Face-Matching Based CAPTCHA: Generation and Analysis

Pracheta Phadnis[1], Tejali Patil[2], Pinki Maurya[3], Chitra Bhole[4]
KJ Somaiya Institute of Engineering & Information Technology, Mumbai, India
Email: pracheta.phadnis@gmail.com[1]
tejali1993@gmail.com[2]
maurya92pinki@gmail.com[3]
cbhole@somaiya.edu[4]

Abstract- In the age of internet, with resources such as email services, cloud services being freely available, misuse of the same is bound to happen. Completely Automated Public Turing Test to Tell Computers and Humans Apart or CAPTCHA are small tests that help identify the humans from the automated programs and bots that try to counterfeit humans. Text-Based CAPTCHAs are currently being widely used all over the internet. These are however vulnerable to a number of simple attacks. The Face-Matching Based CAPTCHA provides an efficient alternative to the Text-Based CAPTCHA by taking into consideration the human pattern matching capabilities and the property of psychovisual redundancy of the human eye.

Index Terms- CAPTCHA; Face-Matching; Pattern matching.

1. INTRODUCTION

CAPTCHAs are nothing but electronic tests that help identify humans from computers [1]. These are widely used by all the service providing websites while signing up for the service to ensure that automated programs do not get access to the service as such programs may bring the entire collaboration down. CAPTCHAs are designed such that they ensure to fulfill two properties; human ease being one, script/bot difficulty being the other.

Various types of CAPTCHAs were developed over time, the most primitive ones being based on OCR manual. These made use of the distortions mentioned in the manual that failed OCR [4]. GIMPY and EZ-GIMPY CAPTCHA based on simple and distorted dictionary words were another form of CAPTCHA to be developed. Baffle-Text uses non English pronounceable words to form the CAPTCHA [6]. iCAPTCHA is another type of CAPTCHA that is based on user interactions. It uses the time required for messages from user to pass back and forth as a parameter [9]. One other method uses hand written words instead of printed characters [5]. All these types of CAPTCHA however fall into one broad category called the Text-Based CAPTCHA. While being efficient to a certain degree, these forms also possess some major shortcomings. All these techniques involve text characters. This introduces the problem of language dependency. The user has to identify with the characters of the language, only then can he/she solve the CAPTCHA. Another issue could be the easy attacks such as segmentation attacks where each character is separated and recognized.

To overcome these problems, CAPTCHAs involving images surfaced. Object recognition CAPTCHAs were developed which were based on object identification. Similarly, face recognition CAPTCHA [3] was developed where user identified two distorted images of same person. Another CAPTCHA [4] required the user to identify pair of faces of same person.

The proposed CAPTCHA is fundamentally based on [4]. The CAPTCHA takes into consideration the pattern matching abilities of human. It also considers the psychovisual redundancy property of the human eye.

2. IMPLEMENTATION DETAILS

Face-Matching CAPTCHA aims at fulfilling the AI-Completeness property of problems which states that a problem is AI-complete if it cannot be solved by computers alone. It does so by embedding images on random background and introducing distortions.

The general flowchart for the generation is as shown in the Figure 1:

The generation of the Face-Matching CAPTCHA consists of four major steps of background generation, non-face noise stitching, face image stitching, random line addition. A database of non-face and face images is used during the generation. These steps are briefly described below:

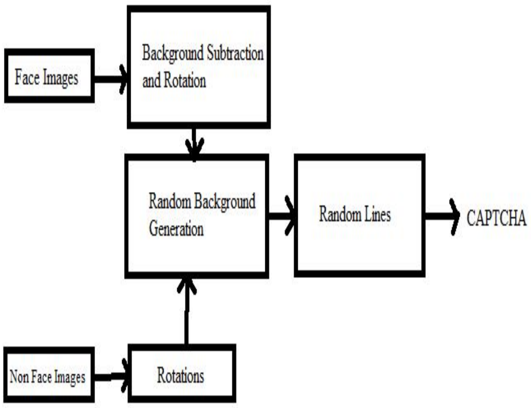

Face Images
Background Subtraction and Rotation
Random Background Generation
Random Lines
CAPTCHA
Non Face Images
Rotations

Fig 1: Flowchart for CAPTCHA generation.

2.1 Random Background Generation

An image, random in nature is generated first which would act as the background or the base image for the CAPTCHA. The image has dimensions 250x500 pixels with each pixel initialized each time to a random RGB value. The resulting image would look like the one shown in Figure 2.

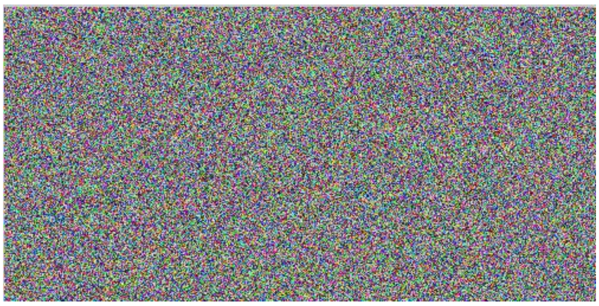


Fig 2:Random Background Image.

