Multimed Tools Appl
DOI 10.1007/s11042-016-4151-2

Segmentation of connected characters in text-based CAPTCHAs for intelligent character recognition

Rafaqat Hussain[1] · **Hui Gao**[1] · **Riaz Ahmed Shaikh**[2]

Received: 31 July 2016 / Revised: 8 October 2016 / Accepted: 11 November 2016

Abstract Over last few years, CAPTCHAs are ubiquitously found on internet as a security mechanism to distinguish between humans and spams. The text-based CAPTCHAs offer users to recognize the distorted text from the challenged images. Having based on hard AI problem, they have emerged as a hot research topic in computer vision and machine learning. The contemporary text-based CAPTCHAs are based on the segmentation problem that involves their decomposition into sub-images of individual characters. This is a challenging task for current OCR programs which is not yet solved to a great extent. In this paper, we present a novel segmentation and recognition method which uses simple image processing techniques including thresholding, thinning and pixel count methods along with an artificial neural network for text-based CAPTCHAs. We attack the popular CCT (Crowded Characters Together) based CAPTCHAs and compare our results with other schemes. As overall, our system achieves an overall precision of 51.3, 27.1 and 53.2% for Taobao, MSN and eBay datasets with 1000,500 and 1000 CAPTCHAs respectively. The benefits of this research are twofold: by recognizing text-based CAPTCHAs, we not only explore the weaknesses in the current design but also find a way to segment and recognize the connected characters from images. The proposed algorithm can be used in digitization of ancient books, handwriting recognition and other similar tasks.

Keywords CAPTCHAs · Artificial Intelligence · Machine learning · Image processing · Crowding characters together · Intelligent character recognition

✉ Rafaqat Hussain
rafaqat.arain@salu.edu.pk

[1] School of Computer Science and Engineering, University of Electronics Science and Technology of China, Chengdu 611731, China

[2] Department of Computer Science, Shah Abdul Latif University, Khairpur 66020, Pakistan

Published online: 18 November 2016

1 Introduction

CATPCHA (Completely Automated Public Turing to tell Computers and Humans Apart) is the standard security mechanism to stop the spam on the internet. It is used to distinguish between humans and bots. If there were no CAPTCHAs or if automated computer programs (bots) could solve the CAPTCHAs then they can cause severe damages, for example they can send thousands of junk emails, signup for free email accounts, post thousands of comments, and vote multiple times in an online poll in no time. Therefore CAPTCHAs are imperative. A good CAPTHCA is assumed to be resistant against automated attacks. CAPTCHAs are being attacked since their introduction; they have been attacked by computer vision researchers and spammers. It is actually an open and friendly war between the CAPTCHA designers and attackers. One step backwards in the CAPTCHA design is actually one step forward in the field of AI (Artificial Intelligence). According to a very important statement of one of the pioneers of this field, Luis Von Ahn where he himself stated that they want their CAPTCHAs to be broken, because they assumed that if their CAPTCHAs would be broken then it will lead to the development in the field of AI, and otherwise a security mechanism would be created to stop the spam [1]. This statement has motivated us to work in this field to examine the security of contemporary text based CAPTCHAs. This has not only helped us to solve a hard AI problem of recognition of connected and distorted characters but also made it possible for us to find the vulnerabilities in the current design.

Various design alternatives of CAPTHCAs were introduced over the years and almost all are being attacked by researchers or spammers. Despite of various other alternatives, Text-based CAPTCHA is still the most popular and widely used security mechanism due to its easy implementations. Figure 1 depicts various design variants along with examples of our attacked CCT-based CAPTHCAs.

Text-based CAPTCHAs request the users to recognize the distorted text in the images in order to prove them as humans. Image-based CAPTCHAs offer users to recognize the

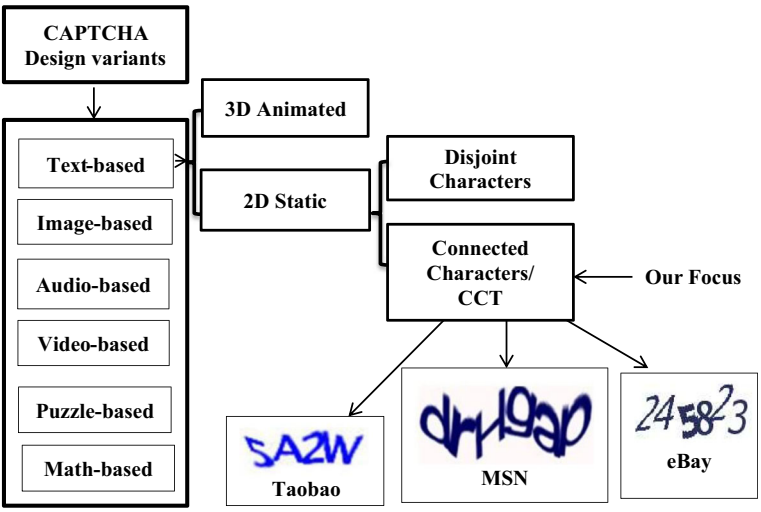


Fig. 1 Various design alternatives of CAPTCHAs along with examples of our attacked CAPTCHAs

objects in an image. Audio CAPTCHAs require the users to solve the speech recognition task. Puzzle-based CAPTCHAs require the solution of a puzzle. There are also some other types of CAPTCHAs such as Video CAPTCHAs which asks the users to recognize the text in a video and Math CAPTCHAs which require the users to solve the mathematical equations.

