks and data sets are displayed at random and security measures for retiring data sets and/or tasks exist. Therefore, the surrounding infrastructure around a CAPTCHA doesn't have to be implemented each time a research group decides to try a new CAPTCHA method and long-term relationships to CAPTCHA-dependent websites can be established, reducing the integration costs for these websites.

This research was motivated through the GroZi project, which aims at creating a technological solution for grocery shopping for blind and visually impaired users. Part of this solution will be a mobile device with a camera running a text-recognition algorithm. Providing training data for this algorithm was the initial goal of this research project.

2. RELATED WORK

This paper builds on [6], which described the general SoylentGrid infrastructure. We now present an implementation, experiments and an ecosystem based on this infrastructure tailored to image labeling.

The idea of a CAPTCHA has first been formally introduced in [1], along with the proposal to use hard AI problems for CAPTCHAs. CAPTCHAs based on recognizing objects – specifically animals – in images are proposed in [2]. Initial observations on the usability of CAPTCHAs can be found in [9]. In [8], the authors present a way to display a CAPTCHA problem that simultaneously provides security and gathers user input for improving OCR systems, which is referred to as ReCaptcha.

The work of [10] also describes a general framework for human computation involving researchers, website administrators, and users. In that work, they broadly outline different models for how the three entities could potentially interact – including models where money or services are exchanged between the person needing human computation and internet user. In contrast, our framework observes that each party provides a different value to the system, and we position them to interact in a cooperative setting. Users provide human computation cycles and desire content from websites. Website administrators want users to access their content but desire security. Researchers provide data to be used for security but desire human computation. This work describes how we channel the goals of each group so that their strengths are leveraged and needs are satisfied.

[*]Visiting Student from University of Potsdam, Germany
[†]Contact author.

3. PROPOSED APPROACH

The SoylentGrid platform keeps track of data from researchers that require human annotation. These data sets contain elements that are hand labeled by the researchers themselves – the control data – and elements without labels that need to be labeled by users – the experiment data.

When a data set is initially uploaded to SoylentGrid, we first subject it to a period of evaluation to gauge the difficultly of the labeling task; this period of time is referred to as a data set's *pilot study*. Details about how we conduct the pilot study are in Section 6.3. After this evaluation is complete, the data set is added to the SoylentGrid and is assigned a priority for how often the data from a particular researcher's set is presented to users. A higher priority for a data set means the data will be labeled sooner. The details of how we determine priority are discussed in Section 6.

From a user's point of view, we present each of the tasks twice side-by-side. Following the approach of [8], one of them is the experiment task, the other one the control task. For the control task, the task solution is already known, whereas for the experiment task, the solution is the label sought by the researcher. The data for the tasks is drawn at random respectively from the experiment and control data sets. It is not revealed which task is experiment and which is control.

If the solution for the control task is correct, the user is granted access to the secured resource and the experiment task data is saved. If the solution is not correct, the user is presented with a new task. This approach provides a CAPTCHA, as long as the given task can only be solved with low probability by a computer program.

Since the user interface of our CAPTCHA is going to be embedded into another website, the user interface has to be constrained in size, our implementation is 400 pixels wide and 200 pixels high. Each of the displayed images is resized to 150x150 pixels.

3.1 Definitions

Three stakeholders participate in the platform:

Researchers: Provide image data sets and need labels

Website Administrators: Need portions of their website protected by CAPTCHAs

Users: Want access to a CAPTCHA-secured resource

Three different tasks are implemented:

Annotation: Type a text label given a bounding box

Detection: Draw a bounding box given a text label

Both: A combination of both tasks

In the following sections, we define each of the stakeholders and their associated tasks.

Figure 1: User Interface 'Both'-Task. Displayed are two product images with bounding boxes around the words to be entered already drawn. The words are also entered already. For the annotation task, the bounding boxes would be supplied by the task, for the detection task, the label would be supplied by the task.

3.2 User tasks

Users want to access a resources on the internet, and an attempt to access one protected by SoylentGrid results in a prompt to complete one of a number of tasks that acts as a CAPTCHA. We describe two tasks we've implemented for our experiments: image annotation and object detection.

In the annotation task, an image is displayed together with an overlay rectangle highlighting a part of that image. The user is supposed to type in the word displayed in the highlighted area. Therefore, we can support labeling multiple areas per image. The user can click on the image to see a larger version of the image as an overlay to the complete website.

For the detection task, a user is given a word and is supposed to draw a bounding box around the given word by either clicking on one corner and dragging and releasing the mouse or by clicking on two corners of the box. In each method, a tentative rectangle follows the cursor.

The 'both'-task is a combination of the two tasks: The user chooses a word from the presented image, draws a rectangle around it and types it. This task has the highest requirement for control data items, because all possible words and their bounding boxes in the image have to be previously known. This task can be seen in figure 1.

