# Utilizing the Power of Human Cycles

**Manuel Blum, PI**
**Luis von Ahn, co-PI**

*Construction of the Empire State Building: 7 million human-hours. The Panama Canal: 20 million human-hours. Estimated number of human-hours spent playing solitaire around the world in one year: 9 billion.*

## 1. Overview

There are many things that computers cannot yet do that people do easily; we want to make the most of this situation. This proposal is about finding clever ways to utilize human processing power, or "human cycles."

As an example, consider our previous work on the ESP Game (www.espgame.org). ESP is a seductive online game: many people play over 40 hours a week. In the process, they provide meaningful, accurate labels for images on the Web. These labels can be used, among other things, to improve the accuracy of image search and help Web browsers block undesirable content. This approach to labeling images is radically novel: rather than using computer vision techniques that don't work well enough, we constructively channel people to do the work in exchange for entertainment. The number of human-hours spent playing games is such that even ultra large-scale problems can be solved by having people play games online. In a few months, the ESP Game has collected over 8 million image labels, creating the largest collection of manually-labeled arbitrary images from the Web. If our game is deployed at a popular gaming site like *MSN Games* or *Yahoo! Games*, all images on the Web can be labeled in a matter of weeks. Because of this, the ESP Game will be hosted by a major Internet company to index all images on the Web starting in late 2005.

We propose to explore other clever techniques for utilizing human cycles constructively. More specifically, we propose the following:

- **Locating specific objects in images.** While the ESP Game can be used to determine whether an image contains a horse, for instance, it cannot be used to determine *where* in the image the horse is located. Such location information can be extremely useful for computer vision algorithms since one of the major stumbling blocks of computer vision techniques today is the lack of training data. We propose to build a new game called "Peekaboom" that will obtain information about the location and shape of the objects in images. *We propose to use the data obtained from this game to build the most comprehensive dataset for training computer vision algorithms and distribute the dataset freely. We believe computer vision researchers will use such a dataset to build the most accurate recognition programs for a wide variety of objects in the real world.*

- **Improving Web browsing for visually impaired individuals.** Visually impaired individuals surf the Web using "screen readers" (programs that read the contents of the screen out loud). Since today's screen readers can only read text from Web pages, the Web still presents a major accessibility problem because there are millions of images on the Web, most of which have no appropriate captions and therefore are completely obscured for those who rely on screen readers. One possible solution to this problem is the ESP Game. If all images on the Web are labeled by the game, screen readers could read the labels out loud. This, however, can be improved. The ESP Game produces labels, not explanatory sentences. While keyword labels are perfect for certain applications such as image search, accessibility would benefit more from explanatory

sentences. *We propose to build a new game whose output is explanatory sentences of images. Using the output of the game, we propose to build a prototype of a system that can improve the accessibility of the Web for visually impaired individuals.* If this game is put on a major Internet portal, like the ESP Game will be, it can dramatically improve the accessibility of the Web.

- **Novel new CAPTCHAs based on the data collected using our games.** The output of our games can also be used to create CAPTCHAs (http://www.captcha.net), automated tests that humans can pass but that current computer programs cannot. (The CAPTCHA project is led by the PI and the co-PI and was the result of previous NSF support to us.) CAPTCHAs have many applications in practical security. The previous results of our work, for instance, are used by Yahoo! and other major Web portals to ensure that only humans obtain free email accounts. Recently, algorithms have been developed to defeat the majority of CAPTCHAs currently in use. *We propose novel constructions of CAPTCHAs based on The ESP Game and Peekaboom that are significantly more resilient against attacks by Artificial Intelligence techniques.*

- **Other ideas to utilize human cycles.** Many large-scale open problems could be solved by channeling human brain power in this unique way. Each problem, however, requires the careful design of a game developed to be enjoyable and, at the same time, guarantee that game-play correctly solves instances of the problem. Designing such games is an art, much like designing algorithms: the game needs to be proven correct and enjoyable; its efficiency (how fast instances of the problem are solved by players) must be analyzed, and more efficient games can supersede less efficient ones. *We propose to more deeply explore the creation of games that solve open problems in Artificial Intelligence such as monitoring security cameras, collecting "common sense" knowledge, and judging the quality of images.*

**Relation to the Open Mind Initiative**

The proposed work is similar in spirit to the Open Mind Initiative (e.g., [13]), a worldwide effort to develop "intelligent" software. Open Mind collects information from regular Internet users (referred to as "netizens") and feeds it to machine learning algorithms. Volunteers participate by answering questions and teaching concepts to computer programs. Our work is similar to Open Mind in that we plan to use regular people on the Internet. However, the difference is that we put greater emphasis on our methods being fun due to the scale of the problems that we want to solve. We don't expect *volunteers* to label all images on the Web for us or to create millions of data points: we expect all images to be labeled because people enjoy playing our games.

**Intellectual Merit**

This project is part of a novel approach to collecting data—pioneered by our research team—that involves channeling humans to solve hard AI problems in the form of entertainment. This type of collaborative system has the potential to greatly influence the future of AI as well as all human-computer interaction. We envision a future in which humans and computers form a symbiosis: humans solve problems (in the form of entertainment) that computers cannot yet solve, with the results being used to improve computational techniques. Furthermore, our research combines ideas from AI and HCI to solve real-world problems and inspire deeper research.

