

Captcha Cracking Using Image Processing

Aditya Singh, Chirag Patil, Ramesh Malviya

Information Technology Department, Vishwatmak Om Gurudev College Of Engineering, Maharashtra, India

ABSTRACT

Today Internet is life line for most of individual person. The day start with looking for any updates and checking online status on social media and giving feedback in form of comment or likes. This feedback or likes can be controlled and manipulated by bots or automated programs which is provided by spammers. Here comes testing method Known as CAPTCHA (" an acronym for "completely automated public Turing test to tell computers and humans apart", trademarked by Carnegie Mellon University)[8]. It is type of test which is used to determine whether the user is Human or not. It is set of random alphabets and numbers which is twisted so that it cannot be recognized by bots but is readable by humans. There is two version of CAPTCHA . Version one which consist of twisted alphabets and numbers and Version two which consist of multiple images to select from according to system. Therefore it is very important to check vulnerability to check CAPTCHA against attack. So in this paper we will check the method to Crack CAPTCHA and check its effectiveness strength and security against spammers and automated programs.

Keywords: Captcha, Image Processig, Ocr, Turing Test, Matlab, Captcha Cracking, Security

I. INTRODUCTION

In last few years internet has changed drastically. No of internet user has increased and so are attacks and its exploitation inform of spammers and hackers. This hackers create automated programs or bots to generate traffic and load on servers which slow down websites and other portals. SO we are going to crack this CAPTCHA using image process and check strength and complexity of CAPTCHA.

1.1 LiteratureSurvey

Many organization have cracked CAPTCHA to check and make better CHAPTCHA and improve security. There are paper published by researchers on same one of them is „Stanford Researchers crack CAPTCHA code“ by Todd Wasserman published by ‘Stanford University has created DeCAPTCHA, It is a software which makes computer read CAPTCHA by clanging up text and rendering them in liable format of alphabets and numbers. It was not fully accurate it was able to crack

CAPTCHA but not fully its accuracy was 60 to 70 percent. Another paper titled as ‘Braking and image based CPATCHA by MICHAEL Merler, JACQUILENE Jacob it was published by Stanford University is on Vidoop CAPTCHA. It is another type of verification process that uses images of the objects, and animals, people, instead of distorted text to distinguish a human from computer program and bots .Authors underestimate that since a computer programs and bots can try to access several times a day without any change in effort.

II. WORKING

In this program we are going to crack CAPTCHA using IMAGE PROCESSING for image based CAPTCHA which are not highly secured. We are going to create a program which will extract the image from web page using developer mode or by taking the snapshot of web page. After that it will search for the image or part were CAPTCHA is. Then

it will process the image or rather say CAPTCHA file which is in .bmp format. It will extract the text using algorithms and give us the result and it will automatically submit the result text in required field and submit the form. This will tell us the level of security CAPTCHA of that web site has. Example of CAPTCHA from web page is shown in Figure1.



Figure 1. Exampe Of Captch On A Web Page

III. METHODOLOGY ANG PROBLEM APPROCH

3.1. System Architectur

System Architecture consist of following parts as proposed by our team.

1. Layout (User Inerface)
2. Data Extraction
3. Pre Processing And Processing
4. Decession Making
5. Result (Output)

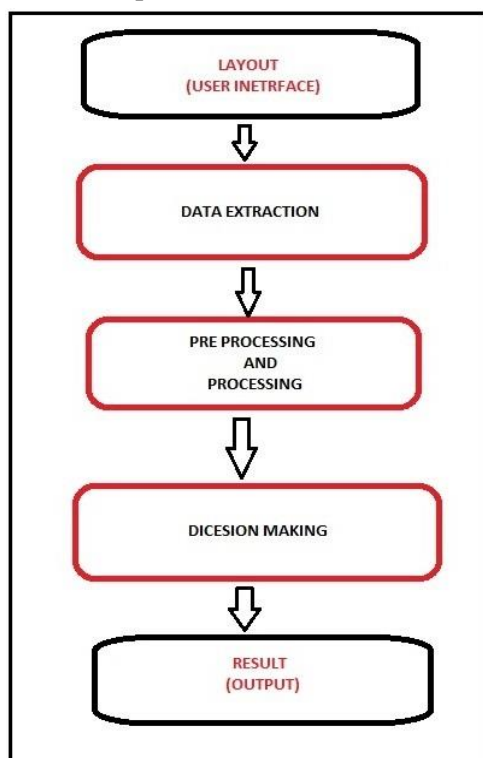


Figure 2. System Architecture And Flow Chart

3.2. Discription

3.2.1. Layout (User Interface)

User Interface it is physical appearance of software with GUI (Graphical User Inter face) at user end with controls and input output data.

