EmojiTCHA: Using Emotion Recognition to Tell Computers and Humans Apart

David Lorenzi[1], Jaideep Vaidya[1](✉), Achyuta Aich[2], Shamik Sural[2], Vijayalakshmi Atluri[1], and Joseph Calca[3]

[1] MSIS Department, Rutgers University, Newark, NJ, USA
jsvaidya@rbs.rutgers.edu
[2] CSE Department, IIT Kharagpur, Kharagpur, WB, India
[3] Cloud Creative Group, Tempe, AZ, USA

Abstract. Any successful CAPTCHA design must creatively balance the three competing criteria of usability, scalability, and robustness to achieve widespread deployment in public facing web services. We propose a novel CAPTCHA called EmojiTCHA which utilizes symbolic representations of human emotions in the form of emojis correlated to an image of real humans expressing the same emotion on their face. By leveraging the Project Oxford Emotion API from Microsoft's cognitive services platform, which provides automated detection of human emotion expressions on human faces, we generate a tagged dataset in an automated fashion. Through the use of image warping and distortion techniques, we can significantly increase the robustness of the CAPTCHA against automated attacks, without compromising on usability, as confirmed by our user study.

Keywords: CAPTCHA · Emotion recognition · Online security · Usability

1 Introduction

The Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) was invented by von Ahn et al. [2] to enable discrimination between humans and computers online. CAPTCHAs are reverse Turing tests administered by computers designed to keep bots from abusing web services and online forms made for human users. CAPTCHAs rely on hard AI problems to provide the challenge question asked to the user (human or bot). This ensures that the challenge question is one that is difficult for a computer to perform with a high degree of success, yet still remains easy for a human to perform quickly.

Designing effective CAPTCHA challenges has been an ongoing subject of research for more than a decade. Since the widespread introduction of the traditional text based CAPTCHA challenge where the user is asked to enter a string of characters to demonstrate they are human, CAPTCHAs have evolved significantly in style, design and complexity over time as they respond to advancements

© IFIP International Federation for Information Processing 2017
Published by Springer International Publishing AG 2017. All Rights Reserved
S. De Capitani di Vimercati and F. Martinelli (Eds.): SEC 2017, IFIP AICT 502, pp. 281–295, 2017.
DOI: 10.1007/978-3-319-58469-0_19

in attacks from image processing, computer vision and attacker creativity. The greatest challenge in designing a successful CAPTCHA that serves its intended purpose to distinguish between human users and bots is managing the tradeoffs between the competing requirements of usability, scalability and robustness with regard to design.

While text based CAPTCHAs have been the *de facto* style for CAPTCHA implementations on public websites that need protection from bot abuse, their usage is quickly falling out of favor as more advanced attacks and deep learning models have evolved that can solve the challenge at ever more accurate rates in an automated fashion. As image processing and computer vision algorithms become increasingly more adept at solving traditionally complex problems such as object and text recognition (and even attempts at scene recognition and identifying contextual information contained in an image) [21], the standard text based CAPTCHA has increasingly become obsolete.

Newer models for CAPTCHA that are replacing traditional text based challenges rely on strong image identification/object recognition tasks as their primary challenge method. One example of this is Google's new image based version of reCAPTCHA, which asks users to select all images that relate to a particular category (e.g., select all images that contain street signs) from a grid of nine images.

Behavior based models that focus on unique traits and actions of the target user such as websites visited, user agent of browser, geographic location of IP address, browser cookies, etc., to determine a probabilistic score as to whether or not the target user is legitimate are also gaining traction as a new way to distinguish between bot and human. Although these models are not like text/image based CAPTCHAs, they represent a new risk-calculation based approach to this problem. While these image based models are emerging as the preferred alternative to text based CAPTCHAs, they require extensive investments in backend infrastructure and data gathering capabilities (typically at Google scale) to operate in a secure and effective manner. Thus, although the these traditional methods eventually will be broken (such as text vs. deep learning models), for the time being, image based CAPTCHAs can still withstand sophisticated attacks.

In this paper, we demonstrate our design and implementation for a new image based CAPTCHA – one that meets all the three criteria of usability, scalability and robustness. The CAPTCHA is constructed entirely from freely available online tools, open source software and emojis. The central challenge question revolves around the task of asking the user to match an emoji whose expressed emotion corresponds to the face contained in the image(s) displayed. Utilizing the Microsoft Cognitive Services Platform's Project Oxford Emotion API [1], human faces and the emotions they are expressing can be detected in an automated fashion from images. This information is subsequently stored and used for preparing a CAPTCHA challenge. However, to prevent the tool itself as well as other image lookup services from being used against the challenge, image warping, noise and distortions are introduced to the image, thus providing security.

2 Preliminaries

In this section, we cover the core components and tools used to construct the CAPTCHA challenge that allows for the design to be usable, scalable and robust – i.e., providing a reasonable level of security for the online form it is protecting.

2.1 Microsoft Project Oxford

Microsoft's Project Oxford is a collection of easy to use artificial intelligence based vision, speech and language APIs that are cloud accessible and can be used in applications by developers. In our CAPTCHA design, we utilize the Face API and the Emotion API. The Emotion API takes an image as an input, and returns the confidence score across a set of emotions for each face in the image, as well as the bounding box for the face, using the Face API. The emotions detected are anger, contempt, disgust, fear, happiness, neutral, sadness, and surprise. These emotions are understood to be cross-culturally and universally communicated with particular facial expressions.