**Broader Impact**

The very goal of this research is to solve important open problems that impact computer users on a daily basis. Our initial work in this area has captured the imagination of leading scientists, executives, and reporters worldwide, earning coverage in major news outlets (e.g., CNN, BBC, Associated Press, The New York Times, USA Today, etc.). Our work epitomizes the practical application of esoteric computer science theory. Moreover, this technique of research can be seen as the ultimate equalizer: everyone can participate in advancing knowledge, regardless of gender, ethnicity, geography, et cetera.

The proposed activities also integrate research and education by training undergraduate and graduate students in research. Because of the nature of our research, our projects are some of the most accessible and popular at Carnegie Mellon for undergraduates students interested in collaborating on research projects. Among the many undergraduate students engaged in this research are Roy Liu (he will be working on the game for locating objects in images next year) and Shiry Ginosar (she will be working on improving accessibility of the Web next year). Our work will be broadly disseminated through conference papers, journal publications, invited talks, and almost certainly through popular news outlets. Furthermore, the final results of our research have the potential to improve accessibility for visually impaired individuals, as well as to create valuable training datasets for public use.

This work provides a unique opportunity to engage the public in some of the deep and far-reaching issues in Artificial Intelligence and computer science in general. The ESP Game, for instance, has served to educate its players—most of them non-experts in computer science—about some of the open problems in Artificial Intelligence. It comes as a surprise to many that computers cannot yet recognize objects in images. We put strong emphasis on educating our players about *why* they are playing.

## 2. Our Previous Work: The ESP Game

To begin the description of our ideas, we first describe our previous work: The ESP Game [4]. The game is based on the following line of thought.

> Images on the Web present a major technological challenge. There are millions of them, there are no guidelines about providing appropriate textual descriptions for them, and computer vision hasn't yet produced a program that can determine their contents in a widely useful way. However, accurate descriptions of images are required by several applications like image search engines. Current techniques to categorize images for these applications are insufficient in many ways, mostly because they assume that the contents of images on the Web are related to the text appearing in the page. Such techniques do not always work because the text adjacent to the images is often scarce, and can be misleading or hard to process.
>
> The only method currently available for obtaining precise image descriptions is having humans label images by hand, which is tedious and thus extremely costly. But, imagine if people labeled images for amusement, because they delight in doing so. What if the experience was enjoyable?

The ESP Game is a collaborative system in the form of a game with a unique property: the people who play the game label images, and the labels are meaningful even if individual players are not trying to provide useful labels.

Our goal for this game was ambitious: to label the majority of images on the World Wide Web. If our game is deployed at a popular gaming site like *MSN Games* or *Yahoo! Games* and if people play it on average as much as other online games, we estimate that all images on the Web

can be properly labeled in a matter of weeks. In fact, 5,000 people playing the game simultaneously can label all images indexed by Google in 30 days. The striking fact is that 5,000 is not a very large number: on a recent weekday afternoon, we found 107,000 people playing in *Yahoo! Games*, 115,000 in MSN's *The Zone*, and 121,000 in *Pogo.com*. A typical game on these sites averages well over 5,000 people playing at any one time. Because of this, the ESP Game will soon be hosted by a major Internet company to index images on the Web starting late 2005.

**General Description of The ESP Game**

The current version of the ESP Game is implemented as a Java applet and can be played at http://www.espgame.org. It is a two-player game meant to be played online by a large number of pairs at once. In each session of the game, players are assigned partners from among all the people playing the game. Players are not told who their partners are, nor are they allowed to communicate with their partners. The only thing partners have in common is an image they can both see.

From the player's perspective, the goal of the ESP Game is to guess what words their partner is typing for each image. Once both players have typed the same text string, they move on to the next image (both players don't have to type the string *at the same time,* but each must type the same string at some point while the image is on the screen). We call the process of typing the same string "agreeing on an image" (see Figure 1).

Player 1 guesses: purse
Player 1 guesses: bag
Player 1 guesses: brown

Success! Agreement on "purse"

Player 2 guesses: handbag

Player 2 guesses: purse
Success! Agreement on "purse"

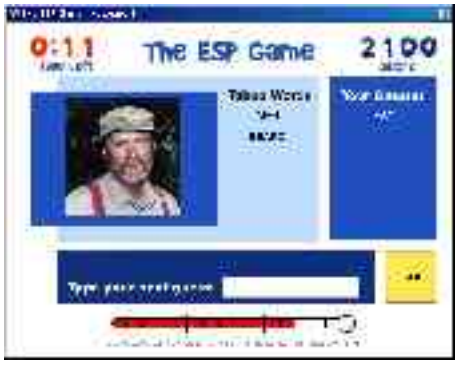


**Figure 1. Left: Partners agreeing on an image. Neither of them can see the other's guesses. Right: The ESP Game. Players try to "agree" on as many images as they can in 2.5 minutes. The thermometer at the bottom measures how many images the partners have agreed on.**

Partners strive to agree on as many images as they can in 2.5 minutes. Every time two partners agree on an image, they receive a certain number of points. If they agree on 15 images in a 2.5-minute time period, they receive additional bonus points. The thermometer at the bottom of the screen (see Figure 1) indicates the number of images that the partners have agreed on. By providing players with points for each image and bonus points for completing a set of images, we reinforce their incremental success in the game and thus encourage them to play multiple games. Players can also choose to pass, or opt out, on difficult images by clicking the pass button, which generates a message on the partner's screen. A pair cannot pass, or opt out, on an image until both have hit the pass button.

