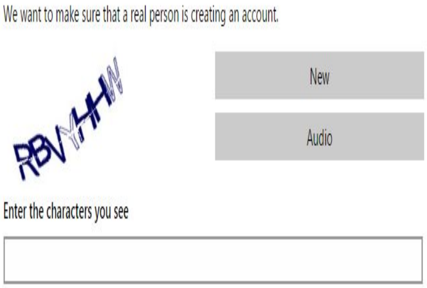
An Efficient and Improved Segmentation Algorithm for CAPTCHA Defeating and Solving

Manisha Jassal, Aditi Pundir

Abstract— **There are many websites that use CAPTCHA (Completely Automatic Public Turing Tests to Tell Computers and Humans Apart) schemes for human interaction proofs for accessing their services only to people rather than spam bots. For Defeating a captcha there is a requirement for two main procedures: segmentation and recognition. The recent research shows that the problem of segmentation is much complicated than the recognition. In this paper, improved segmentation algorithm is proposed. Experimental results show the proposed techniques can achieve segmentation rates from 9% to 15% over the traditional one.**

I. Introduction

There are various online services like social media, webmail and other platforms which are often abused by spam bots. Therefore websites are making use of CAPTCHAs (Completely Automated Public Turing Test to Tell Computers and Humans Apart) as one of their main defense mechanisms against such spam bots. To prevent such abuses, it is vital to design an automatic system. The main job or a purpose of the CAPTCHA is to separate computer programs form people automatically, using a computer based test. The typical CAPTCHA user interface consists of two parts: a character image with noise, and an input textbox then the user is asked to type the characters shown in the image. The challenges for CAPTCHA include recognition of distorted words, identification of the image, logical questions, understanding of speech and mathematical questions. We will be focusing on text-based CAPTCHAs because they were the first to get introduced and remain the most widely used type. Even though (OCR) optical character recognition has advanced a lot, but solving text-based CAPTCHA remains difficult. We have focused mainly on segmentation part of the algorithms and also used chellapilla's algorithm. The chellapilla algorithm uses the image opening and labeling technique for further designing of the segmentation algorithm. Example for a CAPTCHA is shown in the Fig.1; any human would be expected to type this answer as 'RBVYHHW' very easily. Therefore, if a user can correctly respond to this kind of a test, then the system defended by the CAPTCHA will be considering the user to be real human, otherwise the user may be considered as an illegal program, and may be denied access to the service.



We want to make sure that a real person is creating an account.

Enter the characters you see

Fig.1: An example of the MSN CAPTCHA system

While considering the various design principles of well-known captcha systems we see that many websites such a MSN, Yahoo, Google, YouTube, RapidShare are employing user interfaces similar to Fig.1. YouTube uses colored blocks to clutter the image whereas MSN and Yahoo do not use coloured character a noise. Instead, they use straight and curved lines and warped characters as image noise in their CAPTCHA system as the main techniques for confusing segmentation algorithms. This paper proposes an efficient segmentation algorithm for attacking CAPTCHAs. As we all know, that CAPTCHA system have a wide variety, so it is hard to attack all CAPTCHA tests by this algorithm. Therefore, we assume that CAPTCHAs being attacked have some of the following characteristics like: a single coloured displayed picture, using warped characters and straight and curved lines as image noisy clutter to confuse the attacking program or algorithm. Sequential steps that are involved: Pre-processing• Character segmentation• Character recognition• Thus, the main aim of this paper is to establish better segmentation and classification techniques for cracking CAPTCHAs. The remaining paper is organized as follows: Section 2 illustrates Analysis of the CAPTCHAs. Section 3 defines the Chellapilla's Algorithm. Section 4 presents segmentation algorithm scheme. Section 5 covers the results (experimental) on the basis of the algorithms and Section 6 provides the conclusion.