Project Oxford's Emotion API is a REST API provided by Microsoft and can be interacted with online. This tool is what provides the critical functionality that delivers the scalability capabilities for our CAPTCHA design. It supports an automated method to accurately and consistently identify and tag emotions within images that contain people's faces. Indeed, we store the output of the Emotion API in a database along with the image and subsequently use it in a challenge served to a user, which asks the user to identify the emotions depicted in the image. The power of this service is that the algorithm can easily scale with demand on the CAPTCHA challenge service, e.g., instances can be run in parallel to produce the requested volume of tagged output as required by the challenge service, i.e., number of unique challenges that need to be served at a particular rate. In order to prevent the use of this and other similar tools against the CAPTCHA, noise is added to the original face image, thereby ensuring failure to identify emotion or faces on images used in challenges. The process of using and applying image noise is described in more detail in the Methodology Section. More details of the Emotion API can be found at https://www.projectoxford.ai/emotion.

2.2 Emoji Character Set

The emoji character set is a UNICODE character set designed to convey complex ideas and emotions in the form of small and simple ideograms and/or pictograms. The cross-cultural nature of emojis enhances the usability as it removes specific language and alphabets as a barrier to usability. Our challenge asks a user to match emotions of people in an image with an emoji that conveys the same emotion, providing a solid basis for a simple CAPTCHA challenge task that is easy for humans to understand. Furthermore, since this character set consists of images instead of text, techniques used to provide noise to the images will also work on the emoji characters, which can be scaled based on font size and noised to thwart attackers further, though we do not do so right now. Widespread availability of the emoji UNICODE character sets on smartphone and tablet operating systems ensures high portability on mobile devices that use touchscreens. A list of emoji characters is available at http://apps.timwhitlock.info/emoji/tables/unicode. In our particular implementation for experimentation, we have chosen to use the Twitter emoji set which is open sourced by Twitter for public use. Note that in the current implementation we do not use the UNICODE characters, we simply use the image, since this is easier and does not impact the CAPTCHA itself. UNICODE integration is deferred to future work.

3 Related Work

Text based CAPTCHA challenges have been under attack by various computer vision and image processing tools since their release via services hosted online. Segmentation attacks, pixel count attacks, filtration attacks and more have all proven effective against certain implementations. Bursztein et al. provide an overview of the strengths and weaknesses of text based CAPTCHA and demonstrate the need to continue to advance the field of CAPTCHA research [3]. As of 2016, text CAPTCHAs are not being used to a great extent due to deep learning models being able to decipher characters, even distorted and obfuscated ones, at an accuracy of close to 99.8% [10]. The security field is moving forward with new designs to supersede text based CAPTCHA. For example, Google has developed reCAPTCHA to use images from its image search library and streetview images gathered by its Maps program to provide challenges for the user to solve (categorization task and text entry tasks, respectively) [10].

While the concept of image CAPTCHAs has been known for a while, their designs have certain inherent properties that make them strong candidates for communicating complex ideas to humans in a quick and efficient manner. Most versions use some form of object/image recognition [11,15] or categorization task [6,7,16] as their primary challenge. The most common shortcoming of existing image based CAPTCHAs, however, is their inability to scale - due to the fact that the images used need to be manually collected, edited, tagged and indexed, be unique, etc. Also, attackers have had some degree of success beating them using image processing and computer vision tools [22] together with novel machine learning [9] techniques to solve the challenges. Three-Dimensional models [14,19] and spatial/depth perception [13,18] are gaining popularity in the image CAPTCHA space as strong use cases for challenges. This style represents an interesting avenue of research as they present challenges that are not singularly straightforward for a machine to solve, as the primary task asked of the user requires multiple subtasks, such as image manipulation by the user or using a mouse instead of a keyboard [4], that must be completed in conjunction to solve the challenge. The Puzzle Only Solvable by Humans (POSH) [5] is another approach to exploit human interaction for discrimination between humans and computers. A POSH can be generated by a computer, can be consistently answered by a human, and a human answer cannot be efficiently predicted by a computer. However, a POSH does not even have to be verifiable by a computer at all. Usability of CAPTCHAs is also a key issue [20] and new solutions are being devised to provide a fair trade-off between security and usability [8].

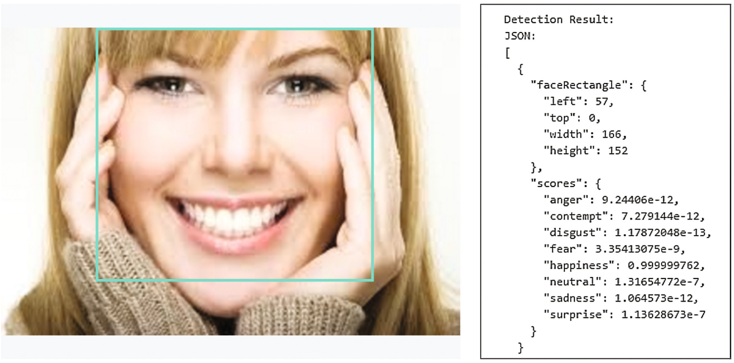
Although human face image based CAPTCHAs have been studied before [15,17], our work requires multiple subtasks, such as identifying the face in the image, determining their expression, and subsequently matching the appropriate emoji to the correct face. Our key contribution is to develop a scalable and usable image based CAPTCHA that is difficult for a machine to solve.

