iCAPTCHA: Image Tagging for Free

Rohit Ashok Khot, Kannan Srinathan
Center for Security, Theory and Algorithmic
Research (C-STAR),
International Institute of Information
Technology, Hyderabad. India 500032
rohit_a@research.iiit.ac.in, srinathan@iiit.ac.in

Abstract

Semantic annotation or tagging of images can greatly improve the accuracy and efficiency of image search engines. However, humans rarely annotate images as they find the task of image annotation boring and laborious despites its benefits in terms of search and retrieval. In this paper, we introduce a novel approach of luring users into image annotation: by embedding image annotation into a CAPTCHA design. A CAPTCHA is a standard security mechanism used by popular commercial websites to prevent automated programs from abusing the online services. Millions of users solve CAPTCHAs daily in order to access web content and services. We aim to utilize human effort spent in solving the CAPTCHA into a productive work of image annotation. We introduce iCAPTCHA, a user friendly and productive CAPTCHA design. Our premise is based on the human ability to recognize images and label them in proper categories. Each time a user solves an iCAPTCHA, he/she is helping to label images in proper categories which will in turn improve image search and retrieval.

1. Introduction

Web, as experts say, is leaving the era of search and entering the era of discovery. With the dawn of content based user websites (e.g. blogs, forums, social networks), people are motivated to communicate with each other and express themselves by sharing images, videos and thoughts (blogs). As a result, visual information (images as well as videos) is widely available on diverse topics and from multiple sources. However, it will require a substantial effort to properly organize this information outburst. Modern image search engines like Google [1], Yahoo [2] collect and index images from other sites, in an attempt to provide access to user to the wide range of images. However, they often struggle to find the right image for a specific need and to reduce the clutter that comes along with the selection. The problem is essentially due to following two factors:

1) Query Dependency:
Current image search engines require users to be specific in terms of the search query while seeking for the visual targets. Most of the times however, it is hard for users to express the need in words. As a result, search query tends to be short, too general and sometimes ambiguous. If the query is not detailed enough, search engine returns plenty of information (images) consisting all subcategories. User then needs to laboriously browse through all information or keep on refining query to get the desired result (image). For example, while searching for images of old Indian actress *Amrita Singh*, it is better to type “Amrita Singh” as query than a general query “Amrita”, which would result in set of images mostly dominated with images of “Amrita Rao”, another popular Indian actress. However, for this to happen, user must know complete name of the actress (detailed query), which many users may not know.

2) Talking with words when we mean images:
Image search engines are word matching tools which analyze the metadata associated with the image (e.g. tags, keywords, and text in same page) for indexing and categorization of images. They assume that content of an image is related to adjacent text appearing in the page. However, this assumption is insufficient because text adjacent to images is often scarce and can be misleading sometimes [3].

It is therefore, essential that image possess meaningful and extensive metadata (often in the form of tags) to facilitate its access. If the image collection has been extensively annotated, technique such as faceted search will help user filter down a collection and show potential targets for browsing [4].

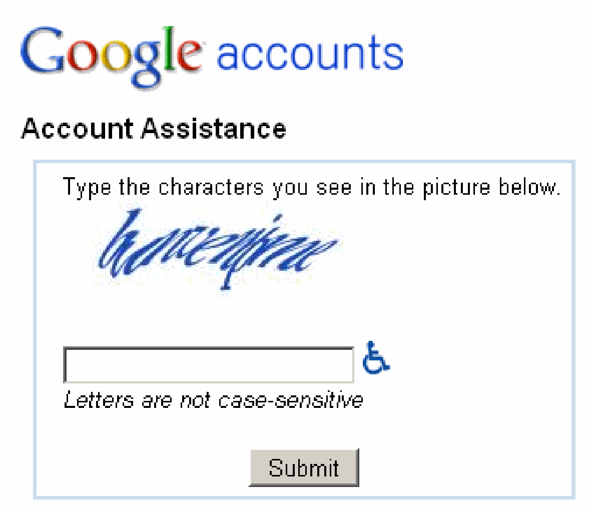
Current computer vision algorithms try to extract meaning by analyzing the visual content (e.g. shape, color, and texture) of the image. However, such approach have found limited success only in specialized setting and are yet to match the performance of humans in image recognition and understanding [5]. Humans on the other hand, have little difficulty in describing the image. However, they find the task boring and laborious. Attempts have been made to lure humans in annotating (tagging) images by embedding tagging into a social activity (e.g. Flickr)[6], by providing monetary incentives (e.g. Amazon Mechanical Turk)[7] and by designing special purpose games (e.g. ESP) [8].

