Online CAPTCHA replacement through Face Detection

Rutwik Chinchole, Aditya Kakad, Sumit Bhirud, Aakarsh Raghunath and Ms. Snehal Mumbaikar
University of Mumbai

Abstract – **The sensitive information stored on the internet should not be easily accessible by any unauthorized personnel, it should be readily available only to the authorized people. Thus integrity, security and confidentiality need to be maintained. There needs to be a security check before any kind of access is granted to any kind of sensitive information. CAPTCHA (Completely Automated Public Turing Test to tell Computers and Humans Apart) is a popular way for authenticating a user. The main goal of any technique is to recognize whether it is a human who is accessing the content or a robot via some recognition techniques. The advancement in recognition technologies has enabled humans to develop bots which can now bypass common authentication techniques which have been in use for a long time. CAPTCHA has been in use since a very long time and has some serious drawbacks. Facial recognition alongside with artificial intelligence can overcome drawbacks like difficulty in solving, ambiguity, and sometimes easy prey for trained bots of CAPTCHA and can easily replace it.**

Index Terms – CAPTCHA, RESTful API, Face Detection, Key points, Accuracy.

1. INTRODUCTION

CAPTCHA means Completely Automated Public Turing Test to tell Computers and Humans Apart. Recognizing both familiar and unfamiliar faces is what humans are good at and that has been established [1], [2]. The CAPTCHA method can be easily replaced by Facial Recognition proposed in this paper.

For differentiating humans captcha focuses on providing the user with different tests[3]. Various services including web and financial services use Captchas to provide security measures against malicious attacks[6]. OCR (Optical Character Recognition Manual) was used to extract text in AltaVista Captcha with distortions known to reduce incorporated OCR accuracy[4]

CAPTCHA is nearing its end and is in serious need to be replaced. Specialized tests like Handwritten character recognition and image or pattern recognition tests are used to determine is it really a human who needs to access the data[5]. Captcha has various forms and each one has a unique way of identifying. Although CAPTCHA has a series of tests to determine the differences it has some real serious drawbacks which can be overcome by using various efficient alternatives. In this paper we propose a technique of using Facial Recognition to detect whether it's a human or a robot accessing the content.

2. CURRENT SYSTEM AND LIMITATIONS

CAPTCHA is a system which can date back to the twentieth century, One question was needed to be answered by Alan Turing - Can Human thinking be mimicked by computer? The goal of any CAPTCHA system is to make challenges and ask questions such that the computers won't be able to pass them but humans can easily get through with it. Various types of Captcha were evolved during its usage and each one introduced some new challenges along with various limitations which can be seen below[7].

Sr. No	Existing Systems	Limitations
1	Honeypot	The form can be mistakenly filled by a human. Once a bot figures out that the form does not need to be filled, it will learn and overcome.
2	TextCAPTCHA	It asks a simple question, which can easily be answered by an NLP bot, rendering it extremely unsafe.
3	Reject Submission	If the form is small and the user has enabled auto-fill extension on the browser, this form will always reject the input.
4	Math Question	NLP bots can easily solve the math problem with the advancement in IBM Watson.

Table 1. Existing system limitations

3. PROPOSED WORK

A computer program or system intended to distinguish human from bot/machine input, typically as a way of automated extraction of data from websites and thwarting spam. This is done with help of CAPTCHA. An algorithm that cracks CAPTCHAs with 90-99.8 percent accuracy is also claimed. Therefore to improve on the security, process of online CAPTCHA verification for websites need to replace which is used to identify whether the access to the website is by a human or through a bot and to rectify issues with the current captcha technology. The face recognition technique is an alternative solution for this.

4. PROPOSED SYSTEM FLOW

The image in Figure 1 shows the simple system architecture the proposed face detection system employees, with the entire system stored on aws for wide availability.

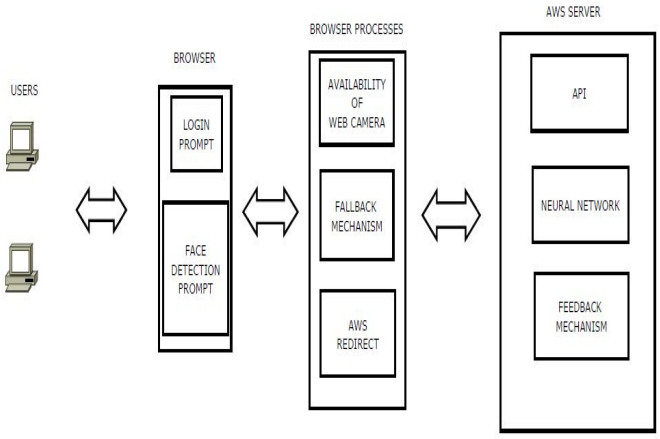

Figure 1. System Architecture

1. First we prompt the user for the verification process.
2. The user allows the image to be captured from the web camera.
3. Once the image is captured, and all the other details are verified, the image is sent to the backend using an API. The RESTful API is used to send the image through the http protocol.
4. Once the image is received in the backend (AWS), the face detection code starts to implement.
5. First the number of faces in the image is detected.
6. If the numbers of faces are greater than 1, prompt is given to the user to capture another image. This is for security reasons that only a single face will be used for authentication.
7. On the successful face detection further access granted.
8. If the face detection is unsuccessful, user will get prompt for another attempt.
9. The number of attempts is restricted to 2 in our system. After this, the regular captcha system will be implemented for the user.
10. In case of unavailability of the web camera/front camera, the system will automatically fall back onto the captcha system.

5. PROCESS USED FOR FACE RECOGNITION

The face detection process is having multiple stages as described below.

