Journal of Computer and Communications, **2014, 2, 14-19**
Published Online January 2014 (http://www.scirp.org/journal/jcc)
http://dx.doi.org/10.4236/jcc.2014.22003

The Recognition of CAPTCHA

Min Wang, Tianhui Zhang, Wenrong Jiang, Hao Song

Mathematics School, Jilin University, Jilin, China.
Email: whateverhg@163.com

Received October 2013

ABSTRACT

CAPTCHA is a completely automated program designed to distinguish whether the user is a computer or human. As the problems of Internet security are worsening, it is of great significance to do research on CAPTCHA. This article starts from the recognition of CAPTCHAs, then analyses the weaknesses in its design and gives corresponding recognition proposals according to various weaknesses, finally offers suggestions related to the improvement of CAPTCHAs. Firstly, this article briefly introduces the basic steps during the decoding process and their principles. And during each step we choose methods which are better adapted to the features of different CAPTCHA images. Methods chosen are as followings: bimodal method in binarization, improved corrosion algorithm in denoising, projection segmentation method in denoised image processing and SVM in recognition. Then, we demonstrate detailed process through the samples taken from the online registration system of ICBC, show the recognition effect and correct the results according to the statistical data in the process. This article decodes CAPTCHAS from three other large banks in the same way but just provides the recognition results. Finally, this article offers targeted suggestions to the four banks based on the recognition effect and analysis process stated above.

KEYWORDS

Recognition of CAPTCHAS; Bimodal Method; Corrosion Algorithm; Projection Segmentation Method; SVM; Recommendations for Improvement

1. Introduction

With the rapid development of Internet, our daily life heavily relies on the existence of the Internet.

As a result information security has become a serious issue. The CAPTCHA emerges in this case. Carnegie Mellon University and Palo Alto Research Center are doing related research.

CAPTCHA is an open question based on artificial intelligence. It challenges the researchers who are devoted in AI which not only includes security researchers but also hackers, so the CAPTCHA is a win-win situation. If a CAPTCHA cannot be decoded then there is no method to tell humans and computers apart; if a CAPTCHA is decoded then an AI problem is solved.

The CAPTCHA works in this way: the server-side generates random characters and stores them in memory (usually a web session object in the system), then writes the characters into an image and sends the image to the browser-side to display. The browser-side inputs the characters in the image and submits to the server-side. If the characters submitted by the browser-side accord with the characters saved in the session object the browser-side can continue else the server-side returns an error massage.

2. Principles and Programs

2.1. Basic Steps in Decoding CAPCHAs

- Image acquisition
- Binarization
- Denoising
- Segmentation
- Recognition

2.2. Principles in Each Step

2.2.1. Principles in Binarization

First we need to set a threshold T according to the grayscale of the image and turn the greyscale image into binaries image through a function defined as Equation (1).

$$\begin{cases} 1 & \text{grey}(x,y) \geq T \\ 2 & \text{grey}(x,y) < T \end{cases} \quad (1)$$

(In the formula 0 represents black point and 1 represents white point).

Considering that the CAPTCHA image is unique, it is better to use bimodal method to set the threshold. The bimodal method thinks that the image consists of foreground and background. In the grayscale histogram, foreground and background both form a peak and in the trough between the two peaks is where the threshold lies (see **Figures 1** and **2**).

2.2.2. Principles in Denoising

In order to increase the difficulty in segmentation and recognition many CAPTCHAs contain noise, so before segmentation and recognition we need to denoise.

The most common noises include spot noise, thin noise, thick noise and block noise. CAPTCHAs of the online banks often contain spot noise and thin noise which can be removed by corrosion algorithm. Original corrosion algorithm will cause a significant loss of character information and inconvenience for the recognition followed because the CAPTCHA image contains very low pixels [1].

In this regard we make corresponding improvements: scan the four adjacent points of the target point, if the up point and down point (or the left point and right point) are both white point then we decide that this point is a noise point and remove it. And we can scan the target area for many times without causing damages to target area (see **Figure 3**) [2].

2.2.3. Principles in Segmentation

In order to handle the characters in the image respectively we need to segment the treated image. For images of small inclination and low adhesion, projection segmentation is better [3].

