CAPTCHA Cracker using MATLAB

Kunal Kathrani
Department of I.T.
K. J. Somaiya College of Engineering

Piyush Agrawal
Department of I.T.
K. J. Somaiya College of Engineering

Khushi Khanchandani
Department of I.T.
K. J. Somaiya College of Engineering

ABSTRACT

There is a lot of activity over the internet. Be it, posting a comment on someone's blog or making a Gmail account or booking a ticket online. But with that also comes the problem of spamming. "Completely Automated Public Turing test to tell Computers and Humans Apart" (CAPTCHA) [8] which are the twisted words that block the entries of bots on website. CAPTHCAs can effectively test if the user is human or machine. Hence it is of great importance that CAPTCHA is well checked for its vulnerability against such attacks. So this paper presents this medium to check the strength of CAPTCHA against the written CAPTCHA cracking code. This can be used by the web developers implementing CAPTCHA, to check well in advance how secure is the CAPTCHA used in their software.

General Terms

Security, Image Processing.

Keywords

CAPTCHA, OCR, Image Processing, MATLAB, Turing Test, CAPTCHA Cracking.

1. INTRODUCTION

In last few years, internet has witnessed brute force attacks as spammers developed bots to access websites and increase the load on the servers. This situation has caused new challenges and it demands the use of stronger CAPTCHAs. The plan is to crack these CAPTCHAs using "Image Processing". The future scope will include a testing module where it plans to test the complexity of CAPTCHA and hence assess the web developer in providing proper security measures for the website.

2. RELATED WORK

CAPTCHA has been cracked by several organisations in the past with same motive to achieve higher security and stronger CAPTCHA sets [7]. The paper titled as 'Stanford Researchers crack CAPTCHA code' by Todd Wasserman published by Stanford University has created DeCAPTCHA, software that makes CAPTCHA readable by computers by cleaning up the text and rendering them in legible letters and numbers. The tool decodes CAPTCHA most, but not all the time. The team was able to decode CAPTCHA up to 66 percentage accuracy. The paper titled as breaking an image based CAPTCHA by Michael Merler, Jacquilene Jacob published by Stanford University is on Vidoop CAPTCHA. It is a verification solution that uses images of the objects, and animals, people, instead of distorted text to distinguish a human from computer program [10]. What the authors underestimate that since a bot can try to access a service thousands of times a day, recognition rates which are considered quite low by the object recognition community.

3. PROBLEM STATEMENT

The idea of the project is to break a text based CAPTCHA and to show that text based CAPTCHA are not highly secure. Currently the technology involved in cracking the CAPTCHA is not very accurate and has high processing time. But for a company implementing security using CAPTCHA they should know what level of security they are having by using a particular type of CAPTCHA. The software will take the screenshot of the website, crop the image to required area, process the image and return the answer. So basically the software will try to crack their CAPTCHA and then notify them the level of security they are having..

4. SOLUTION APPROACH AND METHODOLOGY

4.1 System Architectural Design

Chosen System Architecture

The system architecture designed for the proposed system is as follows: The components of the system architecture are described in detail as follows:

Data Collection: This component is primarily focused on acquisition of data which consists of collection of CAPTCHA images.

Decision making: This component deals by first takes the CAPTCHA image processes it and returns the text in it.

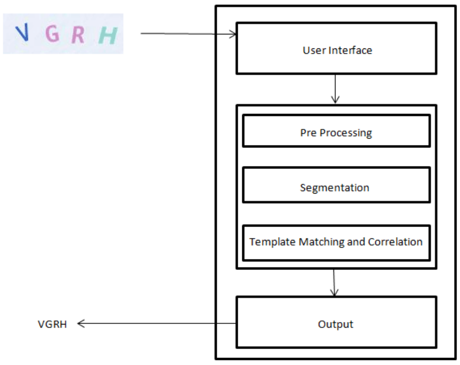


Figure 1: System Architecture

4.2 Detailed Description of Component

Data collection: This component primarily deals with the collection of the input data that is the list of all the CAPTCHA images. The images are then processed in the processing stage.

Data processing: The user should be feeding the link of the webpage for which the CAPTCHA needs to be cracked. The application will be able to extract CAPTCHA as an image. The application will process the image and recognise characters of the CAPTCHA. The application will provide the answer.

4.3 Discussion of Alternative Designs

Alternative designs can be implemented for the proposed system. It can use the cut point detector to find all the **possible** cuts along which to segment CAPTCHA into individual characters and then slicer for getting some meaningful slices then scorer and arbiter for getting the text out of the CAPTCHA [7].

4.4 Component Diagram

Output interface: The final output will show the result which will be access granted or access denied

Files: Source files, executable files, database files.

