Proceedings of the 11th Biannual Conference of the ACM SIGCHI Italian Chapter

2015, Rome, Italy

CHItaly 2015

General Chairs: Paolo Bottoni & Tiziana Catarci

Program Chairs: Maria De Marsico & Daniela Fogli

Proceedings Chair: Maristella Matera

In cooperation with:

The Association for Computing Machinery
2 Penn Plaza, Suite 701
New York New York 10121-0701

ACM ISBN: 978-1-4503-3684-0

Table of Contents

FATCHA: the CAPTCHA are you!

Maria De Marsico
Sapienza Università di Roma
Via Salaria, 113
Rome, Italy
demarsico@di.uniroma1.it

Luca Marchionni
Sapienza Università di Roma
Via Salaria, 113
Rome, Italy
luca.marchionni89@gmail.com

Andrea Novelli
Sapienza Università di Roma
Via Salaria, 113
Rome, Italy
andreanovelli91@gmail.com

Michael Oertel
Sapienza Università di Roma
Via Salaria, 113
Rome, Italy
oertel.michael90@gmail.com

ABSTRACT

In this paper, we propose an innovative type of CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart). These tests are used to allow a service to discriminate human users from (malicious) bots. With FATCHA, the user is simply asked to perform at random some trivial gesture, e.g., moving the head, which will be captured by the computer webcam and recognized by the server hosting the service. A second module in a possible composite service allows the user to authenticate by face recognition instead of using a password. In this way we significantly exploit the potentiality of multimodal interaction for both an advanced Human Interactive Proof (HIP) test and for robust/comfortable authentication.

CCS Concepts

•Security and privacy → Biometrics; Denial-of-service attacks; Usability in security and privacy; *Graphical / visual passwords; Access control;* **•Human-centered computing → Natural language interfaces; Gestural input; Accessibility technologies; •Computing methodologies → Scene understanding; Activity recognition and understanding; Object recognition;** *Image processing;* **•Information systems →** *Spam detection;* **•Social and professional topics →** People with disabilities;

Keywords

Multimodal interaction, human face detection, CAPTCHA, bot, computer security, denial of service, brute force attack.

1. INTRODUCTION

The popular acronym CAPTCHA stands for *Completely Automated Public Turing test to tell Computers and Humans Apart*. As easily guessed, the idea behind this interface trick is to prevent bots (computer programs) from using/abusing certain services, such as forums, website registration facilities, and comments collection. In practice, it is necessary to deny malicious access to everything that could be used to create spam, to breach security with operations such as brute force hacking, and/or to cause denial of service (DoS). In general, the term *bot* denotes a program that accesses the network through the same type of channels used by human users. Bots are not necessarily harmful. The beneficial purposes of such programs are usually related to the automation of tasks which would be too burdensome or complex for human users. The typical example is a web crawler that analyzes the content of the network, and is the basic resource for search engines. Therefore the term is not to be generally interpreted with a negative acceptation. However a large part of bots nowadays consists of programs that flood the network of spam. Just think of a program that draws from the network a huge number of mail contacts and sends them emails, or of scripts that automatically register to forums to spam links to malicious sites. The need has soon arisen to devise countermeasures being more or less suited to counter these malicious bots in order to proactively block any attempt to attack. This has spurred a new research area called Human Interactive Proofs (HIP), whose goal is to defend services from malicious attacks by differentiating bots from human users in the most reliable way. It is interesting that most services potentially affected by this problem require a starting interaction by the user. This observation suggests a key to address the problem. The solution provided by CAPTCHAs is to prompt the user to do something trivial for a human (like writing a sequence of numbers and letters) but complicated for a machine, in order to distinguish, e.g., a fake registration attempt. Typical examples are recognizing distorted text, with no doubt the most popular kind of CAPTCHA, or providing some information about displayed elements. According to some statistics [1], more than 100 million of the former ones are solved every day by people around the world, using a few seconds for each of them to type the recognized distorted characters that appear in the displayed images. In the following, we will mostly refer to this widespread kind of CAPTCHA.