3.2.2. Data Extraction

In this phase after getting command by user from layout or user interface the program will extract CHAPTCHA image from the web page for further process.

3.2.3. Pre Processing And Processing

After the image is extracted the image is pre processed because the original image is unfit for processing so is character in it as well. In this there are chances that the data or character in image is not of mono color so first preference is to make one. Then it is further processed which includes.

- a) Bifurcating according to RGB value
- b) rotation of image
- c) finding original corner of character
- d) resizing of image

3.2.4 Dicesion Making

In this phase after image being processed it can be recognized by using algorithms and Artificial intelligence to recognize character. Such OCR and MATLAB. After character benign recognized it will submit in required field and if it was success full it will enter the page and if not it will reload next CAPTCHA image from web page and process continues [8].

3.2.5 Result (Output)

After successful attempt we will have full access or logged in to web page then that means CHAPTCH is vulnerable.

IV. ALGORITHM

4.1 Data [Preparation

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: Exampe Of Captch Before Processing



Figure 4: Exampe Of Captch After Processing

4.2 Segmenttion And Pixeltion Segmentation

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 Figure 5.[2]

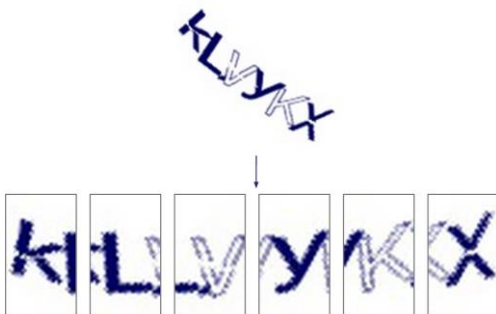


Figure 5. Exampe Of Captch Segmentation

Pixelation

The pixel-to-pixel comparison function requires both character images to be the same width/height. Simple experiments determined a 16 pixel by 16 pixel image was sufficient to retain important characteristics. It was important to decaled the image, because for each additional row/column k , it increased the time complexity. The actual pixilation was done by dividing the original character image into the 256 cells, and determining the average value of each cell (1=black, 0=white). If the average value of each cell was greater than 0.5, then the script set the corresponding pixel in the pixilated image to black, and vice versa.

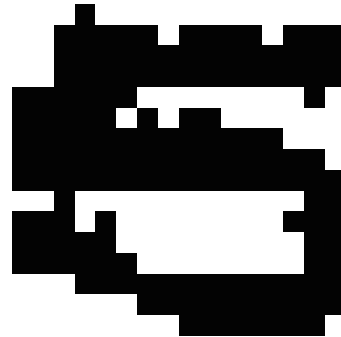


Figure 6. Exampe Of Character Pixelation

4.3 Classify

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

4.3.1 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%.

4.3.2 Vertical Projection

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].

4.3.3. Horizontal Projection

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%.

4.3.4. Template Correlation

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%.

4.4. Template Matching And Correlation Technique

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].

V. RESULT (OUTPUT)

After successful attempt we will have full access or logged in to web page then that means CHAPTCH is vulnerable.

VI. GOALS OF THIS APPLICATION

The main goals of our application are:

1. To show the vulnerability of CAPTCHA
2. Further improvement of CAPTCHA and secure online system.

3. Automate login for person who face problem with CAPTCHA.
4. Testing of web page with security issues.

VII. CONCLUSION

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.

VIII. FUTURE SCOPE

In future automated system can be made to increase security as well as check vulnerability of a web site. It will also help those who face problem in logging in portal due to CAPTCHA.

IX. 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, Qijie 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 Algorithm