4 Methodology

In this section, we discuss the design choices that were made in order to ensure the usability, scalability, and robustness of the proposed CAPTCHA while demonstrating the security it provides from potential attacks.

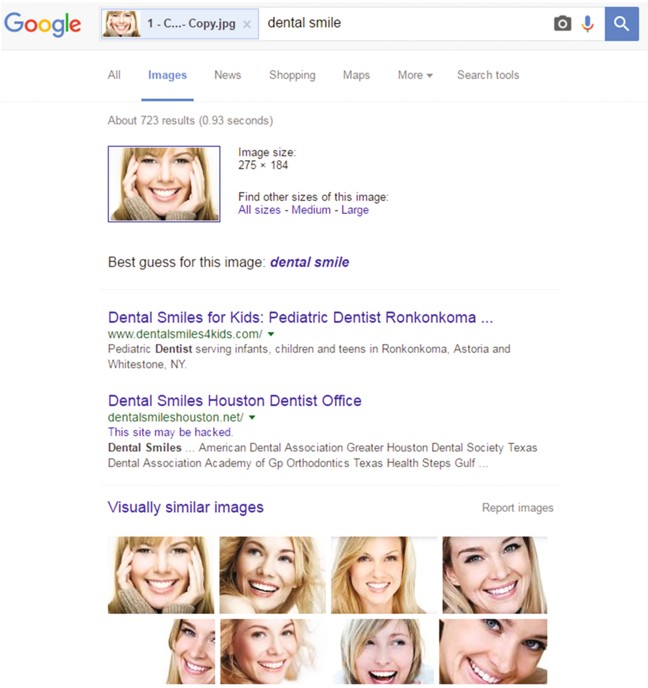
Figure 1 provides an example of the Microsoft Emotion API output. Using a sample face that is smiling (a depiction of the emotion "happiness"), the API provides the coordinates for a faceRectangle, which is a bounding box based on the area in the image (in pixels) where the Face API detected a human face. It also gives a confidence score for each of the eight emotions that it can detect. For the example image shown in the figure, it is seen that the emotion "happiness" was identified with very high confidence. These two pieces of information provide the ability to generate a CAPTCHA challenge where the user is asked to answer what emotion the face in the presented image is expressing. Note that we could also ask the user to identify the face in the image, though this is correspondingly easy for automated attackers to do. In our implementation, a python script is used to interact with the API online and save the results it returns to a local SQL database, along with the image.

Figure 2 depicts an example of the test image served without noise or filters into Google's reverse image search. Note that the results of the search include the



```
Detection Result:
JSON:
[
  {
    "faceRectangle": {
      "left": 57,
      "top": 0,
      "width": 166,
      "height": 152
    },
    "scores": {
      "anger": 9.24406e-12,
      "contempt": 7.279144e-12,
      "disgust": 1.17872048e-13,
      "fear": 3.35413075e-9,
      "happiness": 0.999999762,
      "neutral": 1.31654772e-7,
      "sadness": 1.064573e-12,
      "surprise": 1.13628673e-7
    }
  }
```

Fig. 1. Example output from Emotion API



About 723 results (0.93 seconds)

Image size:
275 × 184

Find other sizes of this image:
All sizes - Medium - Large

Best guess for this image: ***dental smile***

Dental Smiles for Kids: Pediatric Dentist Ronkonkoma ...
www.dentalsmiles4kids.com/ ▾
Pediatric **Dentist** serving infants, children and teens in Ronkonkoma, Astoria and Whitestone, NY.

Dental Smiles Houston Dentist Office
dentalsmileshouston.net/ ▾
This site may be hacked.
Dental Smiles ... American Dental Association Greater Houston Dental Society Texas Dental Association Academy of Gp Orthodontics Texas Health Steps Gulf ...

Fig. 2. Useful information can be produced from default image

image at other dimensions, a keyword guess for what is depicted in the image (e.g. "dental smile"), and a number of visually similar images that all depict "dental smiles", which if one were to ask a person what emotion was being expressed, most likely "happiness" would be the response. Without introducing noise, distortions and filters to the image, an attacker will be able to answer the challenge question without much difficulty. Another straightforward attack would have been to submit it to the same Emotion API application that was used to annotate the challenge image and get the answer.

Figure 3 depicts an example chain of filters applied in a specific order to achieve the goal of altering the image enough that reverse image search (RIS), Google image search service (ISS), and computer vision based attacks cannot determine what is depicted in the image. The key is to introduce the minimal amount of distortion such that the tools used to create the challenges are stopped from returning meaningful results. Note that each filter often has multiple parameters that can be adjusted along a range to introduce variability into their output and how they affect the image. For testing, we have determined a series of fixed values for the filters that provide the level of distortion we required to stop the Computer Vision (CV) attack while still maintaining a reasonable level of usability/ease of understanding of the image.

