Face Recognition CAPTCHA

Gaurav Goswami, Richa Singh, Mayank Vatsa
IIIT-Delhi
New Delhi, India
{gauravgs, rsingh, mayank}@iiitd.ac.in

Brian Powell, Afzel Noore
West Virginia University
Morgantown, WV, USA
{brian.powell, afzel.noore}@mail.wvu.edu

Abstract

CAPTCHA is one of the Turing tests used to classify human users and automated scripts. Existing CAPTCHAs, especially text-based CAPTCHAs, are used in many applications, however they pose challenges due to language dependency and high attack rates. In this paper, we propose a face recognition-based CAPTCHA as a potential solution. To solve the CAPTCHA, users must correctly find one pair of human face images, that belong to same subject, embedded in a complex background without selecting any non-face image or impostor pair. The proposed algorithm generates CAPTCHA that offer better human accuracy and lower attack rates compared to existing approaches.

1. Introduction

It is well established that humans are good at recognizing both familiar and unfamiliar faces [11], [14]. Studies have also shown that humans can recognize faces with variations in expression, illumination, and resolution. Compared to automatic algorithms, it is easier for humans to match face photographs that have artifacts such as noise and poor quality. This capability of human mind can be very useful in designing tests to distinguish between humans and automated systems.

Completely Automated Public Turing Test to Tell Computers and Humans Apart or CAPTCHA is designed to distinguish between genuine users and automated scripts [13]. CAPTCHAs are being used for several services including web and financial services, to provide security against malicious attacks[1]. Research in CAPTCHA has focused on developing tests that are easy for humans to solve and difficult for automated approaches. The most popular form of CAPTCHA today is text in the form of an image which has been distorted so that only a human can accurately identify the letters and numbers. Text in AltaVista CAPTCHA, one of the first text CAPTCHAs, was taken from an optical character recognition (OCR) manual with distortions known to reduce OCR accuracy incorporated [10]. GIMPY CAPTCHA, similar to the AltaVista CAPTCHA [2, 10], uses English dictionary words. In the ScatterType CAPTCHA, individual characters are segmented into pieces and then systematically scattered so that they are difficult to reassemble [3]. Megaupload CAPTCHA uses overlapping characters while MSN CAPTCHA introduces lines connecting individual characters.

Rather than designing tests to be unrecognizable via OCR, some CAPTCHAs have taken an approach of using handwritten text images already known to fail OCR. A database of text images from handwritten mail addresses that could not be detected automatically are used in such CAPTCHAs. When full city names are used, humans are able to identify the word 100% of the time but the computer success rate is about 9% [12]. Similarly, reCAPTCHA was designed using text images scanned from book digitization projects [15]. In reCAPTCHA, users are presented with two text images (one of a word that is unknown and one whose text has been previously determined) and asked to enter both the words. The previously-known word is used as the test while the currently-unknown word's results are stored to help identify that word for future use.

As an alternative to text, several CAPTCHA algorithms utilize image classification or recognition tasks as part of their test. One basic image-based CAPTCHA is ESP-PIX in which a collection of images are shown and the user has to select a description from a pre-defined list of categories [1, 9]. KittenAuth, another image CAPTCHA, poses images of cats to the user [16]. Asirra is similar to KittenAuth and uses a closed database to source the images [6]. A number of other CAPTCHAs rely upon composites of multiple embedded images rather than discrete images as with the previous models. Scene Tagging CAPTCHA requires identifying relationships and relative placement of different images [7]. MosaHIP requires dragging descriptors and dropping them on top of embedded images in a collage [4]. Re-

[1] It is important to safeguard the web-services against bots which can compete intensively for the server's resources hence denying service to genuine human users and consuming most of the bandwidth leading to the failure of the service.

978-1-4673-1228-8/12/$31.00 ©2012 IEEE

cently, a new design has been proposed that uses recognition of geometric patterns. The IMAGINATION CAPTCHA combines geometric shape recognition with categorization in a two-step process. Users have to first mark the center point of an embedded image and then select an appropriate category based on a predefined list to describe that image [5]. The results show human success rate of approximately 70% with a machine random guess rate of about 0.0005% [5]. Further, there are other CAPTCHAs that uses videos and/or audio based tests. However, they are not very popular primarily due to higher bandwidth requirements.