The problem arising at this point is that, the more complicated the task for a machine (e.g., the more distorted the characters), the more it may become not so trivial for a

CHItaly '15 Rome, Italy

DOI: http://dx.doi.org/10.1145/2808435.2808437

human user too. This often causes repeated attempts, and may lead to a users' sense of bother. Despite being an excellent mechanism to prevent misuse of the online services on a large scale, it is worth taking into account the mental workload borne by the user, who is forced to interpret characters that in some cases are actually indecipherable due to their excessive distortion. The first and main contribution of this work is to propose a way to relieve the user by an excessive burden by offering an innovative type of CAPTCHA. The user is simply asked to perform trivial gestures like moving one's head or smiling, which will be captured by the computer webcam and recognized by the server of the service. The random choice of the gesture to perform hinders the possibility to submit a fake video. Moreover, the number of attempts before considering the connection a fake one can be kept very low, as it is highly improbable that a real user fails to perform more than two gestures in sequence, or that the system fails to recognize such sequence.
A second module in a possible composite service provides an authentication method based on face recognition. If desired, the user will be able to authenticate by simply showing the face to the webcam, rather than by the classic login procedure entailing to provide username and password. In this way multimodal user interaction can be used to protect both remote services and local resources.

2. PRESENT CAPTCHA APPROACHES

While all users of Internet services have probably seen a CAPTCHA, their history is less known. The first example of CAPTCHA was developed in 1997 by a research group led by Andrei Broder at AltaVista. The aim was to prevent bots from adding URLs to their search engine. In order to devise the reading task to propose to real users, to distinguish them from OCR-based automatic procedures, they adopted an apparently trivial strategy: they consulted the manual of a popular Brother OCR-equipped scanner and implemented just all the situations that the manual prescribed to avoid, e.g., twisted characters, noisy backgrounds, etc. An example is shown in fig. 1.

This idea was soon adopted by all the major search engines (e.g., Yahoo, HotBot, Excite) and by some free email services like Hotmail and Yahoo to prevent spammers from creating thousands of fake accounts. Nowadays CAPTCHAs are used to protect any website that provides an online service such as email providers, e-commerce sites, social networks, wikis and blogs. In addition, the term denotes any technique, inspired by artificial intelligence, able to distinguish a human from an automatic program, especially in relation with the solution of visual problems. We will mention some relevant examples that represent different approaches.
In [2] Imagination CAPTCHAs are introduced. The users are asked to annotate a distorted image in a controlled way. The distortions are applied so that the resulting images meet the requirements of a low perceptual degradation and of a high resistance to attacks based on the extraction of the image content. The system initially shows a mosaic made up of 8 images side by side. An example of how such a mosaic may appear is shown in fig. 2. The user then has to choose which of the 8 images to annotate, clicking near to its center. After this choice, the image is distorted and the system presents some words from which to choose the correct annotation to pass the CAPTCHA. Words proposed for annotation are carefully chosen to avoid ambiguity.
In [3] a different approach is proposed, which is based on the correct orientation of an image. The way to pass this CAPTCHA is to rotate the image and place it in its natural position. A possible example of image presented to the user is shown in fig. 3. The system contains an extensive database of images from which those trivially recognizable by a bot are filtered out, that is, those in which it is easy even for an automatic procedure to detect the correct orientation. These are mostly images containing faces, which would be easily rotated in the right position just applying an algorithm of face detection to find the right orientation.
At present, one of the most famous techniques is reCAPTCHA,

Figure 2: A simulation of an Imagination CAPTCHA.

Figure 3: A possible CAPTCHA based on correct image orientation.

which is owned by Google and is adopted in real world systems. The authors fully digitalized the entire archive of The New York Times, containing more than 13 million items collected from 1851 to the present day. All articles were scanned and submitted to OCR. Words not recognized by the OCR are then further distorted and used in pairs to generate the image to be displayed to the user (fig. 4). Actually, reCAPTCHA has also a second goal. The solutions entered by humans are used to improve the digitalization process [1]. This is another (somehow secondary) reason why only words that automated OCR programs cannot recognize are

used. However, both to this aim and to meet the goal of a CAPTCHA, the system needs to verify the user's answer. This is the reason why reCAPTCHA displays two words, one for which the answer is not known, and a second control word for which the answer is known, though not provided by the OCR. If the control word is correctly typed, the system assumes a human user and considers also the other word as typed correctly (fig. 4). reCAPTCHA has evolved in a free

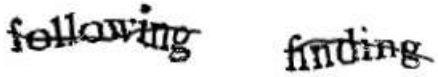

Figure 4: CAPTCHA based on correct image orientation.