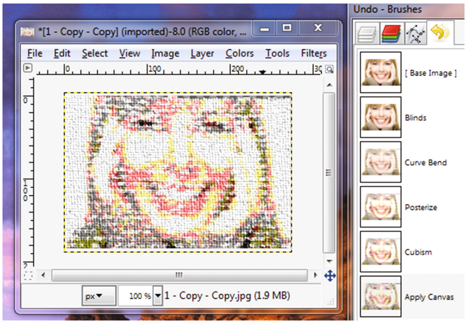

*[1 - Copy - Copy] (imported)-8.0 (RGB color, ...
File Edit Select View Image Layer Colors Tools Filters
Undo - Brushes
[Base Image]
Blinds
Curve Bend
Posterize
Cubism
Apply Canvas
px
100 %
1 - Copy - Copy.jpg (1.9 MB)

Fig. 3. Example of a series of filters applied to the image of Fig. 1

Figure 4 demonstrates how the appropriate level of distorts causes the ISS engines to return results that cannot be meaningfully used to attack the proposed CAPTCHA. Notice that the addition of the "canvas" filter effect influenced the ISS results towards needlepoint/grid based images - none of which focus on facial features. Also, no keyword is returned as well as no images of other sizes. For all intents and purposes, this distorted image is unique to the search engine, despite the original being indexed and tagged by it.

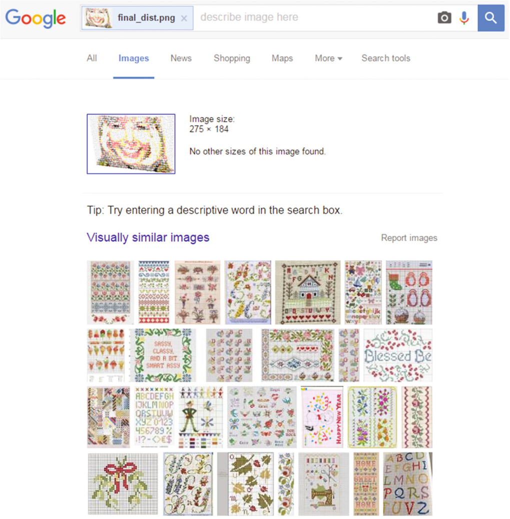


Image size:
275 × 184

No other sizes of this image found.

Tip: Try entering a descriptive word in the search box.

Fig. 4. No useful information can be extracted from noised image

Figure 5 is an example of the Twitter emoji set we used to map to the eight emotions provided by the Microsoft Emotion API. The initial build of our CAPTCHA included all these eight emotions. Microsoft noted that contempt and disgust were experimental emotions, and thus were usually not read

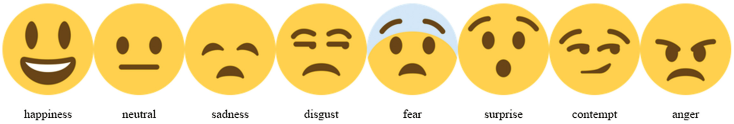


Fig. 5. Twitter Emojis used to represent the 8 emotions

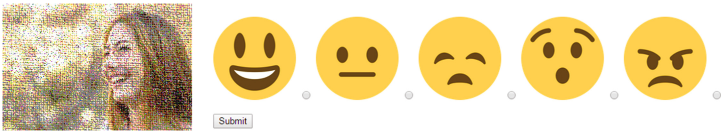


Fig. 6. Reduced emotion version of EmojiTCHA challenge

as accurately as the other six emotions. After an initial round of user testing, we decided to reduce the number of emotions that could be selected to the five emotions depicted in Fig. 6 in an effort to remove confusion and increase usability.

5 CAPTCHA Challenge Generation

This section focuses on the process flow within the toolchain that is used to generate the CAPTCHA challenges. Figure 7 shows the step by step process used to generate a unique challenge. Specifically, the following process is undertaken:

1. Gather images involving one or more people whose faces are clearly visible expressing one of the following eight emotions: anger, contempt, disgust, fear, happiness, neutral, sadness, or surprise. These images can be gathered from anywhere, e.g., using image search engine, downloaded from a camera, etc. They do not need to be manually tagged as the Emotion API will provide that information.
2. Each image is run through the Microsoft Project Oxford Emotion API to detect the number of faces expressing emotions in the image and the facial expressions that fall into one of the eight emotional categories. If at least one clear face is not found or no emotion can be read from the face by the algorithm, the image is discarded. If at least one clear face expressing one of the eight emotions is found, the image is kept and stored in a database.
3. The output of the Emotion API is recorded in the database along with the stored image. The output from the algorithm includes the face bounding box # (which face in the image the emotion information is from), the emotion expressed by the face, and the level of confidence as a percentage for the emotion expressed by the face.
4. The image is next run through a series of filters and manipulations from GNU Image Manipulation Program (GIMP) [12] to distort the image for protecting it against reverse image search attacks and computer vision tool