Many of the existing CAPTCHAs have limitations such as being difficult for human users to solve and having high automated attack rates. Text-based CAPTCHAs have major limitations due to language dependencies. Most of the text based CAPTCHAs are in English and might be difficult to understand and solve for non-English users. Further, some existing image-based CAPTCHAs demonstrate a common weakness - a small number of possible solutions for which *guessing* can have a high likelihood of success.

In order to make CAPTCHAs more secure, the complexity has increased and in some cases, humans find them difficult to solve. To overcome the limitations faced by current CAPTCHAs, this paper proposes a novel image CAPTCHA generation algorithm that uses *face recognition* as the test. While generating the CAPTCHA, the proposed algorithm leverages (1) the limitations of the state-of-art face recognition algorithms and (2) the capabilities of human mind to perform face recognition. On comparing with other existing CAPTCHAs, about 3000 volunteers suggested that the proposed CAPTCHA is easier and more intuitive to solve. The next section provides the details of the proposed algorithm and Section 3 presents the experimental results and key observations.

2. Generating Face Recognition CAPTCHA

As discussed earlier, the objective of a CAPTCHA is to present a *test* which is difficult for automatic algorithms but relatively easy and intuitive for humans. One such *Turing test* can be a face recognition test to distinguish automated algorithms and humans. Even after decades of research in face recognition, there are several challenges in designing effective and accurate automatic algorithms. Conversely, the human mind is very effective in recognizing faces even with complex background and partial/hidden features. The proposed algorithm utilizes these properties to create image CAPTCHAs. In other words, the proposed algorithm is based on optimizing sets of parameters on which standard face recognition algorithms fail but humans can succeed. The process of solving the proposed CAPTCHA is as follows:

- A CAPTCHA image containing human face images, cartoon face/non-face images, and some random object images with some level of distortion - all on a randomly generated background with random shapes and colors, is shown to the user.

- Every CAPTCHA has at least two genuine pairs of human faces along with some additional individual human faces. Out of the two genuine pairs of faces, the user has to select one genuine pair of human faces which belongs to the same person.

- As shown in Figure 1, the user has to mark the approximate center of two face images which he/she recognizes as a genuine pair. If these responses are correct, then the CAPTCHA is considered to be solved, otherwise not.

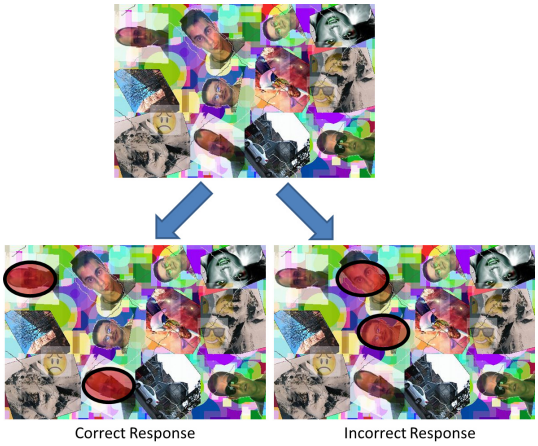

Figure 1. Illustrating the process of solving the face recognition CAPTCHA.

The proposed CAPTCHA generation can be represented as,

$$C = F(\mathbf{f}, \overline{\phi}). \quad (1)$$

Here, F is a function that uses face and non-face images, $\mathbf{f}$, along with parameter, $\overline{\phi}$, to create the CAPTCHA image C. The CAPTCHA image C is a combination of images and image processing operations that use the parameters defined in ϕ. These parameters control the tradeoff between human accuracy and machine accuracy. In the proposed approach, a gradient learning technique is used to learn the range of parameters for which human can solve the CAPTCHA but not the automatic algorithms. The design of the proposed algorithm can be divided into two stages: (1) estimating CAPTCHA parameters and (2) CAPTCHA generation using trained parameters.