service offered by Google to protect web sites, that helps to digitize books, newspapers and old time radio shows.
The group of Manuel Blum at Carnegie Mellon University has designed a number of different CAPTCHAs; some examples can be found at http://www.captcha.net.
A further alternative method involves displaying a (simple!) mathematical equation and requiring to enter the solution as verification. These are sometimes referred to as MAPTCHAs (where M stands for 'Mathematical').
Some of the most promising image-based techniques that are being developed in recent years require the recognition of a familiar face within an image. The core idea is that face detection is still a difficult task for a computer, especially due to problems related to PIE face distortions (Pose, Illumination, Expression) and cluttered backgrounds. Therefore these difficulties can be exploited to devise a robust CAPTCHA. An example of work based on these considerations is the one presenting ARTiFACIAL [4]. For each user request, ARTiFACIAL automatically synthesizes an image with a distorted face embedded in a cluttered background. The user must first find the face and then click on six points (four eye corners and two mouth corners) on the face. If the user can correctly identify these points, the system can conclude the user is a human; otherwise, the user is a machine. Similar considerations are exploited in [5]. In particular, the described approach starts from the assumption that, given two distorted images of a human face, the human user can recognize them as being of the same person quickly, while a computer program will hardly match them. The user is therefore presented with two sets of distorted human face images, each including distorted images of the same group of people. The user is expected to match the same person's faces in these two sets to pass the tests. A further example along a very similar line is presented in [6]. 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 includes the following steps:

1. The user sees a CAPTCHA image, with a randomly generated background with random shapes and colors, that contains human face images, cartoon face/non-face images, and some random object images with some level of distortion;

2. Each CAPTCHA shows at least two genuine pairs of human faces (i.e., belonging to the same person) along with some additional individual human faces; the user has to select one genuine pair of human faces out of the two genuine ones;

3. The user has to mark the approximate center of the 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 the check fails.

3. LIMITATIONS OF CURRENT APPROACHES

We remind here the guidelines for HIP discussed in [4]. The authors list the following requirements:

1. Automation and possibility to grade (the test should be automatically generated and graded by a machine);

2. Easy for humans;

3. Hard for machines;

4. Universality (feasible for any user);

5. Resistance to no-effort attacks (the test should survive no-effort attacks, i.e., those that can solve a HIP test without solving the hard AI problem);

6. Robustness when database is publicized.

A number of problems arise, especially when using text-based CAPTCHAs (see for example [7]. As mentioned above, the first and most evident problem of present approaches to CAPTCHAs, is that if the task is too difficult it may also challenge a human user in a frustrating way.
An even more serious problem with CAPTCHAs based on reading text, or on other visual-perception tasks, is the accessibility of the protected resource for blind or visually impaired users [8]. Access to relevant services may be completely blocked to the mentioned users. First, the main use of CAPTCHAs is as part of an initial registration process, or even as the starting step of every login. Second, being CAPTCHAs expressly designed to be unreadable by machines, common assistive technologies such as screen readers cannot help. Since CAPTCHAs do not have to be necessarily visual, speech recognition can be used to partially address this problem. As a matter of fact, at present, some implementations of CAPTCHAs allow opting for an audio CAPTCHA. People with sight difficulties can choose to identify a word being read to them as an alternative to reading the distorted text. However, people with severe motor impairment (to write the right answer) or speech impairment (to pronounce it) are still hindered in performing their remote tasks. It is further worth considering that providing an audio CAPTCHA allows blind users to hear the pronunciation of the written text, but it still hinders those who are both visually and hearing impaired. Elder people are an increasingly critical example of this class of users, due to the growth of average age in world population. Citing Wikipedia, "According to sense.org.uk, about 4% of people over 60 in the UK have both vision and hearing impairments. There are about 23,000 people in the UK who have serious vision and hearing impairments. According to The National Technical Assistance Consortium for Children and Young Adults Who Are Deaf-Blind (NTAC), the number of deafblind children in the USA increased from 9,516 to 10,471 during the period 2004 to 2012. Gallaudet University quotes 1980 to 2007 estimates which suggest upwards of