Scan the image by column in units of pixel and cumulate the number of pixels whose pixel value is zero to form the histogram of this column (see **Figure 4**). Generally we consider that the area between a peak and a trough is a separate character. We can set a segmentation threshold and define the column whose statistical value is below the threshold as the dividing line. After having decided the left and right boundary of the character, we can conduct a horizontal projection to limit the character into a smaller area (see **Figure 5**).

2.2.4. Principles in Recognition Based of SVM

SVM, namely support vector machine, is a machine learning method targeting at constructing an objective function to separate two modes as far as possible based on risk minimization principle [4].

For multi-classification problems, we can establish classifiers between every two classes.

The SVM algorithm classifies and extracts models through training of small samples. Furthermore, it uses the model extracted and the input data to predict and eventually find the category, namely recognition [5].

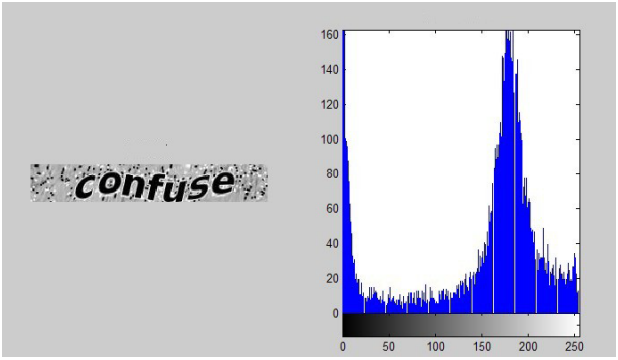

Figure 1. Greyscale image and grey level histogram of CAPCHA.

Figure 2. Image after binaryzation–T = 50.

Figure 3. Image after improved denoising.

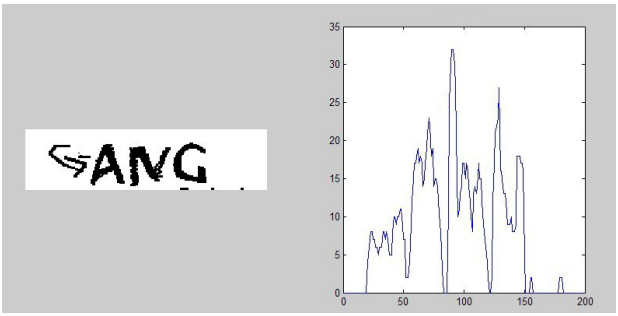

Figure 4. Vertical projection of image.

Figure 5. Image after vertical projection seg- mentation method.

2.3. Implementation Process (see Figure 6)

2.3.1. Example of ICBC

1) Features of CAPTCHAs from ICBC

- No thin noise
- Dark background (no need to gray)
- Consistent font
- Tilted characters
- No adhesions

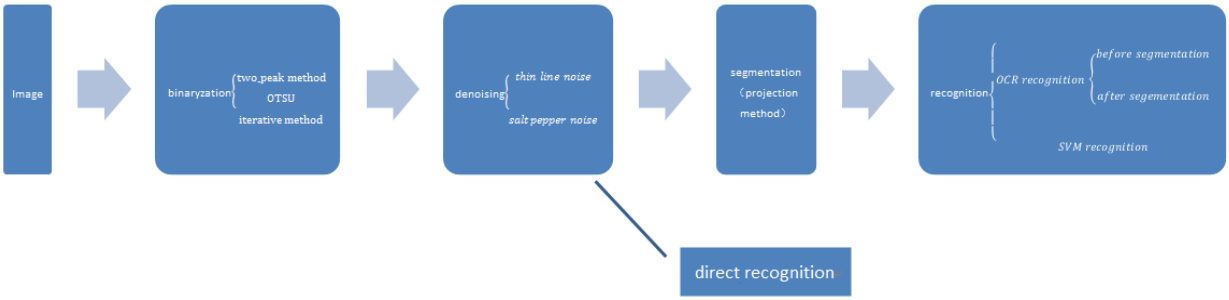

Figure 6. Implementation process[1].