Estimating CAPTCHA Parameters: Since CAPTCHA generation is dependent on several parameters, the training stage involves learning the useful sets of parameters. A set of parameters is considered useful if humans can successfully detect one of the genuine face pairs but the automatic

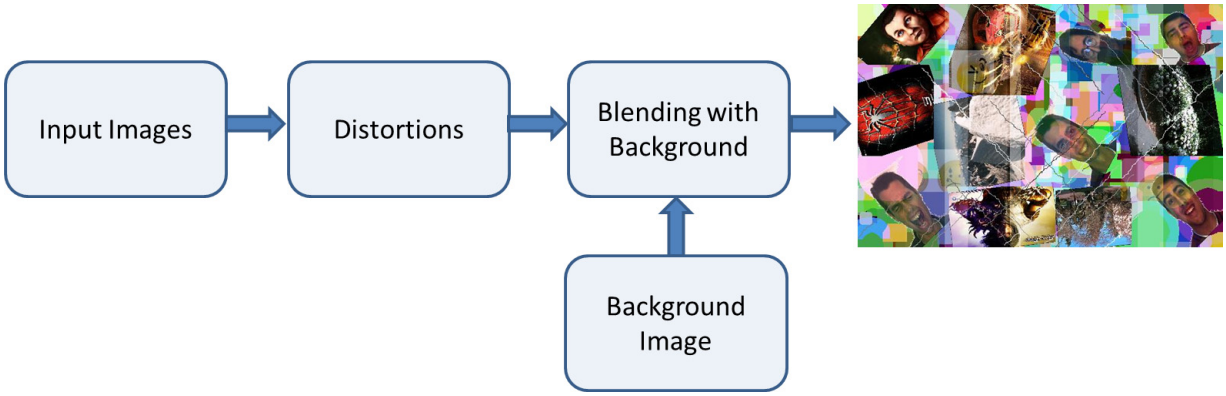

Figure 2. Illustrating the steps involved in the proposed CAPTCHA algorithm.

face recognition algorithm is unable to do so. Let a_h be the human response and a_m represents the automatic algorithm's response, the set of parameters is useful if

$$\overline{\phi}_u = Train_{(C=F(\mathbf{f},\overline{\phi}_i))}(a_h = 1, a_m = 0) \quad (2)$$

Here, $a_h = 1$ represents the correct human response to solve the CAPTCHA and $a_m = 0$ represents the incorrect response by the automatic algorithm. There can be several parameters, $\overline{\phi}_i$, however only those parameters, $\overline{\phi}_u$, are chosen that satisfy the condition given in Equation 2. Parameters associated with other conditions, i.e. $(a_h = 1, a_m = 1)$, $(a_h = 0, a_m = 1)$, and $(a_h = 0, a_m = 0)$ are not useful. Further, *Train* represents parameter learning in a gradient descent manner. For a given $\overline{\phi}_t$, let $E_{h,m}(\overline{\phi}_t)$ be the objective or error function that minimizes the error caused by four conditions associated with a_h and a_m, i.e.,

$$E_{h,m}(\overline{\phi}_t) = \begin{cases} 0 & if\ a_h = 1\ and\ a_m = 0 \\ 1 & \text{otherwise.} \end{cases} \quad (3)$$

In gradient descent, optimal parameters are obtained using Equations 4 and 5.

$$\bigtriangledown E_{h,m}(\overline{\phi}_t) = \frac{\partial E_{h,m}(\overline{\phi}_t)}{\partial \overline{\phi}} \quad (4)$$

$$\overline{\phi}_{t+1} = \overline{\phi}_t - \eta \bigtriangledown E_{h,m}(\overline{\phi}_t) \quad (5)$$

Here, $\bigtriangledown E_{h,m}(\overline{\phi}_t)$ represents the gradient of error function at the t^{th} learning iteration and η is the learning rate that controls the convergence of parameter learning. Since the outcome of gradient descent optimization is dependent on the initial values as well, we have made the initial parameter assignments using small scale experiment with humans and automatic algorithm. As shown in Equation 5, the learning process involves human response (h) as well as the response of automatic face recognition algorithm (m) and therefore converges to the case where humans can solve the CAPTCHA but not the automatic algorithm.