Since the players can't communicate and don't know anything about each other, the easiest way for both to agree is by typing something related to the common image. Notice, however, that the game doesn't ask the players to describe the image: all they are told is to "think like your

partner" and type the same string (thus the name "ESP"). *The string upon which the two players agree is an excellent label for the image.*

**Taboo Words**

A key element of the game, instrumental to generating multiple good labels, is the use of "taboo" words associated with each image, or words that the players are not allowed to enter as a guess. These words are related to the image and make the game more challenging by restricting the possible guesses that the partners can agree on. Imagine if the taboo words for the image in Figure 1 (left) were "purse", "bag", and "brown"; how would you agree on that image?

Taboo words are obtained from the game itself. The first time an image is used in the game, it has no taboo words. If the image is labeled, it receives one taboo word: the word that resulted from the agreement. The next time the image is labeled, it will have two taboo words, and so on.

Players are not allowed to type an image's taboo words, nor can they type singulars, plurals, or phrases containing the taboo words. The rationale behind taboo words is that often the initial labels agreed upon for an image are the most general ones (like "man" or "picture"), and by ruling those out the players will enter guesses that are more specific. Additionally, taboo words guarantee that each image will get many *different* labels associated with it.

**Quality of the Labels**

The labels generated by the game are of very high quality. In Figure 2, for instance, we present 14 images chosen at random from the set of images having the word "car" associated to them by the ESP Game. All of these images contain a car in some form or another.

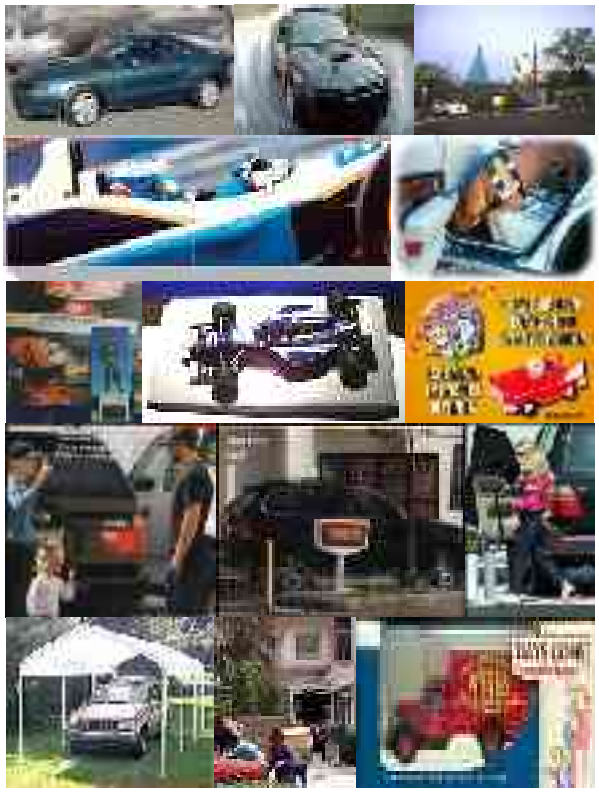

**Figure 2. Fourteen random images that had the label "car" associated to them by the ESP Game (some have been slightly cropped to fit the page better).**

# 3. Proposed Activities

Based on the tremendous success of the ESP Game, we propose to further explore clever techniques for utilizing human cycles. In this Section we delineate our specific goals.

## 3.1 Locating Specific Objects in Images

The ESP Game can be used to determine whether an image contains a horse, for instance, but cannot be used to determine *where* in the image the horse is located. Such location information can be very useful for computer vision algorithms since one of today's major stumbling blocks in advancing computer vision techniques is the lack of training data. We propose to build a game that will obtain information about the location and shape of the objects in the image.

There has been considerable work in computer vision related to automatically labeling images, and the most successful approaches *learn* from large databases of annotated images. Annotations typically refer to the contents of the image, and are fairly specific and comprehensive, and building these kinds of databases is expensive. Methods such as [5] cluster image representations and annotations to produce a joint distribution linking images and words. These methods can predict words for a given image by computing the words that have a high posterior probability given the image. Other methods attempt to combine large semantic text models with annotated image structures [7].

Though impressive, such algorithms based on learning don't work very well in general settings and work only marginally well in restricted settings. For example, the work described in [7] only gave reasonable results for 80 out of their 371 vocabulary words (their evaluation consisted of searching for images using the vocabulary words, and only 80 of the words resulted in reasonable images). A major stumbling block for these approaches is the lack of training data: algorithms are usually trained with only hundreds or perhaps thousands of images mainly because preparing training data is tedious, costly, and time-consuming. The new game we propose to implement, Peekaboom, is a clever method for collecting vast amounts of training data for computer vision algorithms. The data is collected by people playing an entertaining game over the Internet. The game is enjoyable and guarantees that the information that it collects is accurate.