II. ANALYSIS OF THE CAPTCHAS

2.1 Background Clutter:

Captcha breaking programs contains pre-processing, segmentation and classification. But as we know that noisy lines which are labeled as clutters are widely used in the form of straight lines, curved lines and warped characters. If the background clutter consists of shapes similar to letter shapes, and the letters are connected by this clutter, then the segmentation becomes nearly impossible. This analysis suggests that there are eight different types of clutter with properties like color, intersection, size, curvature, length, angle and position. The properties are shown in Fig.2 below

Manuscript received March 26, 2016

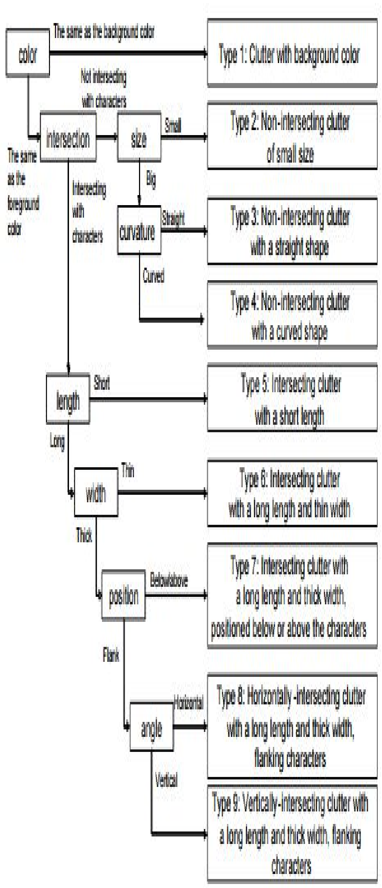
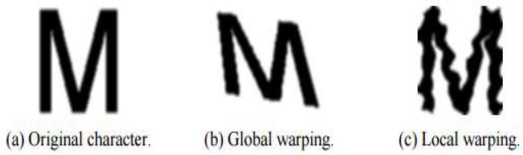

color
The same as the background color
Type 1: Clutter with background color
Not intersecting with characters
intersection
size
Small
Type 2: Non-intersecting clutter of small size
The same as the foreground color
Intersecting with characters
Big
curvature
Straight
Type 3: Non-intersecting clutter with a straight shape
Curved
Type 4: Non-intersecting clutter with a curved shape
length
Short
Type 5: Intersecting clutter with a short length
Long
width
Thin
Type 6: Intersecting clutter with a long length and thin width
Thick
position
Below/above
Type 7: Intersecting clutter with a long length and thick width, positioned below or above the characters
Flank
angle
Horizontal
Type 8: Horizontally-intersecting clutter with a long length and thick width, flanking characters
Vertical
Type 9: Vertically-intersecting clutter with a long length and thick width, flanking characters

Fig.2: Categorization of different types of clutters in CAPTCHA system.

2.2 Warped characters:

There are two types of character warping which are commonly used in CAPTCHAs which are: global warping and local warping. In global warping the whole character is distorted while in local warping, it has many irregular ripples and waves that prevent character recognition. The example of character warping is shown in Fig.3 below.



(a) Original character. (b) Global warping. (c) Local warping.

Fig.3: Examples of character warping

III. CHELLAPILLA'S ALGORITHM

Chellapilla shows that cracking the CAPTCHA is basically a twofold process: 1. First segmenting the CAPTCHA into individual characters, which is harder. 2. Second is classification of the segmented characters for recognition. This algorithm essentially includes three phases which are pre-processing, image opening and labeling. The pre-processing includes thresholding and up-sampling. In pre-processing step the CAPTCHA image is "up sampled". Up sampling enlarges the image, increases its pixel details and therefore smoothens the embedded text, and then binarizing process is done via standard thresholding process.

In this thresholding method all the color values above a predetermined threshold is converted to black and those below to white. Thresholding is performed because the background colors are either darker or lighter than the text characters. Thresholding is the basic act of partitioning pixels into two groups, text and background. It is used to help segment characters and find character pixels. Image opening is the process for allowing segmentation of the characters from the CAPTCHA images. In this process, the already pre-processed image will go through an erosion process several times and then later it will be dilated. The erosion will rub off the character borders one pixel per time whereas the dilation process will mend the character border one pixel per time, which further results in deleting some of the cluttered items from the background. The labeling phase or labeling process finds all the connected components in the image, and considers the larger and clearer ones as characters. The block diagram for chellapilla's algorithm is given below in Fig.4 and some of its problems in Fig.5

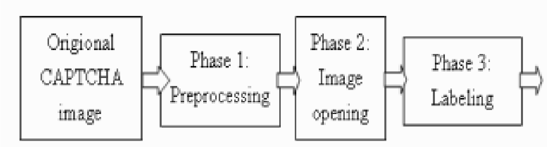
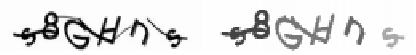


Fig 4: Structure for chellaphilla's algorithm

This algorithm is useful when the clutter is of thinner width than the characters. But, algorithm produces errors because it can recognize the difference between characters and clutters of similar width and it may not delete or erase some clutter.