The proposed CAPTCHA design is a function of the following parameters (and related operations):

- The first parameter is the total number of images, both face and non face images, and is represented as n_{total}. The non-face images are images of cartoons, chosen specifically to generate false positives when used by an automatic algorithm.
- The number of face images in the CAPTCHA, represented as $n_{face} \geq 4$, is the second parameter. Out of n_{face} images, there are two genuine face pairs and the remaining are single images of different individuals. In a given CAPTCHA,

  $$n_{total} = n_{face} + n_{non-face} \quad (6)$$

  where $n_{non-face}$ is the number of non-face images. For a given CAPTCHA, we only need to define n_{total} and n_{face}.
- The third parameter, CAPTCHA background **B**, is important to make sure that background has randomness to confuse automatic algorithms. **B** contains parameters such as the number of background shapes to be generated, the number of dilation operations to be adopted, and the number of random portions to be placed.
- Location (x, y) of each constituent image on the background image is an important factor. It should be set such that genuine human users should be able to distinguish between genuine faces and non-face images. Further, with random location, the segmentation is more difficult than if a static location scheme is used.
- Next, four operations are applied as follows:
  - *Rotate* operation [8] is used to rotate the constituent face and non-face images with θ degree angle. Since, for each constituent image, θ may be different, the parameter is represented in vector form such that $\Theta = \{\theta_i\}$ and $i = 1, 2, \cdots, n_{total}$.

– *Gamma correction* is used to randomly change the intensity of the constituent images with γ parameter. This parameter is also represented in vector form $\Gamma = \{\gamma_i\}$ and $i = 1, 2, \cdots, n_{total}$.
– $Blending\ operation$ [8] is used to smoothly blend the constituent face and non-face images with the background. In this operation, strength of blending S_b is used as the parameter which controls the degree of blending.
– Random lines, with random directions and colors, added in the image. The parameters are: nl (number of lines), dr^0 (direction of each line), cl (color of each line).

As shown in Figure 2 and the steps below, these parameters and operations are used to generate the CAPTCHA.

Step 1: From a set of face and non-face images, randomly select $n_{face} \geq 4$ (i.e., two genuine pairs of face images) and $n_{non-face} \geq 1$ (i.e., the number of non-face images).

Step 2: Each constituent image is processed using the distortion operations. First, each image is randomly rotated with θ^0 and then gamma correction is applied (with parameter γ_i).

Step 3: Each constituent face image is then placed at a randomly selected location (x, y) on the CAPTCHA background **B**. The following two approaches have been used for background generation.

- Random Colors: The background image is created using completely random amount of shapes such as circles, squares and crosses with randomly chosen sizes and colors. These shapes are then pasted on the canvas at random co-ordinates to generate the background image. This background image is then dilated before being used for CAPTCHA generation.
- Random Portions: In this approach, the face and non-face images selected for inclusion in a particular CAPTCHA, are used for the background generation process as well. Parts of the face/non-face images are selected randomly and used along with random colors approach to create the background image. These parts are pasted at random co-ordinates on the background image. This introduces skin color patches on the background and may generate false positives for skin color based algorithms.

Step 4: Finally, random lines and gamma correction are applied on the final CAPTCHA image.

For learning the optimal parameters for CAPTCHA generation, a set of parameters is manually selected and 300 CAPTCHAs are generated corresponding to these parameters. These CAPTCHAs were used as the training set and over a period of three weeks, 1794 responses were collected from 220 individuals. Based on these responses, 10 best performing CAPTCHAs were selected for generating the final set of test CAPTCHAs.

CAPTCHA Generation using Training Parameters: Test CAPTCHAs are generated using the best-performing parameter sets (obtained from training) and the four steps described earlier. Figure 3 shows examples of the proposed CAPTCHAs.

Implementation Details: The size of background image (and CAPTCHA) is 600×400, $4 \leq n_{face} \geq 7$, and $n_{total} = 12$. During training, sets with different parameters, as estimated using the described gradient descent algorithm, are created with all the combinations. Further, a commercial face recognition algorithm, denoted as COTS, is used as the automatic algorithm during training.