In this paper, we introduce a different *yet efficient* approach of manual image annotation*:* by embedding *tagging* into a CAPTCHA design. A CAPTCHA is a standard security mechanism used by popular commercial websites to prevent automated programs from abusing the online services (For example, spreading junk e-mails, polluting online polls, and grabbing thousands of free email accounts). Success of our approach is two fold. First, Millions of users daily solve CAPTCHAs to access website content or web services. Therefore, by embedding image tagging in *a* CAPTCHA mechanism, image tagging is getting done at relatively low to no cost. Users will solve CAPTCHAs because they need to access the web services and not because they want to do the laborious task of tagging. Secondly, with an image based CAPTCHA, we improve the CAPTCHA design by making it more accessible, user friendly and robust than traditional text based CAPTCHA which has been suffering with accessibility and robustness issues.

The rest of the paper is organized as follows. We first discuss the existing CAPTCHA designs, the need for an image based CAPTCHA and how it can help in getting manual metadata for images (in annotating images). In the follow up section, we introduce iCAPTCHA. We describe the complete design and prototype implementation. Finally we conclude with the summary of our design.

2. Background

Internet has been one of the best things happened in the last few decades. It has transformed the way we live and look at the world around us. Most of us start the day by checking emails and reading news articles over the web. From train reservations to online shopping, from chatting to file sharing, almost all essential services are now available online and at free of cost. However, we never had thought that one day we would have to prove our intelligence (that we are human) before we can access any of these services. For example, most of us surely would have encountered a similar crazy image while accessing a popular site Google, as shown in Figure 1.



Type the characters you see in the picture below.

Letters are not case-sensitive

Figure 1: an example of Google CAPTCHA.

This image contains several distorted characters that we must correctly identify and type, in order to access the site content. Since current computer programs like Optical Character Recognition (OCR) are yet to achieve the accuracy of human eye while reading a distorted text, this image represents an instance of a test which is easy for humans to solve and yet difficult for computers to solve. This test is widely known as CAPTCHA [9]. A CAPTCHA stands for '*Completely Automated Public Turing Test to tell Computers and Humans Apart*'. It is a program that can generate and grade tests that humans can pass but current computer programs cannot [10].

The need for such differentiation or the CAPTCHA arose from the vulnerabilities of online forms and data entry. Before CAPTCHA, there was no easy way for a system to verify that the form is filled by a user (human) and not by some automated program running on behalf of the user. A classic cited example is of online polls conducted by Slashdot website [11]. The website conducted a poll to determine the best graduate school in USA. At the end of the polls, MIT and CMU not surprisingly, stood tall in terms of the gathered votes against all other colleges. However, the real reason behind this success was the execution of automated programs, giving plenty of fake votes to MIT and

CMU. Polluting an online voting system was just one example which shows the power of automated script attacks. Other examples include creating fake email accounts, spreading plenty of junk emails etc. CAPTCHAs work as sentries against these attacks, since solving CAPTCHA is difficult for automated programs and is relatively easy for humans.

Today, most of the popular websites like Google, Yahoo, and Wikipedia use CAPTCHAs as a standard security mechanism to defend automated script attacks. As a result, their online services are now not directly accessible. A user must solve the CAPTCHA to access the service. However, solving a CAPTCHA requires a substantial human cognitive effort. Based on the type of cognitive effort required to solve CAPTCHA, CAPTCHAs can be classified into three categories.

1) Text based CAPTCHAs:

They require users to read and type distorted text rendered in an image.

2) Audio based CAPTCHAs:

They rely on sound or speech recognition by the users.

3) Image based CAPTCHAs:

They ask users to perform an image recognition task.