3.3 Researcher tasks

From an researcher's point of view, SoylentGrid is used as an image labeling tool. He provides the system with his data set and configuration parameters. The data set contains labeled control data and and unlabeled experiment data. The configuration parameters consist of the level of tolerance to accept labels as being correct and the number of times a particular unit of data needs to be consistently labeled by users to satisfy the researcher.

3.4 Website Admin tasks

For a website administrator, the motivation to use a CAPTCHA comes from the need to protect certain resources of a web-

site, i.e. web forms, account creation sites, search boxes, due to concerns for the site performance, spam or other malicious behavior. A website administrator needs to integrate our CAPTCHA solution into his web application to make sure that his resources are effectively protected. Key to the adoption of CAPTCHAs is to provide an easy mechanism to use them with popular content management systems such as Drupal and Wordpress. Therefore, we're planning to publish these plugins to facilitate widespread adoption of the platform.

3.5 Usability

With each data set, the researcher who uploaded it has defined a level of tolerance to specify how exact an annotation needs to be for their data set.

For the annotation task, we define our tolerance to be a string edit distance [4] of one between the user supplied annotation and researcher provided annotation in the control data. In the detection task, a our bounding box tolerance allows detections to be correct if each corner of the bounding box is in a 11-by-11 pixel window centered at the corresponding corner of the control bounding box.

The measures of tolerance are specified by the researcher for his particular data set based on their individual need. This parameter plays an important role in affecting how *difficult* a task is for a user and the implications of how this is set are explored in Section 6.

3.6 Security

3.6.1 CAPTCHAs and Internet Security

In the history of CAPTCHAs, we can observe a cat-and-mouse game between CAPTCHA-providers and the development of CAPTCHA-breaking programs, especially when it comes to CAPTCHAs based on text distortion. Those are still widely in place today, but as they're getting harder to break, they've become increasingly complex and frustrating for humans. See [9] for examples.

For our platform, we benefit from the variety in tasks and data sets, it is expected that no single automated program can break all of the different control tasks and data sets. Ensuring the security of the solution then reduces to recognizing broken tasks, data sets and guaranteeing a continuous incoming stream of new, more difficult control data. The diversity of the data in our system is tightly coupled with its security.

3.6.2 Possible Attack Vectors

We now present an analysis of common attack methods on our tasks. The analysis of these methods depends partly on the parameters supplied by the researcher, since a box similarity definition allowing a larger tolerance or a more tolerant string similarity definition would increase the chances of automated programs breaking the CAPTCHA.

Annotation. The trivial attack on a task like this would be to create a dictionary of words in the english language and commonly used product names, choosing one word at random to get through the task. The chance of getting the control right is therefore the inverse of the number of words in the dictionary (m).

$$p_{ann} = \frac{1}{m} \tag{1}$$

Detection. The trivial attack on the detection task would place a random bounding box on the given image surface. The probability of success for this method is given by

$$p_{det} = \left(\frac{11}{150}\right)^2 \cdot \left(\frac{11}{150}\right)^2 \approx 0.00002892... \tag{2}$$

Both. Since this task requires an automated program to solve both tasks successfully, we have

$$p_{both} = p_{ann} \cdot p_{det} \tag{3}$$

While more sophisticated methods to attack the system exist, describing them falls outside the scope of this paper.

4. EXPERIMENTS

Our current implementation is using the Google Web Toolkit (GWT) and uses the wz_jsgraphics JavaScript library for drawing the bounding boxes.

For our experiments, we used the GROZI-120 data set, which has been introduced in [3]. It consists of product images for 120 grocery products, which are completely labeled and can therefore be used either for experiment or control data sets. Because multiple product images are given per product and multiple labeled regions exist in each image, 881 elements were used, each consisting of an image file, a bounding box and an associated label.

Out of this pool, we drew 20 elements at random to put them into the separate experiment data sets for each task type. The common control pool consisted of all elements.

We recorded the following measures, each separate per task type: IP-address, task duration, control image, experiment image, supplied experiment data and a timestamp.

Two separate experiments have been conducted on a focus group of users. For the second experiment, we dropped the 'Both'-task, added the simple edit distance and ignored capitalization for the annotation similarity definition.

4.1 Results

The results of our experiments are presented in table 1.

4.2 Evaluation of User Input

We evaluated the user input for both task types of the second experiment. The ground truth for this comparison was available to us since we took the experiment images from our GROZI-120 data set.

The distribution of entered labels for the 20 experiment images in the annotation task can be found in table 2.