**Rules of the Game: Peeking and Booming**

Peekaboom is played by two partners and is meant to be played online by a large number of pairs at once. Partners are randomly assigned from among all the people playing the game. Players are not told who their partners are, nor are they allowed to communicate with their partners.

The two players take turns "peeking" and "booming." While one player is peeking, the other is booming. The booming player (Boom) gets an image along with a word related to the image, and the peeking player (Peek) gets no image (see Figure 3). Booming consists of clicking parts of the image; when Boom clicks a part of the image, it is revealed to Peek. The object of the game is for Peek to type the word associated to the image—from their perspective, the game consists of a slowly revealing image, which has to be named. From Boom's perspective, the game consists of clicking on areas of the image so that Peek can guess the word associated to it. Once Peek guesses the correct word, the two partners move on to the next image and switch roles.

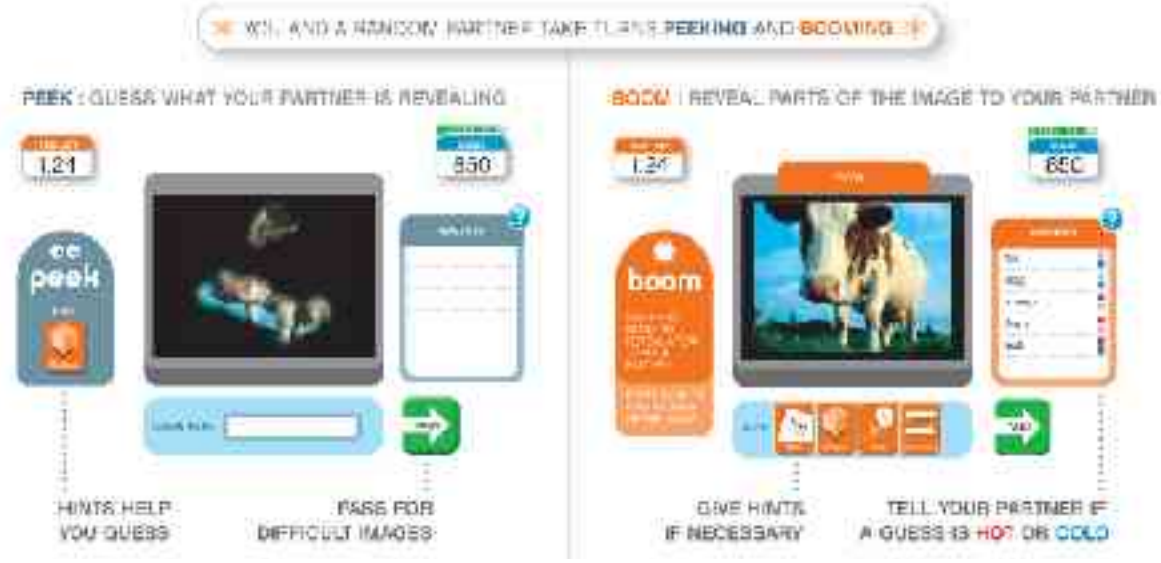


**Figure 3. Basic rules of Peekaboom. Peeking consists of guessing what your partner is revealing. Booming consists of revealing parts of the image so that your partner types the word displayed at the top of the image.**

Partners strive to go through on as many images as they can in a certain amount of time. Every time Peek guesses the right word, both players get a certain number of points. Players can also choose to pass, or opt out, on difficult images. If a player clicks the pass button, a message is generated on their partner's screen; a pair cannot pass on an image until both have hit the pass button. Boom can see all of Peek's guesses and can indicate whether they are "hot" or "cold." (See Figure 4.)

**Figure 4. Boom can indicate whether Peek's guesses are hot or cold.**

Since the players can't communicate and don't know anything about each other, the easiest way for Boom to help Peek is by revealing precisely the areas in the image that are associated to the word.

**Pings**

In order to help Peek, Boom can also "ping," or point to, an area of the image. Pings tell Peek that the word is related to that particular area. In Figure 5, for instance, Boom has already

revealed enough of the image for Peek to see that it is an elephant, but Boom still needs to point to the trunk, since the associated word is “trunk” and not “elephant.”

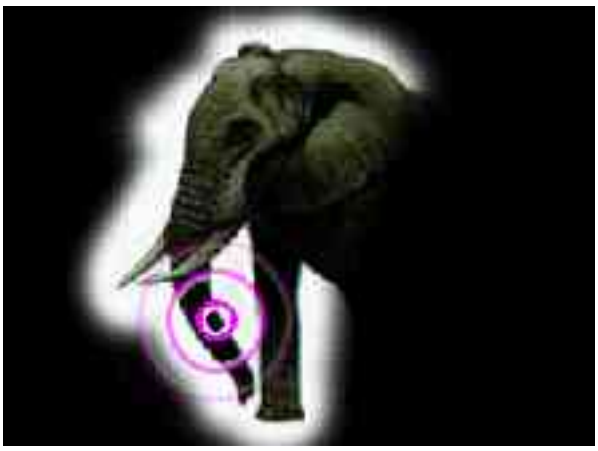

**Figure 5. Pinging. Boom can point to an area of the image that has been revealed already. In this image, the word related to the image is “trunk,” so Boom must “ping” it to help Peek.**

## Hints

Boom can also help Peek by giving hints about the word in question: does it refer to an object in the image (noun), to an object related to the image (noun related), to a verb, or to text appearing in the image? Figure 6 shows the different types of hints that boom can give peek.


