EyamKayo: Interactive Gaze and Facial Expression Captcha

Utkarsh Dwivedi
IBM Research India
New Delhi
utkdwive@in.ibm.com

Ferdous A. Barbhuiya
Indian Institute of Information
Technology Guwahati
ferdous@iiitg.ac.in

Karan Ahuja
Indian Institute of Information
Technology Guwahati
karan.ahuja@iiitg.ac.in

Seema Nagar
IBM Research India
Bangalore
senagar3@in.ibm.com

Rahul Islam
Indian Institute of Information
Technology Guwahati
rahul.islam@iiitg.ac.in

Kuntal Dey
IBM Research India
New Delhi
kuntadey@in.ibm.com

IUI'17 Companion, March 13-16, 2017, Limassol, Cyprus
ACM 978-1-4503-4893-5/17/03.
http://dx.doi.org/10.1145/3030024.3038266

Abstract

This paper introduces *EyamKayo*, a first-of-its-kind interactive CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart), using eye gaze and facial expression based human interactions, to better distinguish humans from software robots. Our system generates a sequence of instructions, asking the user to follow a controlled sequence of gaze points, and generate a controlled sequence of facial expressions. We evaluate user comfort and system usability, and validate using usability tests.

Author Keywords

Gaze; Facial expression; Emotion; Interactive captcha

ACM Classification Keywords

H.5.m. [Information Interfaces and Presentation (e.g. HCI)]: Miscellaneous

Introduction

CAPTCHA systems attempt to distinguish humans from robots (bots), often aiming to prevent bots from accessing portals intended for human access. Examples of traditional captchas are given in Figures 1 and 2. With advances in deep learning, systems to break traditional text/image captchas are abundant. Google's reCAPTCHA uses recognizable images, followed by NoCAPTCHA with a simple checkbox proclaiming "I am not a robot". However, Sivakorn

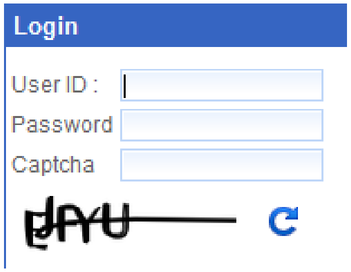

Figure 1: Traditional Text Captcha

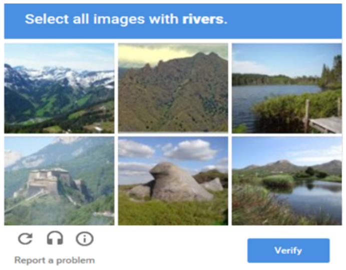

Figure 2: Traditional Multi-Image Captcha

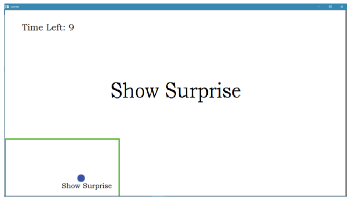

Figure 3: System action screenshot for $3X3$ grid

et al. [7] show the flaws in the reCAPTCHA system, successfully breaking 70.78% of the Image reCAPTCHA challenges. This calls for an improved captcha-security paradigm.

Using human-computer interaction has been proposed as a recent solution to the CATPCHA problem. De Marsico *et al.* [3] conducted a pioneering work, instructing the user to perform head gestures, *e.g.* head rotation or movement, and face recognition in addition, over webcams. Their work poses the needs and challenges for security based upon human-interactive CAPTCHA.

This motivates *EyamKayo*, a first-of-its-kind interactive CAPTCHA using human eye gaze and facial expressions[1]. It uses a combination of (dynamically) generated sequence of gaze and emotion instructions, and tests how accurately a user executes the instructions. Such a system is extremely difficult for bots to break today, if not impossible, because a breaker needs to look like a human being in 3D, as well as act like one by ingesting, understanding and executing instructions like humans. We validate on 20 users, over effectiveness, usability and user comfort tests.

Gaze and facial expressions have been separately used in different contexts, to build interactive systems and applications [5, 6]. However, an interactive CAPTCHA system using a combination of gaze and facial expressions, does not exit in the literature. Our work is thus a novel one.

Our Approach

Using gaze and emotion, for interactive security, is a first-of-its-kind proposition. The objective is to enable a sufficient dynamic captcha environment, that is significantly difficult for even the advanced deep learning systems to break. We propose *EyamKayo* with this objective.