TaskType	#Presented	#Successful	Success ratio	Avg. duration (sec)	Std. Dev. duration (sec)
Annotation	171	71	0.42	20.0	51.4
Detection	168	82	0.49	28.7	38.1
Both	238	59	0.25	40.3	42.1
Annotation	201	185	0.92	7.3	3.8
Detection	367	181	0.49	12.8	6.0

Table 1: Experimental results for Experiment 1 (top) and Experiment 2 (bottom)

Figure 2: User supplied bounding boxes. The green box marks our ground truth for this particular image.

In figure 2, we show one of the experiment images with an overlay of all user supplied boxes and the ground truth box. The average overlap of the user-supplied bounding box with the ground truth box was 82.71%.

5. DISCUSSION

The experiments were conducted with two focus groups consisting of 9 people, the two groups were mostly overlapping. They consisted of undergraduate engineering students, which gave extended feedback on the implementation of the tasks. Among the group, the students ranked the detection task to be more difficult than the annotation task. However, when considering a setting where the CAPTCHA is displayed on a mobile device with a touch interface, we expect the relation to be reversed.

One important observation is the increase of the success ratio on the annotation task from experiment 1 to experiment 2. Apart from increased familiarity with the system, we introduced the string similarity definition and ignored capitalization, which increased the success ratio significantly.

The differences in duration between experiment 1 and experiment 2 are largely attributable to stricter instructions given out to the focus group. As these experiments took

# of images	17	1	1	1
# different labels	1	2	3	3
% Confidence most supplied label	100	90	57	75

Table 2: Distribution of user entered labels

30 minutes (experiment 1) and 15 minutes (experiment 2) respectively, we didn't allow any resting on a displayed task in experiment setting 2. The relative comparison of the average task duration supports the difficulty ranking of our focus group.

Other possible task designs have been proposed in [5], for example the collection of location information of objects in an image by clicking on a raster of given superpixels.

6. SOYLENT ECOSYSTEM

Upon deploying the infrastructure for the SoylentGrid, we want to have the dynamics in place to allow the system to govern itself without human intervention: deciding what data to display to web users in a manner thats both fair and is consistent with the goals of all parties involved. Recall, in the SoylentGrid, we have three groups of people with different objectives: users, website administrators, and researchers. Users want to spend the minimum amount of time working on a task to reach their desired page. On the other hand, website administrators want ease of use for their surfers as well – as to not turn them away – but they also want a high level of security against bots. Finally, researchers are primarily interested in getting their data labeled as quick as possible. A successful set of dynamics will provide incentives to maximize all these measures. We provide a sketch of how such a system would look by basing them on rules governing online ad auctions for search engines.

6.1 Relation to online ad auctions

In online search advertising, advertisers pay search engines to have their text ads be displayed on the results page of relevant keywords [7]. In this setting, the price of placing the ad is determined by an online auction between advertisers interested in the same keywords. A user searching on a keyword is presented their list of results from a search engine along with a list of ads, and if they click through on an ad, the advertiser is charged a fee based on their bidding price. To maximize profits, its in the best interest of a search engine to order the ads based on both the bidding price of and advertiser and the relevance of their ad to the keyword search – to generate the most user clicks. Next, we introduce notation to describe the particular auction system used by

Google and describe how the concepts can be applied to the selection problem in SoylentGrid.

Let $a = 1, ..., A$ represent advertisers who are bidding a price b_a for a position on Google's search result page for a particular keyword. For simplicity, we consider the situation where advertisers are all bidding for the same keyword. Let e_a be the predicted textual relevance of an ad to the keyword. Google's ads are ordered by $(b_a e_a)$; the product of an ad's relevance and bid price. For further detail, we refer to the explanation found in [7]. Table 3 shows the mapping between elements in Google's ad auctions to elements in SoylentGrid.

Let $r = 1, ..., R$ represent researchers who are providing a proportion c_r of their data as a labeled control set. Let d_r be the predicted difficulty of a labeling task for a web user. In the context of SoylentGrid, being displayed higher on the list of advertisements corresponds to having a higher probability of being selected as the data to present to a web user. The probability of a r's data being selected is: calculated as: $\frac{(c_r d_r)}{\sum_i^R (c_i d_i)}$; a product of the amount of control data provided and its difficulty.

Google	SoylentGrid
Position of ad	Likelihood of displaying task
Advertiser: a	Researcher: r
Bid price: b_a	Control data set size: c_r
Pred. ad relevance: e_a	Pred. task difficulty: d_r
Determine position: $b_a e_a$	Determine likelihood: $\frac{(c_r d_r)}{\sum_i^R (c_i d_i)}$

Table 3: Mapping concepts from Google's ad auction system to SoylentGrid

While Google's system is optimized for maximizing ad revenue and user experience, we intend for our mapping to maximize the size of our control data set – which means improved security for websites – and minimize the strain put on web users in reaching their web content. Next, we describe how to estimate the difficulty of labeling tasks.