**Figure 6. Hints in Peekaboom.**

## Input/Output Behavior

By recording all the actions of people playing Peekaboom, we can collect valuable information about the contents of the images and how people decipher them. Given an image and a word associated to it, Peekaboom collects information about which part of the image relates to the word by observing which part of the image Boom reveals to Peek. To get more accurate results, we present an image to multiple pairs of players and combine the data in an intelligent manner. Furthermore, by recording all the places that Boom “pings,” we can get even more information about the location of objects in images. Treating Peekaboom as an algorithm, on input a pair (image, word), Peekaboom outputs an area of the image that is related to the word. For example, in Figure 7, given the image and the word “butterfly,” Peekaboom would output the area of the image that contains the butterfly, along with points inside the butterfly (that come from pings). Using the hints, Peekaboom also outputs whether the word relates to an object in the image, to an object not in the image, to text in the image, or to a verb.


**Figure 7. Peekaboom outputs an area of the image that contains the object in question.**

Combining Peekaboom with the ESP Game gives a full system for annotating arbitrary images for computer vision purposes. Upon being input an image, the ESP Game gives a set of words associated to the image. For each word, Peekaboom gives information about what the word refers to in the image. Such information is useful in multiple ways. First, Peekaboom gives an area of the image related to the word in question, which allows computer vision algorithms to focus training on the relevant parts of the image. Second, Peekaboom gives points inside the object in question (from "pings"), which can also be used for training purposes. Third, Peekaboom can be used to obtain context information: *"what is enough to determine that an object really is that object?"* Such content information has not been thoroughly studied, but Peekaboom will produce vast amounts of data to do so. Fourth, Peekaboom can provide training data for OCR programs. Part of the output of Peekaboom will be a large number of images with text in them, along with an ASCII translation of that text and an area containing the text. These data could be used for training/testing purposes to improve algorithms that read arbitrary text from the Web. Being able to read arbitrary text in images from the Web would be useful for accessibility reasons.

**Validation**

We propose to validate our results from Peekaboom in multiple ways. First, we need to prove that people will enjoy playing Peekaboom. Second, we need to prove that the data produced by Peekaboom is accurate and useful.

To evaluate if people will enjoy playing Peekaboom, we will simply record usage statistics from arbitrary people playing our game online. To present evidence that the data produced using the game is indeed accurate and useful, we plan a user study in which participants evaluate whether the regions in the images are indeed accurate as locations of objects in the images. We also plan to use our data to train computer vision algorithms developed by the Carnegie Mellon computer vision group. We have been in close contact with Alexei Efros, assistant professor at Carnegie Mellon, who is eager to start evaluating our data.

## 3.2 Improving Web Browsing for Visually Impaired Individuals

Visually impaired individuals surf the Web using "screen readers" (programs that read the contents of the screen out loud). Since today's screen readers can only read text from Web pages, the Web still presents a major accessibility problem because most of the millions of images on the Web have no appropriate captions and therefore are completely obscured for those who rely on screen readers. We propose to build a new game whose output is explanatory sentences for images. Using the output of the game, we propose to build a prototype of a system that can

improve the accessibility of the Web for visually impaired individuals. This system has the potential to dramatically improve the accessibility of the Web.

The proposed game is based on the same setting as Peekaboom: two random partners must help each other to obtain points. As with Peekaboom, both partners switch roles every time, between "selecting" and "describing." While one of the partners is selecting, the other is describing. The Describer gets a single image M, while the Selector gets a grid of images, one of which is M. The goal of the game is for the Describer to make the Selector select M. Communication is one-sided in that the Describer can describe M, but the Selector cannot say anything back. (See Figure 8.) By observing the sentences that the Describer enters, we propose to collect natural language descriptions of arbitrary images on the Web.

Similar to the ESP Game, several strategies can be used to ensure that players do no collude to poison the data collected by this game, such as comparing various pairs' results [4].

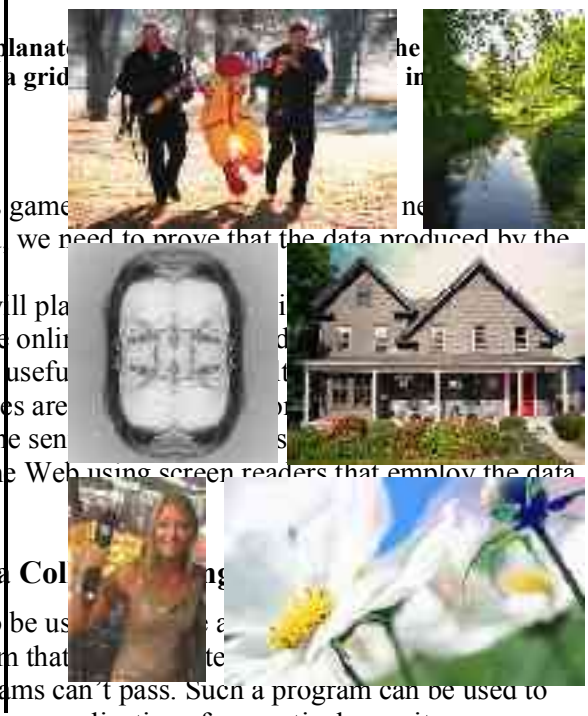


**Figure 8. A two-player game that collects explanat he the players gets an image M and the other gets a grid in**

## Validation

We ults from this game ne that pe game. Second, we need to prove that the data produced by the game is