Text based CAPTCHAs are the most popular of the three, considering their ease of deployment, intuitiveness and potential to offer reasonably good security. However, many of the existing text based CAPTCHA implementations [12, 13, 14] have been broken recently. It has prompted the CAPTCHA designers to create more complex (distorted) CAPTCHAs (like the one in Figure 1) taking away its usability. As we can see in Figure 1, the shown CAPTCHA image is barely readable by human eye, causing strain to the eye and fatigue by unnecessary multiple solving attempts. Therefore, CAPTCHAs are effective only if they are robust (computers can not solve them) and usable (humans can solve them) [15]. Unfortunately, text based CAPTCHAs fails to achieve both robustness and accessibility (usability) simultaneously which prompt us to look for other possible alternatives. Image based CAPTCHA is one such alternative because recognizing images are far better and fun than reading complex distorted text. This approach was first proposed by Tygar et.al in [16] where they discussed alternate image recognition CAPTCHA designs. Other attempts in creating image based CAPTCHA include Assira from Microsoft [17] and hotCAPTCHA from hotornot website [18]. However, all the proposed image based designs were created only as suitable alternatives to text based CAPTCHAs. On the other hand, we are also interested in tapping the human effort spent in solving CAPTCHA into a useful work.

2. Motivation

People around the world, solve millions of CAPTCHAs everyday, if put together, will easily amount for hundred or thousand hours of human effort per day [10]. Although the main purpose of CAPTCHAs is to prevent automated script attacks, the effort humans put in to solve them is otherwise getting wasted.

We thus ask a question:

“Can we channel the wasted human effort into some productive work? If yes then how”

The idea of productive CAPTCHA was first introduced by Luis Von Ahn, the man who also invented the CAPTCHA mechanism. He proposed a novel CAPTCHA design called as reCAPTCHA, which helps in reading and archiving old textbooks. The OCR (Optical Character Recognition) software used in reading books, can not effectively interpret text from the old books that has become pale, dirty and yellow over the time. On the hand, human eye can easily pick and figure out what the text is. In reCAPTCHA, user is presented with CAPTCHA consisting of two text words to interpret. Verifying system knows answer for one of the two words, while the other word comes from the old text book, which system can not read. This fact is never revealed to the user. He therefore must read both the words and enter them correctly to access the web content. As a result, each time he is solving a reCAPTCHA, he is helping the system to read and digitize books.

We take inspiration from the reCAPTCHA design and aim to solve the problem of image annotation and in doing so; we wish to improve the image search and retrieval.

3. iCAPTCHA: Overview

We present iCAPTCHA, a user friendly CAPTCHA design. Instead of annotating images fresh from start, we try to improve the default labels the images have got. That is we attempt to obtain the more proper labels (subcategories) for an image. For example, with our design, we improve the label from general category such as ‘apple’ to more specific as ‘apple fruit’. Our premise is based on the human ability to recognize images, label them and put them into proper categories. Figure 2 shows the overview of the scheme.

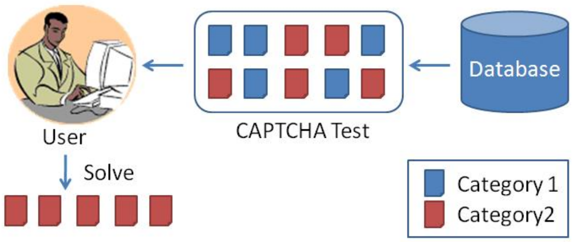


Figure 2: Overview of the iCAPTCHA scheme

We pick randomly a set of 12 images belonging to the two different image categories from the image database and present them as a CAPTCHA test. The task for the user is, to identify all the images belonging to one specified category. We can explicitly tell the category name or show a representative image belonging to the category. If the user correctly identifies all the images belonging to the desired category, he/she is considered to have 'passed' the test. On the hand, failure in recognizing the correct images will mean that user has failed the CAPTCHA test.

The use of images makes iCAPTCHA, language independent, less stressful and suitable for people of all ages and at any level of literacy.

3.1. iCAPTCHA: System Architecture

Before proceeding to the actual design it is essential that we understand the concept of 'tagged database' and 'test database'. Related to them are the concepts of 'category' and 'sub category'. We first briefly explain them.

1) Category and subcategory:

A category represents a short or ambiguous search query (e.g. 'apple') which when fired on popular search engine, normally results in images of many subcategories mixed together (e.g. 'apple fruit', 'apple logo', 'apple iPod' are subcategories for a category 'apple').

2) Test database:

CAPTCHA image test database is prepared by crawling the web for different image categories (as defined above). All resulting images are stored according to their respective categories (image queries) in a secure database at the server side.

3) Tagged database:

We recruit people or ask some trusted volunteers to describe (tag) the subcategories of few representative images, chosen at random from the test database. All labeled images are then stored according the described subcategories in a separate database called 'tagged database' at server side. The support of volunteers is needed only once at the beginning, the tagged database gets updated after each successful iCAPTCHA test.