35,000 fully deafblind adults in the USA. Deafblind population estimates depend heavily on the degree of impairment used in the definition." Last but not least, if audio may be an alternative to text-based CAPTCHAs, it is hard to devise an alternative when images are used instead.
A further problem to consider is the increasing number of techniques aiming at circumventing CAPTCHAs. We do not consider the few approaches which exploit either cheap human labor to recognize them, or bugs in the implementation that allow the attacker to completely bypass the CAPTCHA, since they are related to factors which are "external" to the CAPTCHA nature. In particular, in the case of "cheap human labor" we cannot even say that the CAPTCHA was broken, but the service it should provide, since a real human is involved in solving the problem. Rather, it is interesting to especially consider those approaches based on improving character recognition, most of all by improving image enhancement procedures and segmentation. We can mention the work in [9]. The authors exploit object recognition in cluttered environments to defeat EZ-Gimpy and Gimpy CAPTCHAs, the first based on recognizing a single distorted word and the second requiring to recognize at least 3 words out of those displayed in a cluttered and distorted setting. The authors developed methods based on shape context matching that can identify the word in an EZ-Gimpy image with a success rate of 92%, and the required 3 words in a Gimpy image 33% of the time. As another example, we can mention the project PWNtcha (Pretend We're Not a Turing Computer but a Human Antagonist) by Samuel Hocevar, which is a decoder for text-based CAPTCHAs, online at http://caca.zoy.org/wiki/PWNtcha. Its goal is to demonstrate the inefficiency of many (mostly text-based) captcha implementations, besides claiming their accessibility limitations.
Audio too seems to be somehow vulnerable. A very recent post (Saturday, April 19, 2014) on Debasish Mandal's Blog, accessible at http://www.debasish.in/2014/04/attacking-audio-recaptcha-using-googles.html, discusses "Attacking Audio "re-Captcha" using Google's Web Speech API".
As already mentioned, image-based CAPTCHAs, which require an explicit recognition action by the user, are still more difficult to defeat. In particular, those based on face recognition are very promising. However, they also rise even greater accessibility problems, since there is no alternative way to submit such a CAPTCHA to a visually impaired user. In summary, we can say that most present CAPTCHAs lack one or another of the properties listed at the beginning of this section.

4. OUR PROPOSAL: FATCHA

Starting from the above discussion, we can draw an important consideration. Many problems with present approaches to CAPTCHAs rise from the need for an active perceptual and cognitive participation by the user. In other words, the user has either to recognize, or to modify, annotate, etc. some interface element. If we reduce the active role of the user to a very easy action, which does not involve any perceptual task, we can overcome almost all accessibility problems. We maintain the main feature of CAPTCHA test (easy for human, hard for machine), but in a certain sense we extend the concept of Turing test by asking the user to produce (synthesize) rather than analyze something. The easiest product to ask is a simple gesture, selected at random from a possibly wide yet simple set, which is checked on the server side as it was the response in classical CAPTCHA implementations. In other words, in some sense the user is the CAPTCHA. Moreover, some simple tricks allow to also avoid the most trivial attacks at least.
First of all, we chose an input device requiring the lower amount of physical action by the user, and ensuring the largest availability, namely the computer webcam. Thanks to the pervasive presence of such device on both desktop and mobile equipment, we can reasonably assume its presence. Having chosen such input device to capture the new CAPTCHA, it is a natural consequence, in this prototyping phase, to request and recognize gestures related to the face. For a human being, making gestures like shaking one's head, saying yes or no, or showing the requested ear is trivial; the same actions require a bot to be equipped with a gallery of videos to transmit to the server to be processed, and to choose the right one at the right moment.
We apply some methods inherited from the biometrics field. The first algorithm used is the popular Viola-Jones object detection [10], which is available in OpenCV as well as MATLAB libraries. The algorithm can be trained to detect the presence of a face or its parts (e.g., ear). It relies on a boosting approach [11] and exploits Haar-like features. In particular, AdaBoost is used, that exploits M weak classifiers to build a strong one by their linear combination. In practice Viola-Jones algorithm is initially trained using a training set containing instances of both positive examples (images in which the object to be detected is present) and of negative examples (i.e., images in which the object is not present). At each iteration, AdaBoost updates the weights of the weak classifiers as well as of the training images (positive and negative samples) depending on whether the classification in the previous iteration was correct or wrong. The training aims at identifying the most discriminating features to be used for detection, selecting them from a set that in this case consists of features like the ones in fig. 5 that are used as masks. The value of each feature is given by the difference of the pixels falling under the dark areas and the pixels under the light areas. In the first version of the prototype we implemented, the system detects frontal faces, left ear, right ear, and smile. The user is asked to make a gesture showing the requested part or to smile, the video transmitted by the webcam is processed, and the system decides if the gesture is the requested one. If not, it asks again for a limited number of times, after which it decides that the connection is a fake.

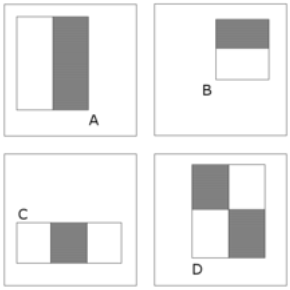

Figure 5: Examples of Haar features.