Algorithm 1 EyamKayo

```
Boolean SuccessStatus = FALSE
Integer EGPairSuccessCount ← 0
Integer EGPairFailureCount ← 0
while (EGPairSuccessCount < JointThreshold) do
    if (EGPairFailureCount > FailureThreshold) then
        SuccessStatus ← FALSE
        break out from the while loop
    end if
    instruct user to gaze at a given zone/box on screen

    while (NOT(TimeOut) OR (UserInputSuccess)) do
        estimate user gaze point in the next video
        frame
    end while
    if (TimeOut == TRUE) then
        EGPairFailureCount ← + 1
        restart loop
    end if
    instruct user to express an emotion
    while (NOT(TimeOut) OR (UserInputSuccess)) do
        detect user emotion in the next video frame
    end while
    if (TimeOut == TRUE) then
        EGPairFailureCount ← + 1
        restart loop
    end if
    EGPairSuccessCount ← + 1
    if (EGPairSuccessCount >= JointThreshold) then
        SuccessStatus ← TRUE
    end if
end while
Output: SuccessStatus
```

[1] We use “emotions” and “facial expressions” interchangeably

EyamKayo - Secure Enough for Captcha? : To masquerade as humans in interactive systems with cameras to detect entities, the bots need to be "good" on several fronts. Their face needs to look alike to a human to avoid face detection failures. They need to read and/or listen to natural language instructions, and execute. For biometric spoofing and replay attacks, they need to dynamically exhibit "liveness behavior", *e.g.*, accurate eye and 3D head pose movements, and express emotional nuances on "face muscles". This needs an intelligent bot that looks and acts like humans. Such a "perfect" bot, is far from reality today, making *EyamKayo* difficult to break.

The Gaze Estimation Subsystem : We use OpenFace's [1] gaze estimation module for gaze tracking. It is real time, in-the-wild, calibration independent and does not need specialized hardware.

The Facial Expression Recognition Subsystem : We use SenTion [4], a person and scale independent framework for facial expression recognition.It's robustness on near-frontal images and 15-18 fps speed makes it practical.

To serve a user access request, we present an instruction sequence to the user, that requires her to perform gaze and emotion based actions. We grant access, if the user follows the instructions sufficiently, without many errors. The instructions can be delivered on voice, or in written form at the portion of the screen that the user is currently seen gazing at. Our methodology is presented in Algorithm 1.

Usability Study

We quantitatively and qualitatively assess the usability of our system for (a) efficiency, (b) effectiveness and (c) user satisfaction. We also report the score computed by system usability scale [2] from a user survey. The combination of the two input modalities, gaze and emotions, has not been explored in human interactive captcha before, and a comparison with traditional CAPTCHA systems or with FATCHA [3] (the only known human interactive proof system for CAPTCHA) is unlikely to be meaningful.

For user study, we recruited 20 undergraduate college students, 10 males and 10 females, all with normal or corrected-to-normal vision, distributed over an age range of 17-21 years. All had significant prior experience with desktop and laptop computers, and had faced traditional visual recognition captcha and OCR captcha before. The screen had a display resolution of $1,366$ px $\times$ 768 px and dimension of 35 cm $\times$ 19.5 cm. To determine a stable number of partitions of the screen (without compromising on gaze accuracy), we pilot-test with 2×2, 3×3, and 4×4 grids in the horizontal and vertical directions, assuming the box center to be the ideal point of gaze. Based upon the observations, we divide the screen into 3×3 parts. The average user distance from the screen is 50 cm, assuming an average gaze accuracy of 7.6 degrees. Based upon the findings of SenTion, we use Happy, Surprise and Angry as the acceptable emotions. Figure 3 shows a screenshot of our system.

The user study was conducted in two phases, training and testing. In training, the participants/users were taught how to use our system. Testing consisted of users using the system on their own and measuring their performance.

A trial consists of gazing at the asked position and showing the asked expression. Each trial had a timeout of 30 seconds. A trial is successful, if a user gazes at the instructed box, and shows the instructed emotion, else it is a failure. If a user fails a trial, he/she is needed to go through the next trial. Each session is a sequence of one or more trials. A session is successful, as soon as one trial is successful, and fails if all the trials fail. In training, each user was exposed to one or more sessions. We actively monitored the user over the training session, and allowed users to get trained until they reported comfortable using our system.

An example of instruction sequence could be the following. (1) **System:** "Look at the top left box on the screen". **User:** Gazes at the suggested box. (2) **System:** "Now, smile". **User:** Smiles. (3) **System:** "Look at the box right below". **User:** Gazes at the box just below the left top box. (4) **System:** "Now, show surprise". **User:** Opens her mouth.

For testing, each user was exposed to 5 sessions, with a maximum of 5 trials per session. We allowed a maximum time of 10 seconds per trial. The sessions were conducted in one sitting. We start a next session, when a user successfully passes the current session, or it times out.