- Fixed position of CAPTCHAS
- Characters' position closer to the top of the image
- 4 characters in each CAPTCHA image
- Frequency statistics of each characters (100 samples) (see **Tables 1** and **2**)

2) Recognition Process of ICBC CAPCHA[2]

- Extract CAPCHA image (png)
- Transform the storage format (png → jpg) (see **Figure 7**)
- Binarization(bimodal method) (see **Figure 8**)

Judgment: further denoising is required.

Annotation: Based on the features of ICBC, if we choose OTSU method or iterative method, we do not need to denoise the image, the effects are shown by **Figure 9.**

- **Denosing[3]**

a) The process of denoising can be divided into two parts: removing image boundary and removing noise points.

b) Judgment criteria of spot noise: if the 9 pixels surrounding the current point except the current point itself are all white, we call it a noise point (definition of black dot: grey level < 30).

c) Triplicate the denoising step can lead to a relative good result see **Figure 10**.

- **Segmentation**

a) Use projection method

b) Definition of black dot: dots with grey level fewer than 8(from experiment)

c) Threshold k (the number of black dots in each column) set as 1(see **Figure 11**) (Setting form experiment, if k is too large, it will cause character being truncated)

d) The left boundary conditions: the current column sum is equal to or less than threshold k, and the next column sum is greater than k (from experiment)

e) The right boundary conditions: the previous column

[1]Here the binarization, denoising and recognition are all conducted through different methods showed in the brackets and their effects are compared. In addition, if not necessary, the process of denoising can be omitted.

[2]Here takes bimodal method in binarization process as an example.

[3]Only appears as examples. Actually we choose Otsu but not bimodal method because Otsu can omit the denoising process.

Table 1. Frequency of all letters.

Letter	A	B	C	D	E	F	G
Frequency (%)	15	17	16	20	13	25	0
Letter	H	I	J	K	L	M	N
Frequency (%)	13	20	20	13	0	0	28
Letter	O	P	Q	R	S	T	U
frequency	0	22	0	19	0	22	16
Letter	V	W	X	Y	Z		
Frequency (%)	19	0	20	14	0		

Table 2. Frequency of all numbers.

number	0	1	2	3	4	5	6	7	8	9
Frequency (%)	0	0	0	12	12	0	21	16	11	0

Figure 7. Original picture of CAPCHA.

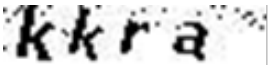

Figure 8. Image after binaryzation.

Figure 9. Up: image using OTSU denosing method Down: image after iterative denosing method.

Figure 10. Image after denosin and before segmentation A.

Figure 11. Threshold value k = 1, image after segmentation A-succeed.

sum is equal to or greater than threshold k, and the next column sum is less than threshold k [6] (from experiment).

f) The current threshold k is quite strict, may cause a small portion of characters cannot be separated (see **Figure 12** and **Figure 13**).Thus we segmented the figures for the second time. Examine the result of last step; if the length of separated figure is preternormal, we directly segment the figure from the middle, the rest pictures move backwards in order (see **Figure 14**)

g) The rate of failure on segmentation (split one character into two (see **Figure 15** and **Figure 16**)) is lower than < 5%

h) Failing characters (of all 400 characters) (see **Figure 17-20**), 'u', 'n' and 'p'

- **Recognition (see Table 3)**

a) OCR recognition

b) Effective analysis of OCR recognition (see **Table 4**)

c) Amendment based on effective analysis of OCR recognition

3) Simple Amendment

According to the statistical data of ICBC, it is easy to see that these CAPCHAs do not include capital letters or punctuation, thus we can deal with the pictures with rudimentary improvement: transform the capital letters in the result into lowercase letters.

However, we also noticed that this transformation is only effective when it happened to letters whose uppercase and lowercase is similar to a certain extend such as y→Y, c→C, v→V, i→I, j→J. In other situations, this process will cause errors.

4) Careful Amendment

Step 1: Remove single and double quotes;

Step 2: Do the following conversions: ~~L~~x , A→x, Q→a;

Step 3: Revise the result according to the transformation table (**Table 5**)[4];

Step 4: Repeat the simple amendment.