In more detail, during the session, the user interacts with the system FATCHA by visiting a web page which includes

the registration form. When the page opens, the system opens a web socket communication on port 9990, on which the client will transmit the frames extracted from the video sequence of the actions performed by the user (5 sec video, 1 frame every 250 ms, i.e., 20 frames to process) The frame extraction procedure is called from a function written using the JavaScript library jQuery and the functionalities of HTML5. At present, the JavaScript library jQuery and HTML5 technologies used to implement the system are compatible with almost all major browsers, such as Google Chrome, Mozilla Firefox and Opera. The servlet listening on the service port is written in Python, and has the task of receiving the frames and submitting them to the procedure developed using OpenCV framework in C++. The summary of the communication is shown in fig. 6. Frames received on the server are processed to verify the presence of the gesture requested by the system.

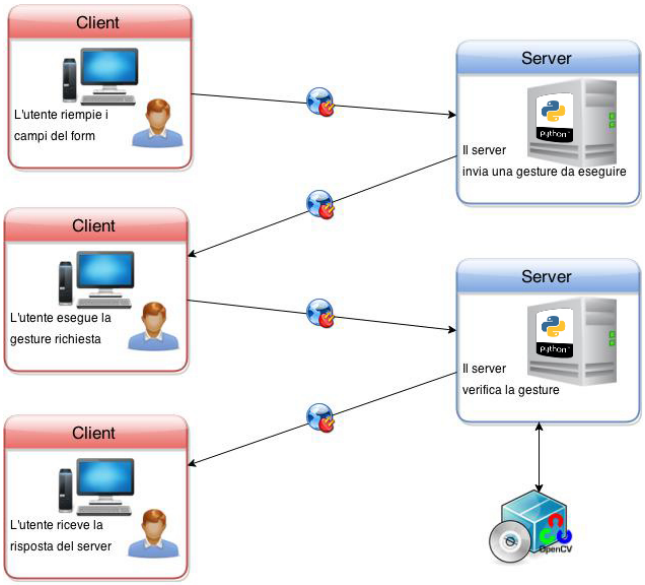

Figure 6: FATCHA communication for HIP.

As for the present prototype, the number of face actions envisaged for the system is not that high. In principle, repeated attempts may guess the right one. The solution may be to increase the number of face actions. However, this would make the framework more computationally demanding. As a matter of fact, facial action recognition is a very active research field due to the complexity of the task for an unrestricted system (see for example [12] and [13]). For this reason, we are planning to add to the gallery of requested actions simple combinations with easily detectable hand configurations. Each combination will require either one or both hands to appear together with one of the presently envisaged face actions, e.g., with a different number of fingers shown. The aim is to compose easy to process elements in order to build a strong configuration.

This part of the system does not rise privacy issues, since there is no need for storing the frames captured by the webcam. They can be destroyed as soon as they are not useful anymore for the intended processing.

5. BEYOND CAPTCHA TEST: AUTHENTICATION

The interesting consequence of acquiring a (discriminative enough) user trait is that, after CAPTCHA test, the same trait can be used for authentication too, substituting or complementing the traditional use of username and password. Actually, FATCHA and authentication modules can be combined in a number of ways for granting access to a specific remote service, according to the security policy adopted by the provider.

Even in this case we borrow from biometrics. We use Local Binary Pattern (LBP) [14] for face recognition. Given the modularity of our proposal, this technique can be substituted.

In LBP method, the value of each pixel is compared with that of its neighbors and acts as a local threshold. If the intensity of the examined neighbor pixel is greater than that of the central pixel, then it is assigned a 1, otherwise a 0 (an example I shown in fig. 7), overall resulting in a binary number (code) assigned to the central pixel. An histogram of such values is computed and used for comparison. In general, instead of applying LBP to the entire image, a grid is defined and histograms computed on each cell of the grid are chained to make up the final feature vector. This localized approach better addresses a number of possible variations in face appearance. Moreover, a different number of neighbors can be considered, and a different radius of the neighborhood window.

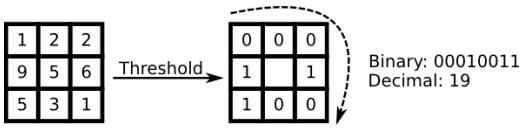

Figure 7: An example of computation of LBP code.

fig. 8 shows the communication between the components in the user recognition phase. When the user wants to authenticate via FATCHA, he/she visits the login page. The technologies used are the same as for the face detection phase. Login data is managed by a script .cgi in Python, which checks the existence and correctness of the provided username and password and/or (according to the adopted authentication policy) runs the procedure developed in OpenCV to recognize the user image captured by the webcam. Recognition can be carried out according to two modalities. In verification, matching involves only the image(s) in the database belonging to the identity possibly claimed by introducing a username (1:1 identity matching), in order to either accept or refuse the claim. In identification, matching is carried out against the whole database (1:N identity matching) to determine the right identity or refuse the user as unknown. In both cases, biometric authentication can be used either to substitute or to enforce traditional one using username and password.