(a)Original image (b) some of clutter cannot be erased

Fig 5: Problems in chellapilla's algorithm

3.1 Chellapilla's Algorithm (OUTLINE):

STEPS INVOLVED ARE

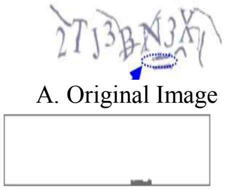
Step1: Identify the boundaries of the words. Step2: Select the way word from the image and discard it. Step3: Check the inclination of the letters. Step4: Up-Sample the image. Step5: Dilate it to reduce the thickness of the character lines. Step6: Remove the noise around characters using 'blur' process. Step7: Threshold to get a binarized image. Step8: The pattern matching technique is applied. Step9: The letter is identified and the line is drawn removing the rest of the character. (a) Find the current starting line. (b) Calculate the next segmentation line (c) Remove all the pixels between the start line and the dividing line and save them for the recognition of the character Step10: If somehow, the segmentation for next point could not be calculated, then fail. Step11: The letters are recognized.

IV. PROPOSED SEGMENTATION ALGORITHM SCHEME

(Segment text from the background and attempt to split it into single character blocks.) The goal of the segmentation is to identify the location of the characters and extract them from the rest of the image. This section includes two main parts: (a) Projection (b) Proposed Segmentation Algorithm These both are basically for improving the success rate of segmentation, and further gives us effective and better results.

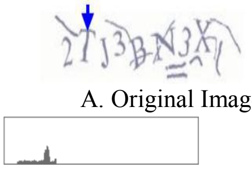
4.1 PROJECTION

The projection method or technique used in this paper is mainly on the idea of projecting the image data onto the X-axis and this further is implemented by summing the number of pixels in each column which are present on the Y-axis of the image. The projection in the X-axis will tend to appear large and unstable, when a component represents a character rather than an item of clutter. Thus, by computing a component's projection value and its variance, it is definitely possible to distinguish between components that are clutter and components that are characters.


A. Original Image

B. The projection in the X-axis for the highlighted component

Fig.6: Example-1 for a character and its projection


A. Original Image

B. The projection in the X-axis for the highlighted component.

Fig.7: Example-2 for a character and its projection.

4.2 PROPOSED SEGMENTATION ALGORITHM

Now, this is the segmentation algorithm for the CAPTCHAs with character warping and also line cluttering. This algorithm is basically based on the chellapilla's algorithm and has main five phases: Pre-processing, Image Opening, Labeling, Component splitting and character extracting. It has a similar process for the first three phases as of chellapilla's.

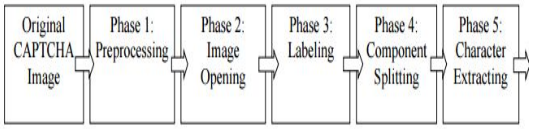


Fig.8: Block diagram for the CAPTCHA segmentation algorithm

Therefore, the fourth phase (component splitting phase) includes projection separation techniques. The components that have not fully separated can by separated by this as image opening phase will have already erased the thin clutters. After all these processes and phases, the original image will be separated into various different connected components. Take an example of a MSN CAPTCHA in the Fig.8 below with 8 characters but the image is broken into discrete connected components and in the final phase (character extracting phase), it outputs the location of the characters.

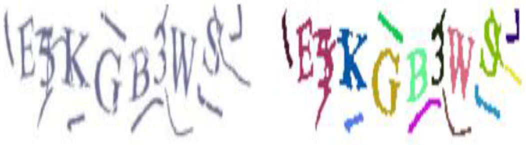

(a)Original Image (b) Image after fourth phase

Fig.8: Example of the MSN CAPTCHA showing different connected component

Below is the example of the proposed segmentation algorithm in Fig.9.

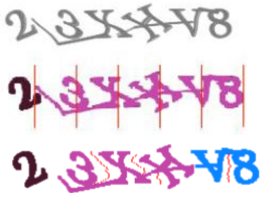
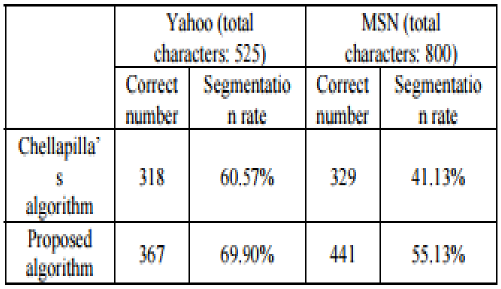


Fig.9: Successful segmentation produced by the proposed algorithm.