6.2 Data set evaluation

Difficulty is measured by the amount of time the average web user spends completing a task and their success rate on the provided control set. When a task takes a web user a long time to complete and is challenging for them to correctly label, then its considered difficult. There are two primary reasons for tasks being difficult: the inherent complexity of the task and the level of tolerance for what counts as a correct label.

Its clear that some labeling tasks take more time and are more error-prone than others; typing in an obscured word is easier than segmenting an image. In this case, researchers don't have much flexibility if what they need is indeed complex.

When a researcher uploads their data to the SoylentGrid, they will be able to specify a level of tolerance for the labels they accept. Presumably, a higher level of tolerance will result in the data being labeled acceptably more often. In the context of Google's ad system, increasing the tolerance is comparable to modifying the text of an ad to be more relevant to a keyword. In both cases the advertiser/researcher has the choice to boost their priority in their respective systems in a way thats possibly of detrimental to their goals. An ad phrased to sound more relevant may be less representative of the actual product or service, resulting in the advertiser attracting – and being charged – for user clicks that are outside of their intended audience. Similarly, a researcher increasing the tolerance on their labels may gather data that's less useful as training data. To gauge the difficulty of a data set, we use a a pilot study to evaluate every uploaded data set before deciding their priority and incorporating them into the full SoylentGrid.

6.3 Pilot study

Before mixing a researcher's data into the larger SoylentGrid, we first evaluate it in a testing phase for some set amount of time. In this phase, web users are still presented with the data, as with the standard SoylentGrid, but the control and test tasks are only taken from the particular researcher's data set and the success rate and task completion duration are measured. After the pilot period is over the data is moved into the larger SoylentGrid and is displayed to web users at a priority that was a function of the measured difficulty and up front label set size. At this point, the researcher also has the choice to either lower the tolerance on their task or label more of the data on their own to alter their data set priority.

6.4 Objectives of the ecosystem

From the perspective of the SoylentGrid, the main thing that can be controlled is what data gets to be presented to a user at any time. Our hypothesis is that giving priority to the data sets that are least difficult and supply the highest proportions of control data will satisfy all three groups involved in SoylentGrid.

Researchers. By making the prioritization scheme explicit, researchers are given the most control over the speed at which their data gets labeled. They have the option of 'increasing their bid' by hand labeling a larger control set, or 'increasing the relevance of their ad' by increasing the tolerance of their labels.

Users. Simply, users will get the data that is measurably least difficult, allowing them to access their content with the least amount of effort.

Website administrators. In addition to minimizing user burden, prioritizing researchers that supply the most control data results in greater security against automated bots.

Following the principles used in online ad auctions, the SoylentGrid can be a self-governed system that fulfills the priorities of all parties involved.

7. REFERENCES

[1] L. V. Ahn, M. Blum, and J. Langford. CAPTCHA: Using Hard AI Problems For Security. In *In Proceedings of Eurocrypt*, pages 294–311. Springer-Verlag, 2003.

[2] J. Elson, J. R. Douceur, J. Howell, and J. Saul. Asirra: a captcha that exploits interest-aligned manual image categorization. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 366–374, New York, NY, USA, 2007. ACM.

[3] M. Merler, C. Galleguillos, and S. Belongie. Recognizing groceries in situ using in vitro training data. In *SLAM*, Minneapolis, MN, 2007.

[4] G. Navarro. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1):31–88, 2001.

[5] A. Sorokin and D. Forsyth. Utility data annotation with amazon mechanical turk. pages 1–8, June 2008.

[6] S. Steinbach, V. Rabaud, and S. Belongie. Soylent Grid: it's Made of People! In *Workshop on Interactive Computer Vision (ICV)*, Rio de Janeiro, 2007 2007.

[7] H. R. Varian. Position auctions. *International Journal of Industrial Organization*, 2006.

[8] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science*, 321(5895):1465–1468, 2008.

[9] J. Yan and A. S. El Ahmad. Usability of captchas or usability issues in captcha design. In *SOUPS '08: Proceedings of the 4th symposium on Usable privacy and security*, pages 44–52, New York, NY, USA, 2008. ACM.

[10] Y. Yang, B. B. Zhu, R. Guo, L. Yang, S. Li, and N. Yu. A comprehensive human computation framework: with application to image labeling. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia*, pages 479–488, New York, NY, USA, 2008. ACM.

APPENDIX

A. ACKNOWLEDGMENTS

The authors wish to thank the UCSD TIES (Teams in Engineering Service) GroZi Spring 2009 team for their support in the experiments. This work was supported in part by an NSF Graduate Research Fellowship for K.W. and NSF CAREER Grant #0448615 to S.B. with an REU supplement for P.F. J.M. was supported in part by the DARPA grant "Computer Science Futures II: Engaging Young Scholars in Computer Science" (NBCH1080007 subaward Z931303) awarded to S.B.