The concept of 'test database' and 'tagged database' is analogous to the concept of test data and training data in the field of Content Based Image Retrieval.

4. iCAPTCHA: Proposed design

iCAPTCHA test comprises of 12 images. First, we fix one category say 'apple' from 'test database' and two related subcategories say 'apple fruit' and 'apple logo' from 'tagged database'. We retrieve few images at random say '*n*' (minimum 1 and maximum 11) from the 'tagged database' corresponding to the selected subcategories. Rest *'12-n'* images we select from 'test database' that belong to the selected category. We shuffle the selected images and present them to the user in a *2x6* matrix (two rows containing six images each) as shown in Figure 3.

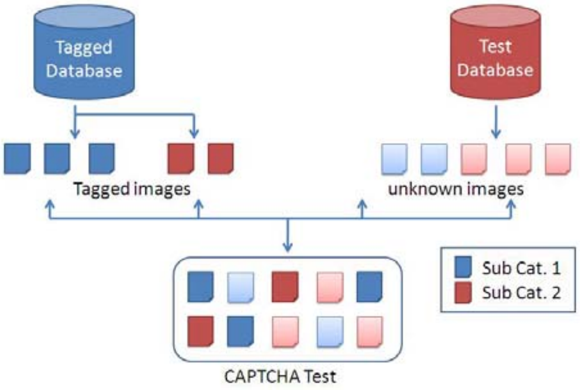


Figure 3: iCAPTCHA test generation process.

Users then must identify all the images that belong to the specified category say 'apple fruit'. Since user does not know which images are from 'tagged database' (i.e. already tagged) and which are not, the best option for him/her is to recognize and correctly select all the images of the required image category. The selected images would not be just from the 'tagged database' but could also be from the unlabelled 'test database'. Therefore, each time user is solving an iCAPTCHA, he/she is actually helping in labeling the images from 'test database' that are part of the given test. The evaluation process is shown in Figure 4.

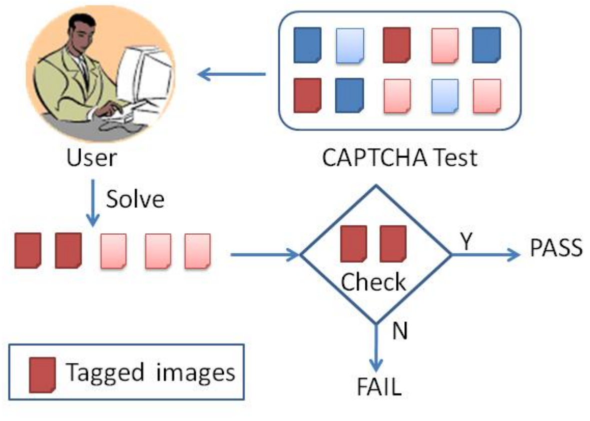


Figure 4: iCAPTCHA test evaluation

When the user submits his/her choice of images, we check whether his/her submitted choice of images contain the images of the specified category i.e. 'apple fruit' from the 'tagged database' (the images whose subcategory we already know) and does not contain any image from the other subcategory ('apple logo') from the 'tagged database'. If the answer is 'yes' then user has successfully passed the CAPTCHA test.

5. iCAPTCHA: Implementation

A working prototype of iCAPTCHA is created in Adobe Flash with PHP at the back end. MySQL is used for data storage.

In the prototype, the desired sub category is specified in words. For example, Figure 5 shows a sample iCAPTCHA test where user is asked to identify all images of 'apple fruit'.

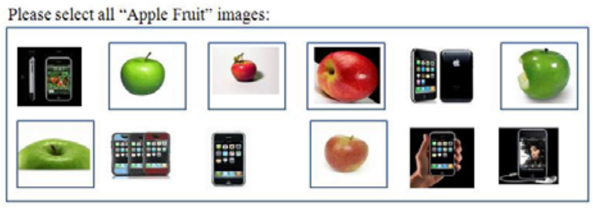

Please select all "Apple Fruit" images:

Figure 5: iCAPTCHA prototype 1: Identify all 'apple fruit 'images

As we can see in the Figure 5 that user has correctly selected all the 'apple fruit' images. But how we know it? Consider that, the first four images (first two images from each row) are from the 'tagged database' and rest eight images are the from 'test database'. For evaluation, we therefore check whether user has selected two correct images (the second image from first row and first image in the second row) and he has not selected the two wrong images (first image from the first row and second image from the second row). These four images are from 'tagged database' whose sub category we already know. Since user has correctly done that, he/she has successfully passed the test. Note that, in the process we also acquired the knowledge about the sub categories for the rest eight images (which are from 'test database'). That is, we came to know the images that belong to the specified category (i.e. third, fourth and sixth image from first row and fourth image from the second row are also images of 'apple fruit'. See Figure 5)

6. Security: Attacking iCAPTCHA

Attacking iCAPTCHA is difficult as computer programs are not yet advanced to automatically detect and label images in particular categories. An alternate attack can be by storing and searching for the images in Google image search engines. However, Google image search engine pages are dynamic in nature, which means the image that exist and ranked today may not be ranked in the same manner tomorrow. WE further take necessary measure such as no two iCAPTCHA tests are similar in nature both in terms of the kind of images that it has and to whom it is given. As a result, attacker, same as user will receive a random iCAPTCHA test each time that has not completely similar to the tests he/she solved before. We recommend that large image database should be constructed from Google image search with large number of categories to avoid any database attacks.

7. Usability study

To test the liability of the proposed design, we conducted a preliminary lab study with eight participants. All the participants were from university campus with their age in the range of 22 to 28. Two participants were female while rest six participants were male. We fixed five sample categories, those are: Apple, Cricket, Sachin tendulkar, Amrita and Rahul (with which all users were familiar with). Task for each of the participants were to solve five iCAPTCHA tests. All the participants successfully completed all the five tests. Early feedbacks were extremely positive with most of them reporting satisfaction with the proposed approach and design. We know the numbers are not be satisfactory in terms of the population they represent, therefore, as a future work, we are in the process of conducting a large scale field study with the diverse population.

8. Conclusion

In this paper, we described a novel CAPTCHA design, based on human ability to recognize images, label them and put them into proper categories. Benefit of our approach is getting the work of categorization and image annotation at virtually no cost. However, in doing so, we specially had taken care that the basic principles of CAPTCHAs like robustness and usability will not get affected. As a future work we are planning to launch a open source plug in of our proposed CAPTCHA design, and conduct a large scale field study.

9. References

[1] http://www.images.google.com

[2] http://images.search.yahoo.com/

[3] Carson, C., and Ogle, V.E. Storage and Retrieval of feature data for vey large online Image collection. *IEEE Computer Society of the Technical Committee on Data Engineering,* 1996, Vol. 19 No 4.

[4] White, R.W., Kules, B., Drucker, S.M., and schraefel, m.c. (2006). *Supporting Exploratory Search*, Introduction to Special Section of Communications of the ACM, Vol. 49, Issue 4, (2006), pp. 36-39.

[5] Datta, Ritendra, Dhiraj Joshi, Jia Li, James Z. Wang (2008). Image Retrieval: Ideas, Influences, and Trends of the New Age. *ACM Computing Surveys 2008*.

[6] http://www.filckr.com/

[7] https://www.mturk.com/mturk/welcome

[8] http://www.gwap.com/

[9] von Ahn, L., Blum M., and Langford J. Telling humans and Computers Apart Automatically. *CACM 47*, 2(1992).

[10] reCaptcha. http://recaptcha.net/.

[11] Slashdot. http://slashdot.com/.

[12] Yan J., and El Ahmad A. S. Breaking Visual CAPTCHAs with Naïve Pattern Recognition Algorithms, in *Proc. of 23rd Annual Computer Security Applications Conference (ACSAC'07).* FL, USA, Dec 2007. IEEE Computer society. pp 279-291.

[13] Yan J., and El Ahmad A. S. *A Low-cost Attack on a Microsoft CAPTCHA*, School of Computing Science Technical Report, Newcastle University, England. Feb, 2008.

[14] Mori G., and Malik J. Recognizing Objects in
Adversarial Clutter: Breaking a Visual CAPTCHA, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*, Vol 1, June 2003, pp.134-141.

[15] Chellapilla K., Larson K., Simard P., and Czerwinski, Designing Human friendly human interaction proofs" *ACM CHI 2005*.

[16] Chew M., Tygar J., Image Recognition CAPTCHAs. ISC 2004: 268-279

[17] J Elson, JR Douceur, J Howell and J Saul. Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. *Proceedings of the 14th ACM conference on Computer and communications security (CCS), 2007*.

[18] http://hotcaptcha.com/