As e if people will pla statistic ying our game onli produc accurate and usefu which participants evaluate whether the sentences are user study, participants will rate how accurate the sen another study, we will ask participants to surf the Web using screen readers that employ the data p

## 3.3 Novel New CAPTCHAs Based on Data Col ng

The data collected using our games can also be us a CAPTCHAs [1, 2, 3]. A CAPTCHA is a program that te humans can pass, but 2) current computer programs can't pass. Such a program can be used to differentiate humans from computers and has many applications for practical security. CAPTCHA stands for "Completely Automated Public Turing Test to Tell Computers and Humans Apart." The T for "Turing Test to Tell" is because CAPTCHAs are like Turing Tests [14]. In the original Turing Test, a human judge was allowed to ask a series of questions to two players, one of which was a computer and the other a human. Both players tried to act human, and the judge had to determine which was actually a computer. CAPTCHAs are similar to the Turing Test in that they distinguish humans from computers, but they differ in that the judge is now a computer. A CAPTCHA is an Automated Turing Test.

CAPTCHAs also differ from the original Turing Test in that they can be based on a variety of sensory abilities. The original Turing Test was conversational—the judge was only allowed to ask questions over a text terminal. In the case of a CAPTCHA, the computer judge can ask any question that can go over a network. There are multiple examples of CAPTCHAs, although the most well-known is the one used by Yahoo! and most other major Web portals that consists of reading distorted text images.

Some applications of CAPTCHAs include:

- **Online Polls.** In November 1999, slashdot.org released an online poll asking which graduate school was the best in computer science (a dangerous question to ask over the Web!). As is the case with most online polls, IP addresses of voters were recorded in order to prevent single users from voting more than once. However, students at Carnegie Mellon found a way to stuff the ballot box by using programs that voted for Carnegie Mellon thousands of times. Carnegie Mellon's score started growing rapidly. The next day, students at MIT wrote their own voting program and the poll became a contest between voting "bots." MIT finished with 21,156 votes, Carnegie Mellon with 21,032 and every other school with less than 1,000. Can the results of any online poll be trusted? Not unless the poll verifies that only humans can vote.
- **Free Email Services.** Several companies (Yahoo!, Microsoft, etc.) offer free email services, most of which suffer from a specific type of attack: "bots" that sign up for thousands of email accounts every minute. This problem can be solved by requiring users to prove they are human before they can get a free email account. Yahoo!, for instance, uses a CAPTCHA of our design to prevent bots from registering for accounts. Their CAPTCHA asks users to read a distorted string of letters such as the one in Figure 9.
- **Search Engine Bots.** Some Web sites don't want to be indexed by search engines. There is an HTML tag to prevent search engine bots from reading Web pages, but the tag doesn't guarantee that bots won't read the pages; it only serves to say "no bots, please." Since search engine bots usually belong to large companies, they respect Web pages that don't want to allow them in. However, in order to truly *guarantee* that bots won't enter a Web site, CAPTCHAs are needed.
- **Worms and Spam.** CAPTCHAs also offer a plausible solution against email worms and spam: only accept an email if you know there is a human behind the other computer. A few companies, such as *www.spamarrest.com* are already marketing this idea.
- **Preventing Dictionary Attacks.** Pinkas and Sander [11] have suggested using CAPTCHAs to prevent dictionary attacks in password systems. The idea is simple: prevent a computer from being able to iterate through the entire space of passwords by requiring a human to type the passwords.

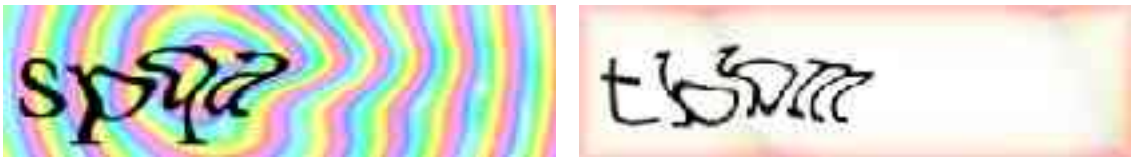

**Figure 9. Examples of distorted-text CAPTCHAs taken from our work (http://www.captcha.net).**

A primary goal of the CAPTCHA project is to serve as a challenge to the Artificial Intelligence community. We believe that having a well-specified set of goals will contribute greatly to the advancement of the field. A good example of this process is the recent progress in reading distorted text images driven by the CAPTCHA in use at Yahoo!. In response to the challenge provided by this test, Malik and Mori [10] developed a program which can pass the

test with probability roughly 0.8. Malik and Mori's algorithm represents significant progress in the general area of text recognition; it is encouraging to see such progress.