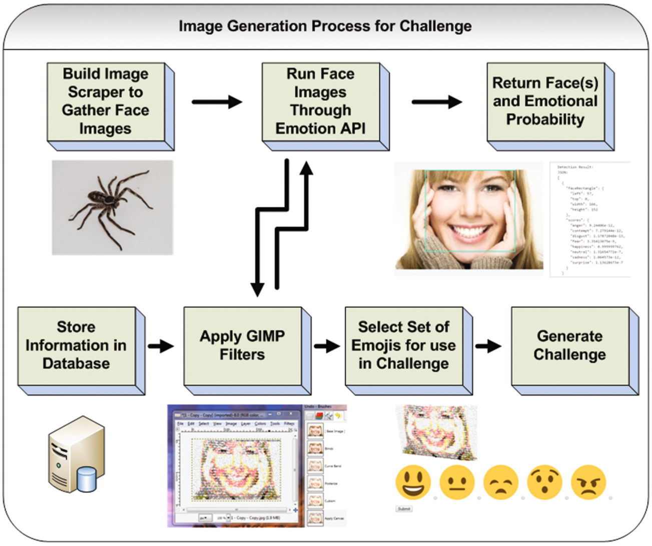


Image Generation Process for Challenge
Build Image Scraper to Gather Face Images
Run Face Images Through Emotion API
Return Face(s) and Emotional Probability
Store Information in Database
Apply GIMP Filters
Select Set of Emojis for use in Challenge
Generate Challenge
Submit

```
Detection Result:
JSON:
[
  {
    "faceRectangle": {
      "left": 57,
      "top": 0,
      "width": 166,
      "height": 152
    },
    "scores": {
      "anger": 9.24406e-12,
      "contempt": 7.279144e-12,
      "disgust": 1.17872048e-13,
      "fear": 3.35413075e-9,
      "happiness": 0.999999762,
      "neutral": 1.31654772e-7,
      "sadness": 1.064573e-12,
      "surprise": 1.13628673e-7
    }
  }
```

Fig. 7. Captcha generation process flow

based attacks. The number, type, and values for each of the filters used can be varied at random for each individual image produced to make it very difficult for attackers to filter the alterations. This step is important as it prevents using the tools that generate the challenge from being exploited by attackers. The output image is tested against the Emotion API to ensure that no emotion is meaningfully detected. The final altered image, now ready to be used as challenge, is stored in the database with the corresponding information used to create it. Figure 8 shows how filters used in a chain can successively distort an image until it meets the needed security criteria.

5. A set of emojis is selected such that one of them matches the emotion recorded from the Emotion API, and the remaining not matching the emotion, i.e., they would be incorrect/nonexistent responses.
6. The challenge is generated and the user is presented with one or more images of distorted faces and a corresponding set of emojis. The user simply needs to match the correct emoji with the facial expression in the image to complete the challenge. The correctness of the response is evaluated against the ground truth stored in the database.

6 EmojiTCHA Usability Study

The goal of this section is to evaluate the effectiveness and ease of use of EmojiTCHA. We conducted user trials with 30 participants and asked them to solve as many challenges as they could in 10 min. The first run included all eight emotions from the emotion engine. The user was served a challenge at random and

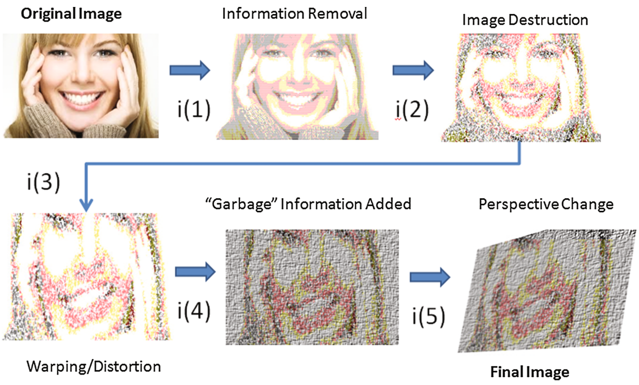
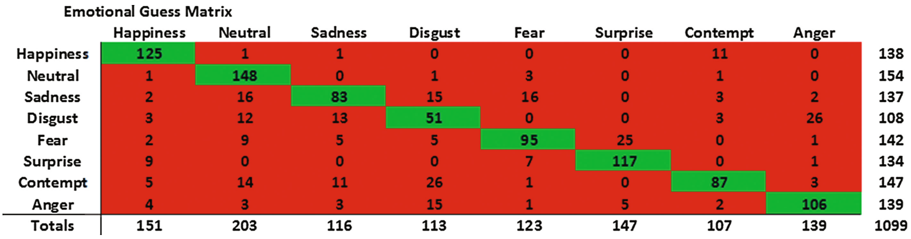


Fig. 8. Filter based distortion

asked to match the corresponding emoji to the emotion depicted on the face displayed. Each of the emotion categories had 10 images that were tagged and identified by the Emotion API. The image filters were applied at random until the image no longer returned a match from the Emotion API, thus some images were more distorted than others.