V. EXPERIMENTAL RESULTS

In this section, experimental results are shown for the chellapilla's algorithm and the proposed algorithm for segmentation. These algorithms are applied to the yahoo and MSN CAPTCHA systems and the segmentation rate is based on the numbers of characters in different images. Suppose take an example of MSN, every image in this system consists of 8 characters. If the algorithm can segment up to 40 characters from 20 images then the segmentation rate would be 40/ (20*8) =0.25 or 25%. The segmentation rate of the proposed algorithm attacking the yahoo CAPTCHA system was higher than the chellapilla's algorithm by 9% and for MSN by 14%.

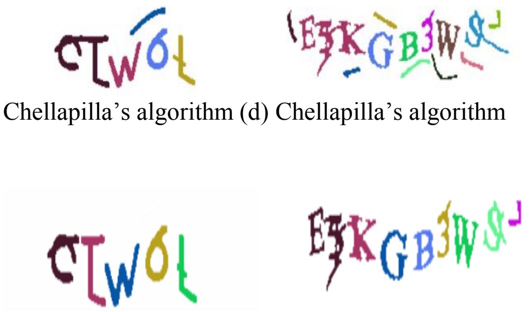
Table 1: Experimental Results



	Yahoo (total characters: 525)		MSN (total characters: 800)	
	Correct number	Segmentation rate	Correct number	Segmentation rate
Chellapilla's algorithm	318	60.57%	329	41.13%
Proposed algorithm	367	69.90%	441	55.13%

The below Fig.10 show the different results for two different types of CAPTCHA.

Original Yahoo Image (b) Original MSN image



Chellapilla's algorithm (d) Chellapilla's algorithm

(e) Projection-only result (f) Projection-only result

Fig.10: Results of Algorithms for two different CAPTCHA systems.

CONCLUSIONS AND OUR FUTURE WORK

In this paper, an improved and effective algorithm was proposed for the segmentation of the CAPTCHAs (mainly for MSN and yahoo, as shown in the examples) containing line cluttering and character warping, proven to be effective against these CAPTCHAs. Based on the results of our implementation and evaluation and also the experimental results, it was found that the proposed algorithm can uniformly improve the segmentation rate over the traditional algorithm and this, further makes useful and effective contributions in the field of CAPTCHA analysis. Our future work will basically focus on increasing the degree of successful implementation of segmentation for CAPTCHAs and on testing and developing various new techniques in this area of research. We believe in the possibility of extending and applying new techniques and methods.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

REFERENCES

[1] M. Blum, L. A. von Ahn, and J. Langford, The CAPTCHA Project, “Completely Automatic Public Turing Test to tell Computers and Humans Apart," www.captcha.net, Dept. of Computer Science, Carnegie-Mellon Univ., and personal communications, November, 2000.

[2] Chellapilla K, Larson K, Simard PY, Czerwinski M. “Building segmentation based human friendly human interactive proofs”, Proceedings of the Second International Workshop on Human Interactive Proofs Springer-Verlag 2005; pp.: 1-26.

[3] G. Moy, N. Jones, C. Harkless, and R. Potter, "Distortion Estimation Techniques in Solving Visual CAPTCHAs," in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 23- 28, 2004

[4] Christos Makris, Christopher Town “Character segmentation for automatic CAPTCHA solving”, University of Cambridge Computer Laboratory, 15 JJ Thomson Avenue, Cambridge CB3 0FD, UK.

[5] S.Y. Huang, Y.K. Lee, G. Bell, and Z.H. Ou, “A Projection-based Segmentation Algorithm for Breaking MSN and YAHOO CAPTCHAs", in Proceedings of the 2008 International Conference of Signal and Image Engineering (ICSIE'08), London, UK. (2008).

[6] Pope and K. Kaur, "Is It Human or Computer? Defending E-Commerce with CAPTCHAs," IT Professional, vol. 7, no. 2, pp. 43–49, 2005. [7] G. Mori and J. Malik, “Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA,” in Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 134-141, 2003. [8] Ankitkr (pdf) projects on CAPTCHA “home.iitk.ac.in “.