Libraries: **MATLAB** libraries.

4.5 Decision making

This component merely decides whether the given CAPTCHA is cracked or not and if it is cracked it is not secured and **hence** not suitable for the website. This component interacts with the software, the data processing unit and the user.

4.6 Use Case Diagram

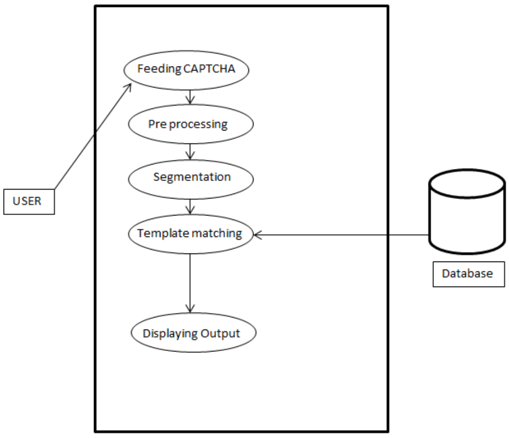
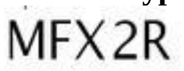


Figure 2: Use Case Diagram

4.7 Algorithm to crack a type of CAPTCHA



Preprocessing

The CAPTCHA image is converted to gray scale image if it's a RGB image. After converting RGB image to gray scale image, the threshold method is performed. First it **calculates** the threshold value using predefined function graythresh().This threshold value is used, where gray-levels below this threshold is said to be black and levels above are said to be white [1].It performs binarization for further segmentation [6].

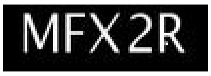


Figure 3: Preprocessed image

Segmentation

Segmentation is the isolation of characters or words. The majority of optical character recognition algorithms segment the words into isolated characters **which** are recognized individually as shown in figure 4. This segmentation is performed by isolating each connected component that is each connected white area. In matrix term each connected 1's can be termed as one segment as shown in figure 5 [2].

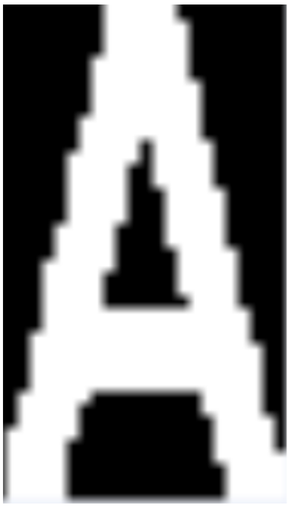
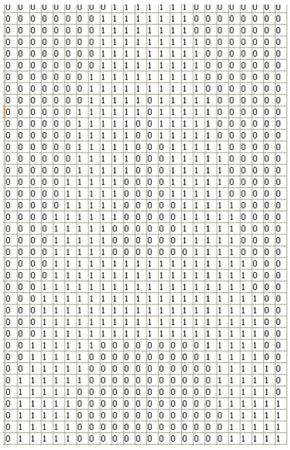


Figure 4: Segmented character



0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Figure 5: Segmented character in form of matrix

Template-matching and correlation techniques

These techniques are different from the others in that no features are actually extracted. Instead the matrix containing the image of the input character is directly matched with a set of characters in templates representing each possible class. The distance between the pattern and each character in template is computed, and the maximum correlation value obtained from segmented character and the characters in template is the character printed to output [2].

Cracking of second type of CAPTCHA

This type of CAPTCHA suffers from several weaknesses, be it fixed font face, fixed font size, no distortions, trivial background noise and it's easy to segment. Here the three step algorithm is used to break the CAPTCHA. The image is preprocessed to remove noise using threshold method.

Thereafter a simple cleaning technique is used to clean the noise. Then the CAPTCHA is segmented using vertical projections and candidate split positions. Four classification methods have been implemented which are pixel counting, vertical projections, horizontal projections and template correlations. The system was trained on a sample of twenty CAPTCHAs to create thirty-six training templates one for each character (0-9 and A-Z). The following success rates have been achieved using the different classifiers: 8% pixel counting, vertical projections 97%, horizontal projections 100%, and template correlations 100%.

Algorithm to crack this CAPTCHA

Steps involved are as follows.

Making of template

Firstly save the CAPTCHA images with file name as its output. Load all the **images** from the training directory. Then perform preprocessing and segmentation as described below. Now the segmented characters of training CAPTCHA are mapped with corresponding characters of file name and saved into template database.

Preprocessing

The CAPTCHA is fed in the CAPTCHA cracker software and converted to gray scale as show in figure 6 using MATLAB inbuilt function rgb2gray(). Further the CAPTCHA image is threshold as show in figure 6 and image is cleaned of noise present in it as shown in figure 9. The final output of this step is preprocessed image.