The CAPTCHAs currently in use on the Web are all variations of the following idea: generate a random string of characters or a word, render the string into a randomly distorted image and ask the user which characters appear in the image. Due to the fact that most of these CAPTCHAs generate text distorted in similar ways over and over, it was possible after a few years of work [1, 10] to write computer programs that can read the CAPTCHA text with high enough success probability to obtain accounts relatively quickly. We propose to build new CAPTCHAs that are significantly more resilient to attacks.

Given the way we plan to build our new CAPTCHAs, they will have the desirable property that any adversary that can defeat them with reasonable success will be truly able to solve open problems in AI. These CAPTCHAs will then be a true win-win situation: either they remain secure or they are broken and a new Artificial Intelligence problem is solved.

**ESP-PIX**

ESP-PIX works by asking the user to identify an object that is common to a series of images presented. All of the images presented are pictures of concrete objects (a horse, a table, a house, a flower, etc). The program picks an object at random, finds four images of that object, distorts them at random, presents them to the user, then asks the question "what are these pictures of?"

The images will come directly from the ESP Game, so ESP-PIX can have an immensely large database of labeled images. If the ESP Game becomes popular enough to label all images on the Web, ESP-PIX would have an image database of hundreds of millions of images, making the task of writing a program that can pass these tests very difficult. And, even if the adversary were to build a version of the ESP Game and label all images on the Web (this task is not trivial and would require the adversary to advertise its version of the Game, thus alerting us of its plan), we will distort the images before presenting them to the user, so that the adversary will have difficulty finding the right image in their labeled database.

The idea for this type of CAPTCHA was first suggested by us in [1, 3], but there was no large database of accurately labeled images from which to build it. After having collected 8 million labels for random images from the Web with the ESP Game, we are now ready to build and polish such a CAPTCHA. Our goal is to deploy this CAPTCHA on real-world Web sites and eventually make it as popular as the distorted-text CAPTCHAs currently in use.

**ESP-TEXT**

Consider the two images in Figure 10. Both images obtained the label "dog" through the ESP Game. One of them contains a dog and the other contains the word "dog" as text inside it. We propose to modify the current ESP Game so that we can learn when a label refers to an object in the image or to a word that appears as text inside it. We propose to make players enter a word in quotes ("") whenever it refers to text that appears inside the image, for which they will get extra points. To prevent players from typing every word in quotes, we will give them some images for which we already know if the word should be in quotes or not—if they enter quotes when they shouldn't, we will penalize them.


**Figure 10. Both images obtained the word "dog" through the ESP Game.**

Similar to ESP-PIX, we can use data obtained by this variant of the ESP Game to create a CAPTCHA that is significantly harder to defeat. We call this CAPTCHA ESP-TEXT. The idea is to use images with text in them collected from the entire Web and ask users to type the text contained in the images (see Figure 11 for a sample of images of the word "dog" taken from the Web). To the users, this CAPTCHA would feel very similar to the CAPTCHAs currently in use, but to an adversary, it would be significantly harder to defeat, as the text will come from the entire Web and will not have any pre-specified patterns. Furthemore, this CAPTCHA will have the property that any adversary that can break it will be able to read text from random images on the Web.


**Figure 11. Images collected from the Web with the word dog in them.**

**Peekaboom CAPTCHA**

The output of Peekaboom can also be used to create a CAPTCHA. The CAPTCHA would use the images collected by Peekaboom and ask the user to click on a certain object (e.g., "click on the cat"). If Peekaboom becomes popular enough to annotate millions of images, this CAPTCHA would have an extremely large image database, making the task of writing a program that can pass these tests very difficult. And, even if the adversary were to build a version of Peekaboom and annotate all images on the Web (this task is not trivial and would require the adversary to advertise its version of the game, thus alerting us of its plan), we will distort the images before presenting them to the user, so that the adversary will have difficulty finding the right image in their labeled database. We propose to build and test this new CAPTCHA, which will add to the library of already existing CAPTCHAs. We stress that having CAPTCHAs based on a variety of problems is useful because they serve as challenges to the AI community.

**Validation**

We propose to validate our new CAPTCHAs in two ways. One, every CAPTCHA developed must be human tested: can humans pass the tests with high enough accuracy? How much time on average does it take for humans to pass them? Second, to show that computer programs cannot pass the tests generated by our new CAPTCHAs, we plan to make them challenges to the AI community (as we did with our previous work on distorted-text CAPTCHAs).

### 3.4 Other Ideas to Utilize Human Cycles

In addition to the systems described so far, we propose to explore the feasibility of solving other problems by cleverly utilizing human cycles. We mention the following:

- **Judging the Quality of Media.** Humans are easily able to determining whether an image or a sound-clip is "good": is it not blurry or noisy, is it pleasing, the objects in the image easy to see, etc. Such information can be useful for search engines, which should return high-quality results first. We plan to develop a game that enables us to determine the quality of an image, sound, or video clip.
- **Determining Similarity of Images.** Being able to decide whether two images are "similar" to each other has many applications. The system described in [8], for instance, assumes the existence of a large database of similarities between arbitrary images. Building such a large database is a difficult task, but turning the process into a game can solve the problem. We propose investigating the idea of turning this process into a game of "memory" (in which two players take turn flipping two cards and they get points if the two cards are of the same or similar objects).
- **Monitoring Security Cameras.** One of the main stumbling blocks for installing more security cameras around the world is that it is expensive to pay humans to watch the cameras 24 hours a day. What if people played a game that could alert somebody when illegal activity was going on?
- **Collecting "Common-Sense" Knowledge.** Multiple efforts throughout the world (e.g. [13]) have started collecting "common-sense" knowledge for Artificial Intelligence purposes. It is estimated that 80 million common sense "facts" are enough to create a system that is seemingly intelligent in multiple respects. The problem, of course, is that collecting 80 million common sense facts is not an easy task. We propose to turn this process into a fun multi-player game.