After the amendment above, we raised the recognition correctness rate to 55% (**Table 6**). Tough the correctness beyond the current samples would be lower than 55%, we do not create errors through whole amendment procedure and thus increase get a better result than just use simple amendment.

a) SVM recognition

We got 400 segmented characters figures after processing 100 sample ICBC CAPCHAs. But after we abandoned the failing characters, there were only 368 characters that are available for SVM training. Then we use uniformed the size of all the 368 pictures in order to obtain unified image pixels matrix.

[4]Transformation table is the result of statistical analysis on current 100 samples, and can only amend result in the following situations: A. transform one character to two characters B. contain capital letters C. appear letters that is not used in the website. Errors beyond situations above cannot be corrected. For example: recognize letter "I" as letter "j".

Figure 12. Before segmentation B.

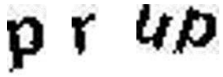

Figure 13. Threshold value k = 1, image after segmentation B–fail.

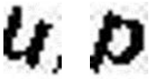

Figure 14. Image after twice segmentation.

Figure 15. Before segmentation C Figure.

Figure 16. Failure of segmentation C.

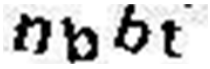

Figure 17. Before segmentation D–letter n.

Figure 18. Failure on segmentation of letter n.

Figure 19. Before segmentation E–letter u.

Figure 20. Image after twice segmentation: Failure on segmentation of letter u.

Table 3. Recognition rate of OCR.

Recognition rate before segmentation	55%
Recognition rate after segmentation	14%

Statistical study showed that these characters can be divided into 23 classes, *i.e.* j, x, p, u, 6, t, n, a, I, r, y, d, h, 4, e, b, c, f, 3, v, 7, 8.Class label of each class is marked as number 1to 23. Using Libsvm 3.17 in Matlab, we train the 368 characters and generate the model, then use function svmpredict to predict the input testing characters then get the predict label, namely, the result of classification.

Table 4. Table of recognition errors[5].

Character	c	U	p
Recognition Result	e*, t, (, C,	1t, tr, o*, ll, LI	d*, 0, f, l7, I7
Character	f	H	j
Recognition Result	t*, r,	r,/, l7, tj, tl,/, lj	I*, J
Character	8	v	a
Recognition Result	B	/t, /l*, V	Q
Character	b	x	4
Recognition Result	h*, 0, /D*, tD	I, k, A*, y*, v,/, y, l(, Y	S, q, < 1 7
Character	k	t	y
Recognition Result	/e, i(, t(*, /v, &	r,T	v*, Y, j
Character	n	7	i
Recognition Result	rr, ti, r7*, f7, t7*, o, tl*, fl, rt	1*, I, ?, lr	1, j, J
Character	3	6	e
Recognition Result	?	b*, d	(!
Character	d	up	r
Recognition Result	a*, ct, cf*, (t*, c, (l, (;f, (/	/1/O	v

[5]errors with mark* occur more than once.

Table 5. Transformation table (OCR recognition result→ revised result A/revised result B).

ll→u/h	2L→a	tJ→h	r7→n
fl→n	i(→k	I→j	rt→n
l(→x	?→3/7	A→x	<l→4
lr→7	/→x	s→4	1→7/i
Ll→u	lo→p	tl→h/n 7	t7→n/d
LI→u	//→h	0→p/b	f7→n
Lr→u	lj→h	o→p/b/u/n	ri→n
l7→h/p/a/n	tj→h	i7→p	r)→n
t(→k	q→4	(→c/b	(t→d
(l→d	/<→k	Q→a	&→b/k
ci→d	(/→d	(!→e	es→8
c/→d	(I→d	/t→v	Qf→d
ct→d	(f→d	/l→v	tD→b
L→x	cf→d	(i→d	

Table 6. Recognition result after careful a- mendmen.

Recognition rate before amendment	Recognition rate after simple amendment	Recognition rate after careful amendment
14%	22%	55%

We then select 30 CAPCHAs randomly for prediction with above-mentioned method. The result shows in **Table 7**.

2.3.2. Results of the Other Three Websites

We use similar method to deal with the other three websites, and the result is as **Table 8-10**.

3. Suggestion

In fact, not only the four websites but also other major banks in China have serious problems on CAPCHA and with some simple tools we can easily recognize quite a number of them. Furthermore, if we also combine more advanced tools such as machine learning or clustering analysis with current method, we can even reach a better result.