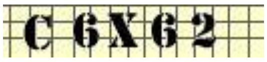


Fig 6: Original Image

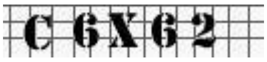


Fig 7: Grey Scale Image

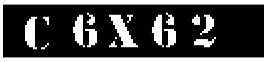


Fig 8: Threshold Image

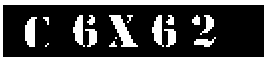


Fig 9: Further Cleaned Image

Segment

The preprocessed CAPTCHA block is segmented into individual characters as shown in figure 10. Detailed explanation of this step is as follows. Firstly the size of preprocessed image is obtained, then every individual column is scanned till the data is found. Then the obtained character is cropped. Twenty rows and columns are padded with 0's as shown in fig 11.

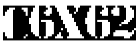


Fig 10: Segmented Image

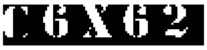


Fig 11: Padded Image

Classify

Firstly the templates are loaded then it can use four types of classification techniques as follows:

Pixel Count

First, the vertical segmentation divides the characters into 5 segments. Next each segment is scanned to get the number of foreground pixels in it. Then, the pixel count obtained in the previous step is used to look up the mapping table [11.. Finally it gives the output with success rate of 8%.

Vertical Projections

Vertical projection is applied to a segmented image, each of which contains one character. The process of vertical projection starts by mapping the image histogram to that of the template vertical histogram which finally has more mapping involved to it is the output. Finally it gives the output with success rate of 95% [3].

Horizontal Projections

Horizontal projection is applied to a segmented image, each of which contains one character. The process of vertical projection starts by mapping the image histogram to that of the template horizontal histogram which finally has more mapping involved to it is the output. Finally it gives the output with success rate of 100%.

Template Correlations

The matrix containing the image of the input character is directly matched with a set of characters in templates representing each possible class. The distance between the pattern and each character in template is computed, and the maximum correlation value obtained from segmented character and the characters in template is the character printed to output. Finally it gives the output with success rate of 100%.

5. CONCLUSION

As we know increase in number of CAPTCHA usage in every web services. As CAPTCHA is termed as secure to prevent DOS attacks as it avoids running of automated scripts.

Finally the conclusion by cracking the CAPTCHA is that it is not completely secure. This paper shows the technique to crack CAPTCHA so that one can implant CAPTCHA cracker by giving CAPTCHA image as an input and retrieving CAPTCHA text as output.

6. FUTURE SCOPE

An automatic software can be made which will auto detect the CAPTCHA in the website and will feed the CAPTCHA to the website automatically. The solutions can also be provided to them about which security levels they can include in their CAPTCHA.Automated learning can also be given to the CAPTCHA cracker which is supported by human so that multiple training sets can be created.

7. ACKNOWLEDGMENTS

Our sincere thanks to our project guide Mrs. Khushi Khanchandani Ma'am who has been a great mentor for the complete project and the completion of our technical paper.

8. REFERENCES

[1] Mr.Nithya.E and Dr. Ramesh Babu D R June 2013 OCR System for Complex Printed Kannada Characters.

[2] Sandeep Tiwari, Shivangi Mishra, Priyank Bhatia, Praveen Km. Yadav May 2013 Optical Character Recognition using MATLAB.

[3] Jeff Yan and Ahmad Salah El Ahmad A Low-cost Attack on a Microsoft CAPTCHA.

[4] Silky Azad & Kiran Jain, CAPTCHA: Attacks and Weaknesses against OCR Technology

[5] Prof. (Mrs.) A.A. Chandavale and Prof. Dr.A.M. Sapkal and Dr.R.M.Jalnekar ,Algorithm To Break Visual CAPTCHA.

[6] Hina Parveen and Sudhir Singh, Captcha Recognition and Robustness Measurement using Hybrid Approaches

[7] Elie Bursztein, Jonathan Aigrain and Angelika Moscicki The End is Nigh: Generic Solving of Text-based CAPTCHAs

[8] Christoph Fritsch, Michael Netter, Andreas Reisser, and Günther Pernul, Attacking Image Recognition Captchas A Naive but Effective Approach

[9] Luis von Ahn1, Manuel Blum1, Nicholas J. Hopper1 , and John Langford CAPTCHA: Using Hard AI Problems For Security

[10] Bin B. Zhu*1, Jeff Yan2, Qiujie Li3, Chao Yang4, Jia Liu5, Ning Xu1, Meng Yi6, Kaiwei Cai7, Attacks and Design of Image Recognition CAPTCHAs

[11] Jeff Yan, Ahmad Salah El Ahmad Breaking Visual CAPTCHAs with Naïve Pattern Recognition Algorithms.