According to the chosen security policy, the modules in fig. 6 and fig. 8 can be chained in a unique secure access procedure. Of course, the authentication part may rise privacy issues, since the biometric data of the user must be transmitted on the communication channel and stored on the authentication server. However, nowadays many solutions are

available to address these problems. Among them, we can mention biometric cryptosystems and cancelable templates (for a survey see for example [15]). In the first case, cryptographic techniques are used to protect the sensible data. Cancelable biometrics entails the systematically repeatable distortion of biometric features aiming at protecting sensitive user data. If a cancelable feature is compromised, the distortion rules are changed, and the same biometrics is mapped onto a new template, which is used afterwards.

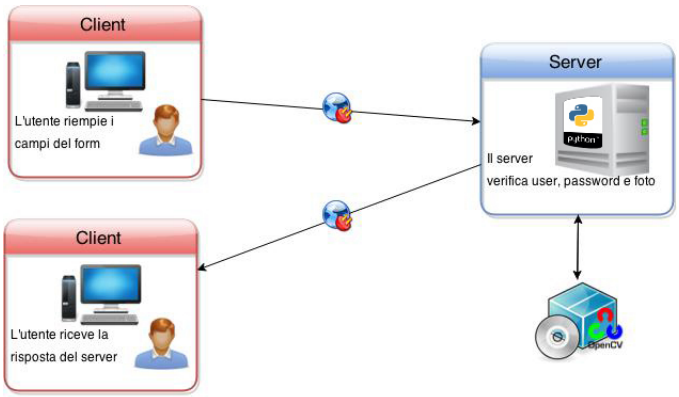

Figure 8: FATCHA communication for recognition.

6. TEST RESULTS

To analyze the correctness of the gesture recognition, the system was tested by 25 users. Each of them had to perform about 5 gestures chosen randomly by the system. We use the following conventions:

- TP = gesture made by the user is equal to that prompted by the system and is correctly recognized;
- FP = gesture made by the user is different from that prompted by the system but is recognized;
- TN = gesture made by the user is different from that prompted by the system and is correctly not recognized;
- FN = gesture made by the user is equal to that prompted by the system but is not recognized.

The measures used to evaluate recognition performances are:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$\text{F-Score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (4)$$

We also need to evaluate the effectiveness of the CAPTCHA scheme. Considering HIP test security, quoting [7] "we deem a [text-based] captcha scheme broken when the attacker is able to reach a precision of at least 1%". However, it is out of our scope to simulate an attack to the system, which would entail injecting on the channel usually used for the webcam a video with the required gesture, after having caught the system request (a text message or an audio recording). For the time being, we assume that FP can be considered as a good measure for a zero-effort attack, where the bot only has to submit a random video with a gesture, and pass the test even if the gesture is different. For the system it is extremely important not to allow a bot to overcome the CAPTCHA, i.e., to have false positives tending to 0. We can also measure the "robustness" of the single gestures, i.e. how much it is difficult to erroneously recognize them. In this way too easy gestures can be discarded from the list of possibly requested ones. To this aim, besides values provided by FP for each gesture, we also evaluate the False Positive Rate (FPR) both for single gestures and for the overall testing.

$$FPR = \frac{FP}{TN + FP} \quad (5)$$

Table 1 shows the results obtained for the HIP test. It shows that the condition of false positives tending to 0 is respected by practically all the gestures: only in the case of smile it happened that the system deemed valid a relatively high number of actions that actually corresponded to different gestures/expressions. In this situation, as Figure 9(a) shows, the system returns a percentage of correct responses of about 82%, with a 15% of false negatives, meaning a good robustness to user errors. These results represent a good achievement given the very limited set of alternatives. However, FPR is above 8,5%, which is quite high. It is clear that such results are penalized from the bad values obtained with smile recognition. As a matter of fact, the recognition of gestures or expressions derived from the emotions is one of the most critical tasks faced nowadays in pattern recognition for HCI [16]. Figure 9(b) shows the results obtained without including the detection of a smile. The system shows a significant performance improvement in all four measures being considered: 1) Accuracy: from 0.82 in 0.869 (+0.05); 2) Precision: from 0.949 to 0.988 (+0.04); 3) Recall: from 0.775 to 0.802 (+0.027); 4) F-score: from 0.853 to 0.885 (+0.032). The percentage of correct responses increases to 86.8% and false positives decrease to 0.7%. However, the most impressive improvement comes from considering the decrease of FPR: from 0,086 to 0.017 (-0.069. i.e., a decrease of above 80%).