2.2 Non-face Noise Stitching

Noise elements add to the random nature of the background. This randomness is useful in confusing the face detector algorithms which may try to detect the faces in the CAPTCHA. The algorithm sets the number of images to be picked from the dataset to a value greater than four, which means, there are at least five non-face images to be chosen. It then choses as many images as needed from the dataset at random.

Each image then undergoes rotation by a random degree. The rotated images are then stitched at random positions on the background image. The resulting image is as shown in Figure 3.

2.3 Face Image Stitching

The face images form the essence of the CAPTCHA. It is by recognizing two matching images from these that the user would solve the CAPTCHA.

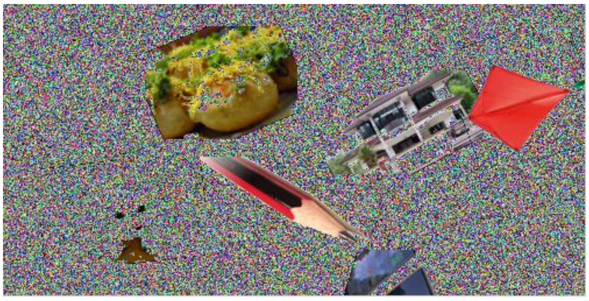


Fig 3:Image after stitching non-face noise.

Similar to the non-face noise images, the algorithm sets the number of face images to be chosen to be a random value greater than three, which means there would be at least four face images. Once the number is set, those many images are chosen at random from the face image dataset.

Each chosen image undergoes two stages now, face detection and cropping and background subtraction and rotation. The images are passed through the standard Viola-Jones face detector. The detected face is then cropped to form a new image. The cropped image then undergoes rotation by a random degree. It then undergoes background subtraction where each pixel's intensity is checked with the background intensity and if the value lies within a threshold, that pixel is eliminated. When this is done, certain data of the image is lost, but due to the pattern matching abilities and property of psychovisual redundancy of human eye, humans' recognition is not affected.

The resulting image is then stitched on the image obtained in the previous step at random positions.

When the images are chosen, one among them is duplicated which forms the answer for the CAPTCHA. When these images are stitched onto the background, the positions of the answer, that is, the duplicate images are noted down. The resulting image is as seen in Figure 4.

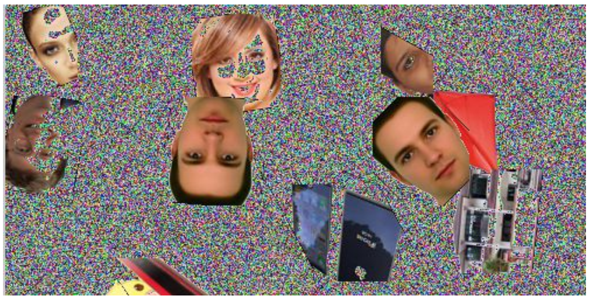


Fig 4: Image after face image stitching.

2.4 Random Line Addition

To increase confusion, the stitched image undergoes a final step of random line addition thus completing the generation process. In this stage, a maximum of hundred random lines are added or drawn at random positions on the stitched image. The resulting image is as seen in Figure 5.

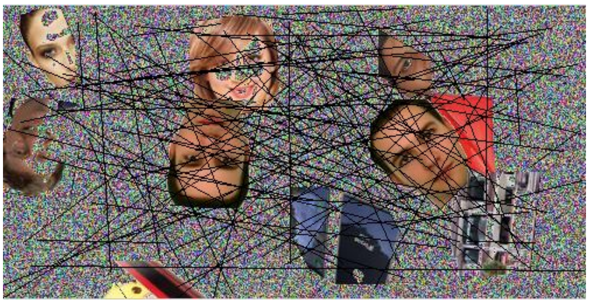
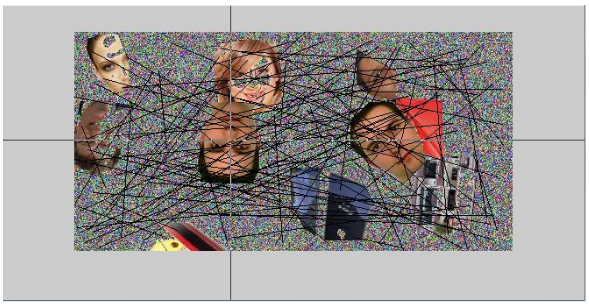
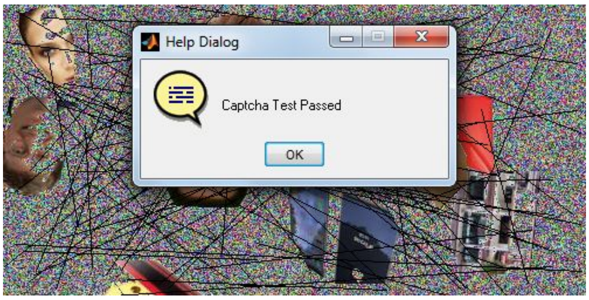


Fig 5: Final CAPTCHA after random line addition.