3. Experimental Results and Analysis

The proposed CAPTCHA is evaluated with several human users and the performance is compared with the automatic face recognition algorithm. This section presents the description of images used to generate the CAPTCHA, experimental protocol, and key results.

3.1. Database and Protocol

Generating CAPTCHA images requires human face and non-face images including cartoon face images. For experimental evaluation, we have selected face images from the AR face database and cartoon images (from photobucket.com). The cartoon 'face' images are chosen so that they generate false positives for face detection algorithms as they cannot distinguish well between human faces and cartoon faces. 40% of the images from these databases are used for training and the remaining 60% unseen images are used for testing. For human evaluation, about 3000 volunteers contributed in training and testing of the proposed CAPTCHA algorithm. Haar face detector, Google's Picasa, and a commercial face recognition algorithm (COTS) are used as the automatic algorithms to evaluate (break) the robustness of the generated CAPTCHAs. Even though the CAPTCHA requires face recognition as the test, Haar face detector is also chosen for evaluation because face detection in the first step for automatic face recognition. If an adversary cannot locate all the faces in the CAPTCHA, it is highly unlikely that it will be able to find a matching pair.

In both training and testing, each CAPTCHA is evaluated 4 to 10 times by different human users and once by each automatic algorithm. Therefore, we can compute the average accuracy for human responses (across all

Figure 3. Samples of the proposed face recognition based CAPTCHA. A user has to select one genuine face pair (two images belong to same user) in order to correctly solve the CAPTCHA.

CAPTCHAs) using the following equation,

$$Accuracy = \frac{Total\ number\ of\ correct\ responses}{Total\ number\ of\ responses} \times 100 \quad (7)$$

where, correct response means that in a given CAPTCHA, approximate centers of constituent genuine face pairs are correctly marked.

3.2. Analysis

With different settings during training, the human success rate was 96%. After the training phase, with optimized parameters, human tests are performed using the final set of 500 CAPTCHA images. Human users achieved 98% accuracy on the testing set. Key observations are as follows:

- To evaluate the human performance of the proposed CAPTCHA algorithm, we collected responses on a set of 500 CAPTCHAs. A total of 30,109 responses were collected from 3,000 users with an accuracy of 98% (29,507 correct responses).

- During this evaluation, a comparison among the proposed algorithm, Google's CAPTCHA and IMAGINATION CAPTCHA was also performed. For each of these existing and proposed CAPTCHAs, 2,997 responses were collected. For Google's CAPTCHA, 1,008 responses were correct whereas 1,646 responses were correct for IMAGINATION CAPTCHA. On the other hand, 2,937 responses were correct for the proposed face recognition CAPTCHA. Further, the volunteers suggested that the proposed CAPTCHA is much more intuitive and easier to solve compared to other two CAPTCHAs.

- We also performed a face detection and recognition test on the generated CAPTCHA images via the Haar Cascade detector in Open CV, Google's Picasa photo library's face detector, and a commercial face recognition software. These automatic algorithms were not able to detect all the faces from even a single CAPTCHA, with the maximum number of detected faces in any CAPTCHA being two (not the same genuine face pair). As shown in Table 1, automatic algorithms were not able to solve any CAPTCHA.

- Using the CAPTCHA breaking approach [17], another analysis was performed to evaluate the robustness of the proposed algorithm. In our experiments, the CAPTCHA breaking approach was unable to find the approximate locations of the face pairs. Therefore, the proposed algorithm is more stable and resilient to automated attacks.

- In order to estimate the security provided by this CAPTCHA design, we present a probability analysis of the success of a completely random guessing algorithm that attempts to break the CAPTCHA:

  The algorithm requires two user responses (clicks) in

Human	Haar Face Detector	Picasa	COTS
98%	0%	0%	0%

Table 1. Accuracy of human and automatic algorithms on 500 test Face CAPTCHAs.