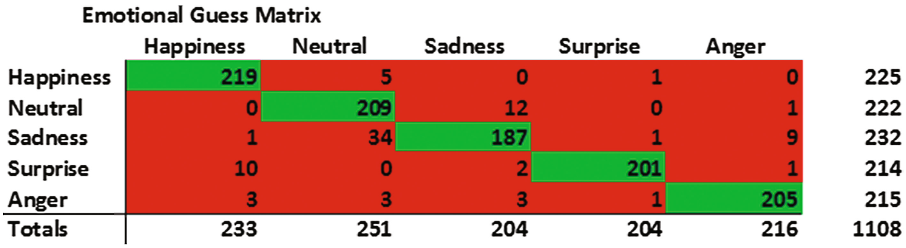
Emotional Guess Matrix

	Happiness	Neutral	Sadness	Disgust	Fear	Surprise	Contempt	Anger	
Happiness	125	1	1	0	0	0	11	0	138
Neutral	1	148	0	1	3	0	1	0	154
Sadness	2	16	83	15	16	0	3	2	137
Disgust	3	12	13	51	0	0	3	26	108
Fear	2	9	5	5	95	25	0	1	142
Surprise	9	0	0	0	7	117	0	1	134
Contempt	5	14	11	26	1	0	87	3	147
Anger	4	3	3	15	1	5	2	106	139
Totals	151	203	116	113	123	147	107	139	1099

Fig. 9. Confusion matrix for complete set of emotions

Figure 9 depicts the confusion matrix (i.e., the emotion guess results) for the first run with the complete set of eight emotions. The totals on the horizontal axis represent the number of times a challenge with the correct response being the emotion in green was served whereas the totals on the vertical axis represent the number of times that a particular emotion was given as a response for a challenge with the correct answer in green. Figure 10 gives the results for the second run with the reduced set of five emotions as options.

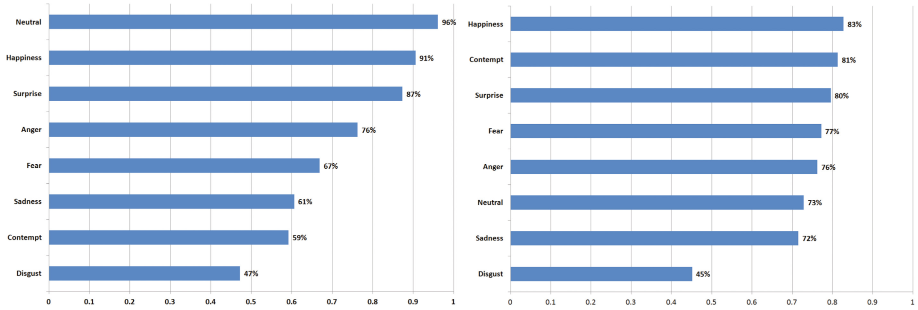
To make it easier to analyze the results we plot the precision and recall results for the different emotions in both runs. Figures 11a and b give the recall and precision for each emotion when all 8 emotions are used. The recall gives the percentage of images of each emotion correctly identified with that emotion whereas precision gives the percentage of images identified with a particular emotion that do actually have that emotion. The emotions from best performing to



Emotional Guess Matrix

	Happiness	Neutral	Sadness	Surprise	Anger	
Happiness	219	5	0	1	0	225
Neutral	0	209	12	0	1	222
Sadness	1	34	187	1	9	232
Surprise	10	0	2	201	1	214
Anger	3	3	3	1	205	215
Totals	233	251	204	204	216	1108

Fig. 10. Confusion matrix for reduced set of emotions



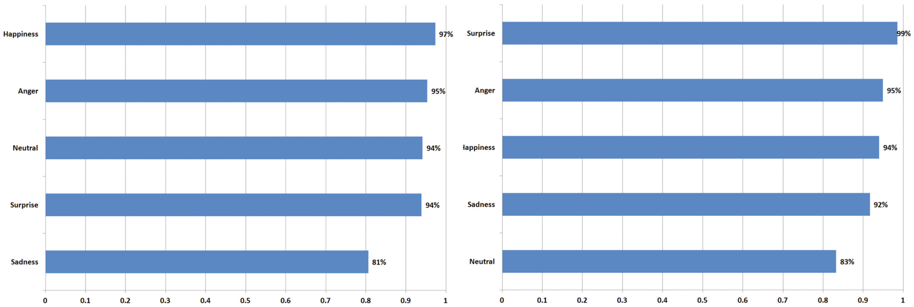
(a) Emotion Recall

(b) Emotion Precision

Fig. 11. Results with all 8 emotions

worse performing on recall are: neutral (96%), happiness (91%), surprise (87%), anger (76%), fear (67%), sadness (61%), contempt (59%) and disgust (47%). The emotions from best to worse performing on precision are: happiness (83%), contempt (81%), surprise (80%), fear (77%), anger (76%), neutral (73%), sadness (72%) and disgust(45%). Note that disgust was the worst performing emotion in both cases. It was expected that the more abstract of the universal emotions might be more difficult to discern for humans, namely, disgust and contempt. For recall, these emotions performed the worst, scoring significantly lower than the top three emotions. Interestingly, for precision, contempt was the second best emotion recognized, although the scores for the challenges were somewhat lower than the scores for recall, they were much more consistent across emotions, with disgust being the outlier. One aspect is the overlap of emotions that are consistently mistaken for another emotion that may appear similar. For example, we see that disgust was mistaken for anger 26 times. It is easy to imagine that a disgusted face can take a similar shape to an angry one. We also see this in contempt and anger being mistaken for disgust as well for 26 and 15 times respectively. More user testing will need to be conducted so that any set of emotions served to the user in a challenge will be ones that are not easily mistaken for each other. However, this can also provide a way to make it more difficult for machines – if a competing emotion detection algorithm is ranking a facial expression it is possible that it will score and categorize it differently than the Microsoft Emotion API.