To help the four banks becoming safer, we conclude the experiment above and come to several suggestions that are in common use among many websites based on the recognition process of the 4 banks:

1) Vary the position of CAPCHA can make it more difficult for attackers to extract it [7].

2) Vary the fond of characters will decrease the effectiveness of machine learning.

3) Unfixed number of characters in each CAPCHA will also increase the difficulty in locating the characters and segmentation.

4) Complicate the background by adding interferential characters can confuse the automatic recognition.

5) Use multilayer noise, especially line noise.

6) Curve noise may have better effect on counterattack.

7) Use different types of CAPCHA in one website will

Table 7. SVM recognition rate.

Recognition rate of single character	Recognition rate of whole image
67.5%(81/120)	20%(6/30)

Table 8. Bank of communications.

OCR recognition rate before segmentation	OCR recognition rate after segmentation	SVM recognition rate
8%	11%	18%

Table 9. China construction bank.

OCR recognition rate before segmentation	OCR recognitio rate after segmentation	SVM recognition rate
20%	10%	21%

Table 10. Bank of China.

OCR recognition rate before segmentation	OCR recognition rate after segmentation	SVM recognition rate
38%	37%	24%

increase the safety factor of website

8) Add more overlap, adhesion or projection overlap to make it difficult to segment correctly.

9) Increase the frequency of certain letter which is more likely to be segmented falsely. (These letters can be concluded through sample recognition, and vary from website to website).

10) Use other small characters as noise, and only ask the user to input the large letters.

11) Increase the frequency of certain letter which is more likely to be recognized falsely, especially those who can hardly be recognized through correction of errors. Take ICBC as an example, these letters are: x-y, x-v, f-t, i-j, y-v, b-h, a-d. Besides, these letters have certain similarity among different websites.

Of course, deeper exploration is needed for practicing these suggestions on CAPCHA design while human could still be able to recognize it.

4. Conclusion

Automatic recognition rate of effective CAPCHA should be lower than 1%. Considering we already got 20% or more of it through our recognition procedure, it is apparent that the safety factors of these websites are very low.

Apart from that, we also show the detailed procedure of practicing the method as well as the solution of problems which might be encountered through whole procedure.

Finally, we offer some suggestions to the 4 major websites respectively so as to help them realize the danger of current CAPCHA and make targeted changes to the CAPCHA and then form a safer online trade environment.

REFERENCES

[1] G. Yin, "The Recognition of CAPCHAs Based on SVM," 2010

[2] R.-E. Fan, P.-H. Chen and C.-J. Lin, "Working Set Selection Using Second Order Information for Training SVM," *Journal of Machine Leaning Research*, Vol. 6, 2005, pp. 1889-1918.

[3] N. Qu and L. Li, "Filtering Algorithm of Salt and Pepper Noise based on Fuzzy Theory," *Computer Knowledge and Technology*, Vol. 5, No. 10, 2009, pp. 2699-2700.

[4] X. Y. Wen, N. Gao, P. N. Xia and J. S. Jin, "Assorting Thoughts & Rcognition Technique of CAPCHAs," Computer Engineering.

[5] P. Lu, "Research and Application on SVM," Hunan University, 2007.

[6] L. Y. Wang, "Extraction and Classification of Image Features," Xi'an Electronic and Engineering University, 2006.

[7] L. Wang, R. Zhang, D. Yin, J. C. Zhan and C. Y. Wu, "Recognition of Touching Characters on CAPCHAs," *Journal of Computer Engineering and Applications*, 2011.

[8] R. L. Duan, Q. X. Li and Y. H. Li, "Summary for Methods for the Detection of Image's Edge," *Optical Technology*, 2005.

[9] M. Yang, L. B. Zeng and D. C. Wang, "A Fast Algorithm for Mathematical Morphology and Corrosion & Expansion Operation," *Computer Engineering and Applications*, 2005.

[10] Q. L. Han, M. Zhu and Z. J. Yao, "Extraction of Image's Characteristic Segment Based on Hough Transform," *Instrumentation Science*, 2004.

[11] H. Ji, J. X. Sun, X. F. Shao and L. Mao, "Outlook of Methods for Image Edge Extraction," C*omputer Engineering and Applications*, 2004.