This forms the face-matching CAPTCHA. To solve the CAPTCHA, the user has to simply click on the pair of identical/matching human faces. If the users click points lie within a certain window of the previously noted answer, the algorithm identifies the user to be a human user, that is, the CAPTCHA test is solved. This can be seen in the Figure 6.





Help Dialog

Captcha Test Passed

Fig 6: Face-Matching CAPTCHA.

3. RESULT ANALYSIS

The algorithm was tested and responses from several users were recorded. The accuracy of the CAPTCHA is calculated using the standard accuracy formula as mentioned in “Eq. 1”. Here passing of the test means the instance when the user correctly identifies the identical pair. The test cases are all the samples generated by the algorithm which were used for testing:

$$Accuracy = \frac{Total\ Number\ of\ Tests\ Passed}{Total\ Test\ Cases} \times 100$$

$$Accuracy = \frac{Total\ Number\ of\ Tests\ Passed}{Total\ Test\ Cases} \times 100$$

$$Accuracy = \frac{Total\ Number\ of\ Tests\ Passed}{Total\ Test\ Cases} \times 100$$

To calculate accuracy, 200 responses from 100 users were collected. The accuracy was found to be 93%. During the testing, the simple GIMPY CAPTCHA accuracy for humans was also tested. It was found to be 99%. For a simple automated algorithm based on segmentation [2] and dictionary matching, the accuracy was found out to be 75%.

To test the newly generated CAPTCHA’s effectiveness, the CAPTCHA was tested against two different face detectors. The Viola-Jones standard detector that uses Haar-features could not detect any faces. Same was the case with Picasa face detector.

The CAPTCHA was also passed through a skin detector. The detector uses the HSV colour model and extracts the Cb and Cr components from the image. These components are then compared with threshold values and those within the threshold are returned as skin components. These components appear in form of blobs. The blobs are separated and two blobs with similar properties such as dimensions are considered as candidate solutions. The corresponding pixels from the CAPTCHA are then fetched and they undergo eye and face detection again. Based on the results of these detections, they are selected or rejected as answers. Using this test, the accuracy of solving the CAPTCHA is 10%.

This 10% accuracy can be further reduced by using noise images that are close to skin tones. This would reduce the probability of detection by increasing the choices for detection.

The accuracy findings are listed in Table 1 and Table 2.

Table1: Accuracy findings for GIMPY text CAPTCHA.

Human	Program
99%	75%

Table 2: Accuracy findings for Face-Matching CAPTCHA

Human	Viola-Jones	Picasa	Skin Model
93%	0%	0%	10%

4. CONCLUSION

With fast developing technologies, the internet being more and more vulnerable, traditional text based CAPTCHAs may fall short in providing extremely high levels of security. The proposed CAPTCHA would be efficient as it makes use of human properties of complex pattern matching and psychovisual redundancies. This CAPTCHA also eliminates language dependencies, location dependencies found in other CAPTCHAs. The accuracy for humans is also on a higher side whereas that for machines is quite low thus making it more suitable to be used.

5. ACKNOWLEDGEMENTS

The authors would like to thank all the people who volunteered for the testing of the project as well as all those who made useful suggestions.

6. REFERENCES

[1] Bursztein E; Martin M; Mitchell J. C(2011), *Text-Based CAPTCHA Strengths and Weaknesses*, CCS'11, Chicago, Illinois, USA, October 17–21.`

[2] Chandawale A. A; Jalnekar R. M; Sapkal A. M(2009), *Algorithm to Break Visual CAPTCHA*, Second International Conference on Emerging Trends in Engineering and Technology, ICETET-09, 978-0-7695-3884-6/09.

[3] Gaj K; Misra D(2006), *Face Recognition CAPTCHAs*, Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW 2006).

[4] Goswami G et.al.(2012), *Face Recognition CAPTCHA*, IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS).

[5] Govindaraju V; Rusu A(2004), *Handwritten CAPTCHA: Using the difference in the abilities of humans and machines in reading handwritten words*, Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004).

[6] Ince I. F; Salman Y. B; Yengin I(2008),*Designing Captcha Algorithm: Splitting and Rotating the Images against OCRs,* Third 2008 International Conference on Convergence and Hybrid Information Technology.

[7] Izquierdo E; Macias C. R(2011), *Image CAPTCHA based on Distorted Faces*

[8] Malik J; Mori G(2003), *Recognising objects in*
Adversarial clutter: breaking a visual CAPTCHA, IEEE Conference on Computer Vision & Pattern Recognition (CVPR), 2003, IEEE Computer Society, vol. 1, pp.I-134-I-141, June 18-20.

[9] Truong H. D; Turner C. F; Zou C. C(2011), *iCAPTCHA: The Next Generation of CAPTCHA Designed to Defend Against 3rd Party Human Attacks*, IEEE ICC proceedings.

[10] http://www.captcha.net/