## 4. Timeline

Work on the proposed research will begin with Peekaboom, which we should be able to complete within 6-9 months. The first task is to build an application for it, which should consist of a game server and a game client, both implemented in Java. After completing a first rough version of the game, we plan to begin pilot testing it. We then will also work with a graphic designer to give it an appealing, easy-to-use look. Based on the user feedback and the graphic design, we plan to complete a final version of the game and release it on the Web. The game will be linked from the ESP Game site (which receives high traffic). After full release of the game, we propose to promote it for several months and collect millions of fully annotated images for a training dataset to be used by computer vision. Researchers from the Carnegie Mellon computer vision group are eager to obtain such dataset, which will also be made available to the rest of the academic community.

Development of the game that will improve accessibility of the Web will begin after completion of Peekaboom and is expected to be completed within 6-9 months also. As with Peekaboom, the first task is to build an application for it, which should consist of a game server and a game client, both implemented in Java. After completing a first rough version of the game, we plan to begin pilot testing it. We then will also work with a graphic designer to give it an appealing, easy-to-use look. Based on the user feedback and the graphic design, we plan to complete a final version of the game and release it on the Web. The game will be linked from the ESP Game site (which receives high traffic). A full system that can obtain descriptions of images from the Web and report them to blind people using screen readers is expected within 12-18 months after that. Additional research will be pursued in tandem to these projects.

## 5. Results from Previous NSF Support

The PI, Manuel Blum, was a co-PI for NSF Grant Number 0085982, entitled ITR: Algorithms: From Theory to Application; the co-PI was a student supported by this grant. The grant supported the ALADDIN Center's research and educational activities on applied algorithms. These research activities were centered around a sequence of PROBlem-oriented Explorations (PROBEs). Each PROBE focused on a specific problem domain, and lasted for approximately one year. Its purpose was to find algorithmic solutions and appropriate mathematical formulations for that problem domain, and thus involved both domain experts and algorithm designers. As part of the CAPCHA PROBE, we discovered new techniques for restricting accesses so only humans, and not machines, can enter and register at certain cites. These techniques were subsequently adopted by Yahoo! and most other major Web portals. Publications [1, 2, 3, 4] all resulted from this award.

## References Cited


[1] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. The CAPTCHA Web Page: *http://www.captcha.net*. 2000.

[2] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. CAPTCHA: Using Hard AI Problems For Security. In *Lecture Notes in Computer Science, Volume 2656,* as *Advances in Cryptology, Proceedings of Eurocrypt '03,* pages 294-311, 2003.

[3] Luis von Ahn, Manuel Blum, and John Langford. Telling Humans and Computers Apart (Automatically) or How Lazy Cryptographers do AI. In *Communications of the ACM,* February 2004.

[4] Luis von Ahn and Laura Dabbish. Labeling Images with a Computer Game. In *Proceedings of CHI 2004*.

[5] K. Barnard and D. A. Forsyth. Learning the Semantics of Words and Pictures. *International Conference of Computer Vision*, 2001, pages 408-415.

[6] Columbia Center for Telecommunications Research. *webSEEK.http://*www.ctr.columbia.edu/webseek/

[7] P. Duygulu, K. Barnard, N. de Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. *7th European Conference on Computer Vision, 2002,* pages 97-112.

[8] Takeo Kanade and Shingo Uchihashi. User Powered Content-Free Approach to Image Retrieval. In *Proceedings of International Symposium on Digital Libraries and Knowledge Communities in Networked Information Society 2004 (DLKC04),* March 2004. pages 24-32.

[9] R. Lempel and A. Soffer. PicASHOW: Pictorial Authority Search by Hyperlinks on the Web. *WWW10*.

[10] Greg Mori and Jitendra Malik. Breaking a Visual CAPTCHA. Unpublished Manuscript, 2002. Available electronically: http://www.cs.berkeley.edu/~mori/gimpy/gimpy.pdf.

[11] Benny Pinkas and Tomas Sander. Securing Passwords Against Dictionary Attacks. In *Proceedings of the ACM Computer and Security Conference (CCS '02),* pages 161-170. ACM Press, November 2002.

[12] H. Schneiderman and T. Kanade. Object Detection Using the Statistics of Parts. *International Journal of Computer Vision,* 2002.

[13] D. G. Stork and C. P. Lam. Open Mind Animals: Ensuring the quality of data openly contributed over the World Wide Web. *AAAI Workshop on Learning with Imbalanced Data Sets*, 2000, pages 4-9.

[14] Alan M. Turing. Computing Machinery and Intelligence. In *Mind,* Vol. 59, No. 236, pp. 433-460. 1950.