Results

The effectiveness of our system during training phase is depicted in Figures 7 and 8. All but one user passed every session within the limit set for maximum trials in each session. Figure 7 shows the number of sessions needed by each user. Most users do not need more than 4 sessions for training. Figure 8 shows the average number of trials at-

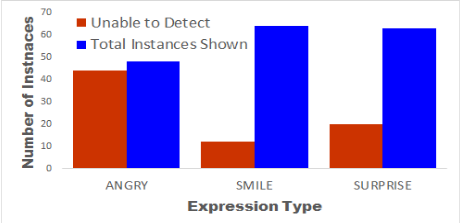

Figure 4: Accuracy of our system on different types of emotions in testing phase

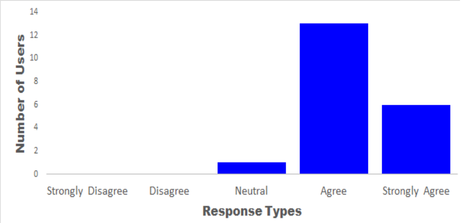

Figure 5: User response to: “I thought the system was easy to use”

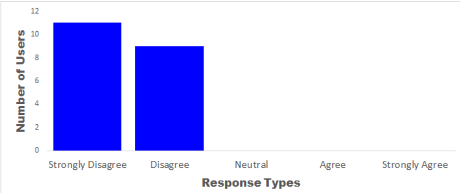

Figure 6: User response to: “I needed to learn a lot of things before I could get going with this system”

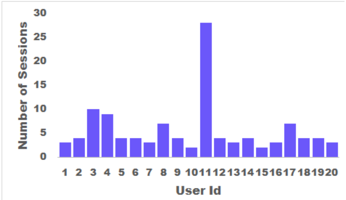

Figure 7: Sessions needed by each user in the training phase

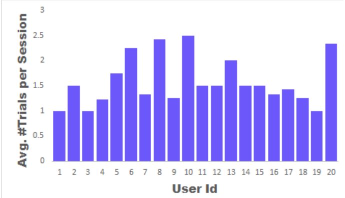

Figure 8: Average number of trials required per session in the training phase

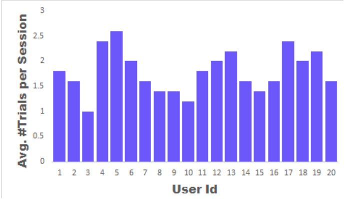

Figure 9: Average number of trials required per session in the testing phase

tempted in a session by each user. Most users do not need more than 2 trials per session.

For testing, each user was given 5 sessions. The maximum time allowed for each trial was 10 seconds. Each user passed all the sessions. Thus, the False Rejection Rate is zero (0). Figure 9 shows the effecitveness of our system.

Our survey results indicate the system in the current form is useful. We obtain a mean system usability score [2] 65.38, an acceptable usability range. Figures 5 and 6 show the responses of the users for the two most important questions asked from usability perspective. Evidently, the users agree the ease of use, without requiring any additional learning.

Conclusion

We proposed *EyamKayo*, a first-of-its-kind captcha system that uses a dynamically generated sequence, combining eye gaze estimation and facial expressions (emotions) of users. We tested on 20 users for effectiveness, usability and user comfort. Our tests indicate high success rates, low error rates, high usability and sufficient user comfort. Our system can, in future, replace traditional CAPTCHA.

REFERENCES

1. Tadas Baltru, Peter Robinson, Louis-Philippe Morency, and others. 2016. OpenFace: an open source facial behavior analysis toolkit. In *WACV*. IEEE, 1–10.
2. John Brooke. 1996. SUS-A quick and dirty usability scale. *Usability eval. in industry* 189, 194 (1996), 4–7.
3. Maria De Marsico, Luca Marchionni, Andrea Novelli, and Michael Oertel. 2016. FATCHA: biometrics lends tools for CAPTCHAs. *Multimedia Tools and Applications* (2016), 1–24.
4. Rahul Islam, Karan Ahuja, Sandip Karmakar, and Ferdous Barbhuiya. 2016. SenTion: A framework for Sensing Facial Expressions. *arXiv preprint arXiv:1608.04489* (2016).
5. Mohamed Khamis, Florian Alt, and Andreas Bulling. 2015. A field study on spontaneous gaze-based interaction with a public display using pursuits. In *ISWC (UbiComp)*. ACM, 863–872.
6. Carlos H Morimoto and Marcio RM Mimica. 2005. Eye gaze tracking techniques for interactive applications. *Comp. Vis. and Img. Understanding* 98, 1 (2005), 4–24.
7. Suphannee Sivakorn, Iasonas Polakis, and Angelos D Keromytis. 2016. I am robot:(deep) learning to break semantic image captchas. In *EuroS&P*. IEEE, 388–403.