Furthermore, most users were able to solve challenges in a very short duration. Additionally, after solving a few challenges, users are able to significantly increase their subsequent accuracy. To summarize, our study shows that EmojiTCHA in its current form is quite accessible to a wide range of users, with respondents coming from different continents. Figures 12a and b give the recall and precision when only 5 emotions are used in the challenges. The emotions from best performing to worse performing on recall are: happiness (97%), anger (95%), neutral (94%), surprise (94%), sadness (81%). The emotions from best to worse performing on precision are: surprise (99%), anger (95%), happiness (94%), sadness (92%), neutral (83%). The performance for all scores in both categories increased significantly in limiting the number of choices for the user to select. This shows that using a smaller set of emotions significantly improves usability of the system. Note that this does not significantly compromise on security since the challenge image still does not provide any results, and the probability of random guess is still 1/5 instead of the original 1/8.



(a) Emotion Recall

(b) Emotion Precision

Fig. 12. Results when only 5 emotions are used

We have also experimented with an alternative design which improves usability, but provides equivalent security to the original version. For example, Fig. 13 shows an example where only two emotions are used, but users are asked 3 challenges instead of 1. This keeps the security of random guessing to $(1/2)^3 = 1/8$ giving us equivalent security to the original single challenge with 8 emotions, but potentially is significantly easier for humans to answer.

7 Design Limitations and Security Analysis

The images that were chosen for use in the challenges were hand curated to ensure that the desired emotion was demonstrated in the image. Work is currently in progress to tune the image scraper and the Emotion API checker to

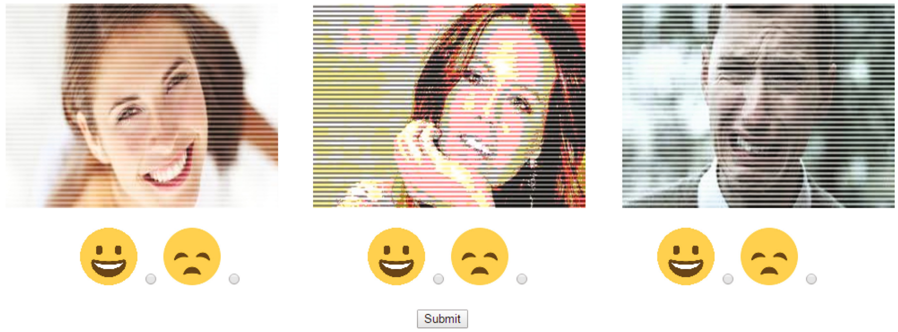


Fig. 13. Example with 3 challenges using 2 emotions each

accept an "emotion" threshold score as a percentage to ensure that there is a high degree of confidence in any particular emotion expressed in a face. Any images that have emotions that register below the threshold can be discarded accordingly. Additional work must be done to determine the optimal "co-emotions" to display with one another to ensure there is minimal mixup by between the emotions displayed on the screen. These will all be addressed in future iterations of the study.

Furthermore, in our current implementation, we only use a single emotion from a single face per challenge. Multi-face, multi emotion challenges are currently under development. We are also currently developing ordinality rules for filter application to minimize the number of rounds of filter applications that are required to ensure the security guarantees that CV attacks and RIS attacks will not be successful. The test images used in the experiments for the user study only provide the security guarantee for emotion API attacks - they do not ensure RIS or ISS attacks are not successful, although many of the images do indeed stop these attacks in their current form.

Note that the security of the EmojiTCHA depends on two different factors. First, is the ability of the attacker to successfully de-obfuscate the image, and then use the Emotion API to solve the challenge. Second, is the ability of the attacker to randomly guess the correct response. Regarding the first, the image is obfuscated using lossy filters which result in information loss which cannot be reversed. This provides a layer of security. Furthermore, we ensure that the obfuscated image is robust against reverse image search attacks, which are based on image similarity. Note, that this process can be adjusted as required, if improvements are made in cracking techniques, or in filtering techniques. Regarding the ability to randomly choose the correct response, the current 8-emotion version has a fixed probability of 1/8 for a correct random guess. However, as we have discussed above, reducing the number of emotions improves usability. At the same time, we can ask for a higher number of challenges. Similar to Fig. 13, if we ask three challenges, where the number of emotions is increased to three, then the probability reduces to 1/27, which is significantly stronger. We have actually implemented this, and are currently carrying out comparative usability testing. Note that since a fresh CAPTCHA is provided on every refresh, the possibility of a brute force attack is limited as long as sufficient images are included in the CAPTCHA database. Furthermore, since the noise addition is randomly carried out, noise can be added to the same image multiple times, resulting in fresh challenges.