As discussed above, the complementary requirement is to consider the needs of system users. For the purposes of usability it is also important to limit false negatives, i.e., those gestures made properly by the user which are considered invalid by the system. Making the user repeat the gesture, does not create a security hole, as in the case of false positive, but if this situation happens too frequently this may hinder a comfortable use of the service: if it is correct to make rigid checks to determine the correctness of the gestures, it is also important to keep the system usable. A service that accepts only the gestures performed perfectly might force the user to repeat the movement a large number of times therefore soon becoming boring and frustrating. In order to evaluate the behavior of the system with respect to this requirement, we consider the False Negative Rate (FNR):

$$FNR = \frac{FN}{TN + TP} \quad (6)$$

Gestures	Measures								
	TP	TN	FP	FN	Tries	Accuracy	Precision	Recall	F-score
NO	21	10	1	8	40	0.775	0.955	0.724	0.824
YES	22	15	0	3	40	0.925	1	0.88	0.936
SHOW RX EAR	19	16	0	5	40	0.875	1	0.792	0.884
SHOW LX EAR	19	17	0	4	40	0.9	1	0.826	0.905
SMILE	19	6	5	10	40	0.625	0.792	0.655	0.717
TOTAL	100	64	6	30	200	0.82	0.949	0.755	0.853
TOTAL-SMILE	81	58	1	20	160	0.87	0.988	0.8	0.884

Table 1: Results of gesture recognition tests.

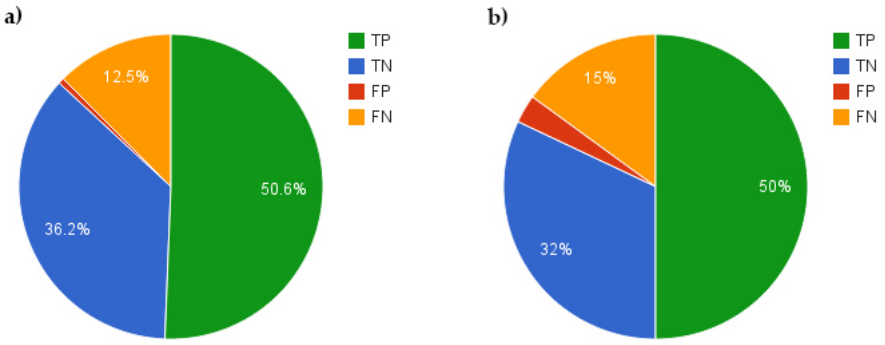

Figure 9: System performance in percentages with (a) and without (b) considering smile expression.

We find again that smile is one of the problems, but this time the improvement discarding it is not dramatic, passing from 0.23 to 0.2 only. The rate of false negatives is still 1 in 5 attempts. This is not that bad, due to the good security level that always calls for a compromise. Moreover this results can be improved by adopting more accurate and up-to-date techniques for gesture recognition.

7. FACE RECOGNITION PERFORMANCE

Authentication was not the primary goal of this work. However, since a simple face recognition module was developed and preliminary simple testing was performed, it is worth reporting the results obtained during the experimental phase. The system was tested in identification mode: the user does not claim an identity, and the system recognizes him/her from the face among a set of enrolled (registered) identities. In order to avoid a simple spoofing attack consisting in presenting a still image or a video of an enrolled user, FATCHA test is used before authentication.
Identification tests were performed with a dataset of 17 users, each with an average of 4 photos in the enrollment gallery. The access attempts were performed by 13 out of the 17 persons in the dataset (genuine users) and by other 8 persons not present (impostors). All tests were conducted in a controlled setting (webcam, location and lighting were fixed). A video sequence similar to that exploited by FACTCHA was used, i.e., for each test, 5 sec video, one frame every 250 ms (20 frame to be processed), with the difference that only frames where a face was detected were processed for recognition. The procedure is the following. For each frame in which a face is detected, identification is performed and only the first identity in a list ordered by similarity is returned for that frame. While processing the set of frames (possibly less than 20 if the face was not detected for some reason, e.g., a rotation of the head), the system usually finds one (always the same in all frames) or two identities. In the case of different identities, the system returns the identity returned most times. In our tests, we evaluated performances also taking into account the second identity returned most times after the first. The test set consisted of 101 tries. The 8 impostors appear 21 times while genuine users appear 80 times. The Cumulative Match Score (CMS) at rank 1 (i.e., the percentage of times the right identity was at first place) is 0.713, which is a good one given the capture device and the simple recognition approach used. CMS at rank 2 (i.e., the percentage of times the right identity was at first or second place) is 0.911. In more detail, we achieved:

- Top one = 72: in 72 cases the value returned most often was the correct identity of a registered user or the value "not known" for an anonymous user;
- Top two = 20: in 20 cases the value returned most often after the first was the correct identity of a registered user or the value "not known" for an anonymous user;
- True positive = 54: these are the cases out of 72 where the most frequent value returned was the correct id of the registered user;
- True negative = 18: these are the cases where the most frequent value was "not known" and it was correct;
- Misses = 9: cases in which the two most often returned values were both errors;
- Misses + top two = 29;
- False positive = 1: in 1 out of these 29 cases an impostor user was recognized as a registered one;

- False negative = 28: in 28 out of these 29 a genuine user was not recognized (including cases when is returned as the second candidate). It is worth underlining that a more accurate evaluation of recognition performance is out of the scope of this work. However, given the extremely prototypical implementation of the recognition module, we can say that results are quite aligned with present approaches in face recognition.

8. CONCLUSIONS

The proposed prototype implements an innovative type of CAPTCHA, based on the use of gestures captured through the computer webcam. In this way, a multimodal HIP test is set up. After this, it is possible to run facial recognition of the human face, by processing a new video sequence.
The opinions informally provided by users that tested FATCHA were very positive; the results are satisfactory, and we are confident to even improve them by further development.
The exception affecting system performance is the smile detection, which has led to an increase of false positives. To improve this aspect we plan to follow two strategies: a first one is to use a larger data set to train the system to recognize this difficult expression, the other one is to use a different algorithm. Moreover, we plan to adopt more accurate techniques for gesture recognition to make the system more usable.
The current version of FATCHA works with a limited number of gestures. It should be realistic to repeat the specific action only after a large (nearly infinite) number of attempts. To increase the number of gestures to be executed we plan to add those related to the hand. In this way, also calculating the possible combinations made with hands and with the face, the number of gestures can be increased significantly.
As for the part of face recognition, the system developed can be used as a further strengthening of the user authentication procedure, by adding the user's identity biometric check to the classic authentication by username and password. Since face authentication with the only face can be circumvented with the use of images of a registered user of the system, the HIP test can be used even to prompt the user, who wants to authenticate, to make a small gesture in order to prove to the system that the images captured by the webcam are not coming from a photo or a pre-recorded video. In this way FATCHA would handle in a uniform way both registration and authentication phases.

9. REFERENCES

[1] Luis Von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. recaptcha: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008.

[2] Ritendra Datta, Jia Li, and James Z Wang. Imagination: a robust image-based captcha generation system. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 331–334. ACM, 2005.

[3] Rich Gossweiler, Maryam Kamvar, and Shumeet Baluja. What's up captcha?: a captcha based on image orientation. In *Proceedings of the 18th international conference on World wide web*, pages 841–850. ACM, 2009.

[4] Yong Rui and Zicheng Liu. Artifacial: Automated reverse turing test using facial features. *Multimedia Systems*, 9(6):493–502, 2004.

[5] Deapesh Misra and Kris Gaj. Face recognition captchas. In *Telecommunications, 2006. AICT-ICIW'06. International Conference on Internet and Web Applications and Services/Advanced International Conference on*, pages 122–122. IEEE, 2006.

[6] Gaurav Goswami, Richa Singh, Mayank Vatsa, Brian Powell, and Afzel Noore. Face recognition captcha. In *Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on*, pages 412–417. IEEE, 2012.

[7] Elie Bursztein, Matthieu Martin, and John Mitchell. Text-based captcha strengths and weaknesses. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 125–138. ACM, 2011.

[8] Matt May. Inaccessibility of captcha. *Alternatives to Visual Turing Tests on the Web. I: W3C (red.), W3C Working Group Note, work in progress*, 2005.

[9] Greg Mori and Jitendra Malik. Recognizing objects in adversarial clutter: Breaking a visual captcha. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–134. IEEE, 2003.

[10] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[11] Robert E Schapire. The boosting approach to machine learning: An overview. In *Nonlinear estimation and classification*, pages 149–171. Springer, 2003.

[12] Maja Pantic and Leon JM Rothkrantz. Facial action recognition for facial expression analysis from static face images. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(3):1449–1461, 2004.

[13] Daniel Weinland, Remi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, 2011.

[14] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):2037–2041, 2006.

[15] Christian Rathgeb and Andreas Uhl. A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security*, 2011(1):1–25, 2011.

[16] Georgia Sandbach, Stefanos Zafeiriou, Maja Pantic, and Lijun Yin. Static and dynamic 3d facial expression recognition: A comprehensive survey. *Image and Vision Computing*, 30(10):683–697, 2012.