which it has to solve the CAPTCHA. Both of these inputs have to be correct. Each CAPTCHA has two genuine pairs, and therefore four images which can act as a valid first input. The total number of pixels in the image is 600×400. The number of pixels covered by four genuine images is $100 \times 100 \times 4$ (assuming uniform size of 100×100 for every image, only for calculation purposes). Therefore, the probability of a random first response being valid is $\frac{4 \times 100 \times 100}{600 \times 400} = 0.16$. Now, the remaining area of the image is $600 \times 400 - 100 \times 100 = 230,000$ and the number of pixels that would result in a successful guess is 100×100 (carrying forward the assumption made for the first response) which correspond to the correct pair image. The probability of the second response being successful assuming the first click is successful is $\frac{100 \times 100}{230,000} = 0.043$. Therefore, the probability of breaking the CAPTCHA with a random guess is $0.16 \times 0.043 = 0.00688$ or $7 \, in \, 1000$. Please note that this calculation assumes that there are no overlaps in two images and also that all images have same uniform size, which is not the case in the actual implementation.

4. Conclusion

This paper presents a face recognition CAPTCHA algorithm that utilizes the difference between face recognition capabilities of human mind and automated algorithms. The proposed methodology offers major benefits over traditional text-based CAPTCHAs, most notably language independence. Further, the experimental results suggest that the proposed CAPTCHA algorithm is efficient with a high human accuracy and resilience towards automatic algorithms. By incorporating the proposed CAPTCHA into existing online authentication schemes, developers can substantially reduce the likelihood of credentials-based attacks. In requiring users to solve the CAPTCHA in addition to providing a username and password, an additional dimension of complexity can be added that requires human effort. The proposed CAPTCHA's point-and-click-based implementation adds this additional stage with minimum difficulty for users.

5. Acknowledgement

The authors would like to thank the volunteers who participated in this research.

References

[1] The official captcha site. http://www.captcha.net.

[2] H. Baird and K. Popat. Human interactive proofs and document image analysis. In *Document Analysis Systems*, pages 531–537, 2002.

[3] H. Baird and T. Riopka. Scattertype: a reading captcha resistant to segmentation attack. In *Proceedings of the SPIE Conference on Document Recognition and Retrieval*, pages 16–20, 2005.

[4] A. Basso and S. Sicco. Preventing massive automated access to web resources. *Computers and Security*, 28(3-4):174–188, 2009.

[5] R. Datta, J. Li, and J. Wang. Exploiting the human-machine gap in image recognition for designing captchas. *IEEE Transactions on Information Forensics and Security*, 4:504–518, 2009.

[6] J. Elson, J. Douceur, J. Howell, and J. Saul. Asirra: a captcha that exploits interest-aligned manual image categorization. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pages 366–374, 2007.

[7] P. Golle. Machine learning attacks against the asirra captcha. In *Proceedings of the 15th ACM Conference on Computer and Communications Security*, pages 535–542, 2008.

[8] R. Gonzalez and R. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., 2006.

[9] Esp-pix. Carnegie Mellon University.

[10] K. Kluever. Evaluating the usability and security of a video captcha. Master's thesis, Rochester Institute of Technology, 2008.

[11] H. Lamba, A. Sarkar, M. Vatsa, R. Singh, and A. Noore. Face recognition for look-alikes: A preliminary study. In *Proceedings of the International Joint Conference on Biometrics*, pages 1–6, 2011.

[12] A. Rusu and V. Govindaraju. Handwritten captcha: Using the difference in the abilities of humans and machines in reading handwritten words. In *Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition*, pages 226–231, 2004.

[13] S. Shirali-Shahreza and M. Shirali-Shahreza. Bibliography of works done on captcha. In *Proceedings of the 3rd International Conference on Intelligent System and Knowledge Engineering*, volume 1, pages 205 –210, 2008.

[14] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell. Face recognition by humans: Nineteen results all computer vision researchers should know about. *Proceedings of the IEEE*, 94(11):1948 –1962, 2006.

[15] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. recaptcha: Human-based character recognition via web security measures. *Science*, 321:1465–1468, 2008.

[16] O. Warner. The cutest human-test: Kittenauth. ThePC-Spy.com.

[17] B. B. Zhu, J. Yan, Q. Li, C. Yang, J. Liu, N. Xu, M. Yi, and K. Cai. Attacks and design of image recognition captchas. In *ACM conference on Computer and communications security*, pages 187–200, 2010.