5.1 Detect Faces Using a Haar Cascade Classifier

- We have used OpenCV's implementation of Haar feature-based cascade classifiers to detect human faces in images. OpenCV provides many pre-trained face detectors, stored as XML files.

5.2 Add Eye Detections

- A Haar-cascade eye detector can be included in the same way that the face detector was. This allows the image under detection to include eyes as a keypoint for verification.

5.3 De-noise an Image for Better Face Detection

- Using OpenCV's built in color image de-noising functionality called fastNlMeansDenoisingColored -we de-noise this image enough so that all the faces in the image are properly detected. Once we have cleaned the image, we run our trained face detector over the cleaned image to check out its detections.

5.4 Create a CNN to Recognize Facial Keypoints

- We notice that facial key point detection becomes a regression problem at high levels when we include 15 key points, thus we employ a convolutional neural network to recognize the patterns in the images.

- We need to train a regressor, and for that, we need a training set of images, a set of facial images and keypoint pairs to train on. We use this dataset from Kaggle.

Figure 2. Visualization of the subset of training data.

5.5 Compile and Train the Model

- Figure 2 shows the visualization of training data and after we visualize the data and check that the correct keypoints are being mapped, we start to train the model

- We first experiment with the choice of optimizer to use. After testing, the best fitting optimizer is the Adam, which we implement. We use the 'fit' method to train the model.

5.6 Visualize the Loss and Test Predictions

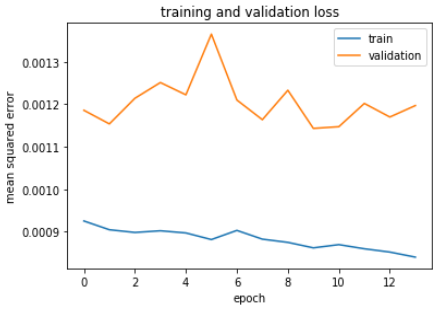

Figure 3. Training and Validation Loss

- We plot the training and validation loss data as shown in Figure 3 to check where the model starts to overfit or underfit.
- This would help us to determine the number of epochs and manipulating the dropout layers to avoid over fitting and underfitting the model.

5.7 Visualize a Subset of the Test Predictions

- For each training image, there are two landmarks per eyebrow (four total), three per eye (six total), four for the mouth, and one for the tip of the nose.
- The below Figure 4 shows visualization of predictions on test data

Figure 4: Visualization of test predictions.

6. RESULTS AND ANALYSIS

After performing initial tests on both the systems, using a timer function, we were able to determine a result from the time consumed for the entire process. The concluding results showed that the captcha standalone system was unreliable in terms of efficiency with respect to time. From an accuracy stand point, the CAPTCHA system was vulnerable to bots. Bots were able to figure out the various captchas and go through the security clearance without hinderance.

We conducted numerous tests on different environment. The environment variables included the lighting of the environment, the light on the face (visibility), background lighting, and internet connection. We identified that these were the variables that affected the accuracy and the success rate of the model, and fluctuating these conditions caused fluctuations in the accuracy of face detetction.

Sr. No	Environment	Result
1	Optimal : Well lit, face clearly visible, no background light, Strong internet connection.	No of tests : 15 Successful : 14 Successful in 1st attempt : 12 Successful in 2nd attempt : 2 Not successful : 1 Success percentage : 93.33
2	Good : Fairly lit, face visible, no background light, mid range internet connection.	No of tests : 15 Successful : 12 Successful in 1st attempt : 11 Successful in 2nd attempt : 1 Not successful : 3 Success percentage : 80
3	Strained : Dimly lit, face visible (low light), background	No of tests : 15 Successful : 8

	lights, fair internet connection.	Successful in 1st attempt : 3 Successful in 2nd attempt : 5 Not successful : 7 Success percentage : 53.33
4	Bad : Poorly lit, face blurred or blackened due to poor lighting, extreme background light, fluctuating internet connection.	No of tests : 15 Successful : 2 Successful in 1st attempt : 0 Successful in 2nd attempt : 2 Not successful : 13 Success percentage : 13. 33

Table 2. Tests and Results

Sr No.	Parameters	CAPTCHA	FACE DETECTION
1.	Speed	**15-42 Seconds**	**4-9 Seconds**
2.	Accuracy	**Takes attempts anywhere from 1-6.**	**Maximum of 2 attempts.**
3.	Security	**Provides security but vulnerable to bots.**	**Provides more security as it is difficult to read images**
4.	Optimization	**No possible way for optimization**	**Additional image inputs will improve the accuracy of the system.**
5.	Efficiency	**50-90%**	**Approximately 94%**

Table 3. Analysis of the systems

The tests to determine the accuracy of the system were based on variables that influenced the face detection. The main variable which we figured out were environment lighting, how well the face was lit, and how good the internet connection sustained. The best case scenario took the accuracy up to 94%, with access provided in as little as 4 seconds. In a fairly lit environment, the accuracy dropped a little but still gave a high percentage of 80, with time required anywhere from 4 to 10 seconds. This was faster than the CAPTCHA systems in place.

7. CONCLUSION

After conducting various tests in controlled environments, the results gathered showed that it was highly beneficial to use Face Detection in most of the cases. It provided with faster results and better accuracy and security, as it is difficult to bypass a face detection system through conventional bot development. Our system would be highly beneficial for people who are not able to solve a CAPTCHA due to unfortunate literacy incapability, and also for people suffering a language barrier on the internet.

8. REFERENCES

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

[2] P.Sinha,B.Balas,Y.Ostrovsky,andR.Russell.Facerecognition by humans: Nineteen results all computer vision researchers should know about. Proceedings of the IEEE, 94(11):1948 –1962, 2006.

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

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

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

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

[7] May, Matt (2005-11-23). "Inaccessibility of CAPTCHA". W3C. Retrieved 2015-04-27.