8 Conclusions and Future Work

In this paper, we have designed a new CAPTCHA that is based on emotion recognition that has the advantage of being scalable and usable while providing good security. In the future, additional work around new form types and challenge questions will be experimented with. For example, testing out multiple emotions in a single image and having a user identify all of the emotions – a multi-answer CAPTCHA. Another example would be asking the user to identify the opposite emotion of that depicted in an image (e.g. pick sad if the image is showing a happy face). Additional work around creating a challenge with an "emotional mix" where a random set of 4 or 5 choices are selected for the challenge from the 8 possible choices. Finally, experimenting with a "not here" answer may be worthwhile to increase the security against a random guess being correct. We plan to work on these extensions in the future.

References

1. Microsoft Cognitive Services Emotion API (2016). https://www.microsoft.com/cognitive-services/en-us/emotion-api
2. Ahn, L., Blum, M., Hopper, N.J., Langford, J.: CAPTCHA: using hard AI problems for security. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 294–311. Springer, Heidelberg (2003). doi:10.1007/3-540-39200-9_18
3. Bursztein, E., Martin, M., Mitchell, J.: Text-based CAPTCHA strengths and weaknesses. In: Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS 2011), pp. 125–138. ACM, New York (2011)
4. Chow, R., Golle, P., Jakobsson, M., Wang, L., Wang, X.: Making CAPTCHAs clickable. In: Proceedings of the 9th Workshop on Mobile Computing Systems and Applications (HotMobile 2008), pp. 91–94. ACM, New York (2008)
5. Daher, W., Canetti, R.: Posh: A generalized CAPTCHA with security applications. In: Proceedings of the 1st ACM Workshop on Workshop on AISec (AISec 2008), pp. 1–10. ACM, New York (2008)
6. Datta, R., Li, J., Wang, J.Z.: Imagination: a robust image-based CAPTCHA generation system. In: Proceedings of the 13th Annual ACM International Conference on Multimedia (MULTIMEDIA 2005), pp. 331–334. ACM, New York (2005)
7. Elson, J., Douceur, J.R., Howell, J., Saul, J.: Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. In: Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS 2007), pp. 366–374. ACM, New York (2007)

8. Fidas, C., Hussmann, H., Belk, M., Samaras, G.: ihip: Towards a user centric individual human interaction proof framework. In: Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA 2015), pp. 2235–2240. ACM, New York (2015)
9. Golle, P.: Machine learning attacks against the Asirra CAPTCHA. In: Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS 2008), pp. 535–542. ACM, New York (2008)
10. Goodfellow, I.J., Bulatov, Y., Ibarz, J., Arnoud, S., Shet, V.: Multi-digit number recognition from street view imagery using deep convolutional neural networks. arXiv preprint arXiv:1312.6082 (2013)
11. Gossweiler, R., Kamvar, M., Baluja, S.: What's up CAPTCHA? A CAPTCHA based on image orientation. In: Proceedings of the 18th International Conference on World Wide Web (WWW 2009), pp. 841–850. ACM, New York (2009)
12. Lecarme, O., Delvare, K.: The Book of GIMP: A Complete Guide to Nearly Everything. No Starch Press, San Francisco (2013)
13. Nejati, H., Cheung, N.M., Sosa, R., Koh, D.C.I.: DeepCAPTCHA: an image CAPTCHA based on depth perception. In: Proceedings of the 5th ACM Multimedia Systems Conference (MMSys 2014), pp. 81–90. ACM, New York (2014)
14. Ross, S.A., Halderman, J.A., Finkelstein, A.: Sketcha: a CAPTCHA based on line drawings of 3D models. In: Proceedings of the 19th International Conference on World Wide Web (WWW 2010), pp. 821–830. ACM, New York (2010)
15. Rui, Y., Liu, Z.: Artifacial: automated reverse turing test using facial features. In: Proceedings of the Eleventh ACM International Conference on Multimedia (MULTIMEDIA 2003), pp. 295–298. ACM, New York (2003)
16. Shirali-Shahreza, S., Shirali-Shahreza, M.: Categorizing CAPTCHA. In: Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence (AISec 2011), pp. 107–108. ACM, New York (2011)
17. Sim, T., Nejati, H., Chua, J.: Face recognition CAPTCHA made difficult. In: Proceedings of the 23rd International Conference on World Wide Web (WWW 2014), pp. 379–380. ACM, New York (2014)
18. Wei, T.E., Jeng, A.B., Lee, H.M.: Geocaptcha: A novel personalized CAPTCHA using geographic concept to defend against 3rd party human attack. In: 2012 IEEE 31st International on Performance Computing and Communications Conference (IPCCC), pp. 392–399, December 2012
19. Woo, S.S., Kim, B., Jun, W., Kim, J.: 3DOC: 3D object CAPTCHA. In: Proceedings of the 23rd International Conference on World Wide Web (WWW 2014), pp. 397–398. ACM, New York (2014)
20. Yan, J., El Ahmad, A.S.: Usability of CAPTCHAs or usability issues in CAPTCHA design. In: Proceedings of the 4th Symposium on Usable Privacy and Security (SOUPS 2008), pp. 44–52. ACM, New York (2008)
21. Ye, Q., Doermann, D.: Text detection and recognition in imagery: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **37**(7), 1480–1500 (2015)
22. Zhu, B.B., Yan, J., Li, Q., Yang, C., Liu, J., Xu, N., Yi, M., Cai, K.: Attacks and design of image recognition CAPTCHAs. In: Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS 2010), pp. 187–200. ACM, New York (2010)