The task of recognizing the connected characters is supposed to be a trivial task for humans but extremely difficult for current OCR (Optical Character Recognition) programs. The research has proved that computers have outperformed humans in terms of recognition of individual characters [5]. Unlike early CAPTCHAs which were based on recognition of distorted characters (recognition resistant), current CAPTCHAs are based on the problem of segmentation of connected characters (segmentation resistant). The segmentation is the only step where human outperforms machines. Instead of recognizing 'what the character is'; current research is focused on 'where the character is' i.e. locating the characters [14]. An example of such type of CAPTCHA is shown in Fig. 2. Segmentation of such CAPTCHAs, i.e. splitting individual characters is a challenging task.

In this research we have used simple image processing techniques like thresholding, thinning and pixel count methods along with a multilayer feed forward neural network with back propagation. The characters in a CAPTCHA image are partially segmented on the basis of open and closed characters. This first step results in segmentation of closed characters. As the closed characters contain at least two pixels in each column therefore these characters are properly segmented in this step. On the other hand the open characters are over segmented due to multiple single character columns within a character. This over segmentation of open characters is reduced by calculating minimum horizontal distance between consecutive segments. The remaining incorrect segments are discarded by using a trained neural network. The closed single characters and simple open characters are segmented using the said method.

However the characters having multiple ligatures with adjacent characters are not segmented. These complex ligatures between adjacent characters results in large chunks. These non-segmented large chunks containing two or more than two characters are segmented by using recognition based segmentation method with the use of neural confidence. The neural classifier (trained to recognize the individual characters) is utilized to recognize the characters using a sliding window in these large chunks. The character recognized with highest confidence is segmented from the image. It is worth mentioning here that our proposed method works on a 'Divide and conquer' rule, i.e. each step results in a reduced number of characters in the CAPTCHA image which simplifies the later operations and improves the efficiency of our attack.

We have attacked the popular CCT based CAPTCHAs in this work. Taobao, MSN and eBay CAPTCHAs which are based on this CCT mechanism are attacked with an overall precision of 51.3, 27.1 and 53.2% respectively. We have chosen these CAPTCHAs as representatives of CCT mechanism which is a very popular and

(a) **(b)** **(c)**

Fig. 2 Examples of Taobao (**a**), MSN v.2016 (**b**), and eBay (**c**) CAPTCHAs with connected characters

contemporary anti-segmentation mechanism [9]. These are implemented by popular and heavy traffic sites and likely to be attacked by malicious users in order to block them for real users. Time distribution presented in the Experimental results and analysis section shows high efficiency of our attack.

This paper is divided as follows; section 2 discusses the literature review, section 3 discusses our proposed method, Section 4 presents experimental results and analysis of our method and compares it with other approaches, while section 5 presents the conclusion and future work.

2 Related work

Breaking CAPTCHAs programmatically is not a new research paradigm. Mori and Malik have broken Gimpy and EZ-Gimpy CAPTCHAs with 33 and 92% accuracy respectively [12]. They used shape contexts to identify the objects in a severe clutter. The research has proved that adding background confusion does not prevent the CAPTCHAs from automated attacks [3, 12, 14]. In fact, for the usability purposes, the challenged text has to be emphasized by color or size so that a human can differentiate it from the background. This design restriction or usability issue is exploited by the automated attacks. Chellapilla and Siamrd used different machine learning techniques to break different types of CATPCHAs. They proposed rules for building better HIPs (Human Interaction Proofs). Their work has led to the segmentation resistant principle which is still accepted in designing the more secure Text-based CAPTCHAs [5]. Microsoft followed their principles and designed their CAPTCHAs accordingly for various online services. However their CAPTCHAs were attacked by Yan and Ahmad by using novel attacks like pixel counts after some preprocessing [16].

Yan and Ahmad also presented an approach to break Google CAPTCHA of version 2010, they have used shape patterns to segment the connected characters [6]. Bursztein et al. offered a tool to decode CAPTCHAs from 13 popular sites [3]. Starostenko et al. have proposed three-color bar code with heuristic recognition of characters. They achieved 56.5% segmentation accuracy while 95% recognition rate with an overall success of 54.6% on reCAPTCHA of version 2011 [15]. Zhang and Wen presented an approach to decode the CATPCHAs which is based on Fuzzy matching. They have proposed two methods for mask creation. These masks are used for matching the similarity with test data [18]. Huang et al. proposed the middle axis point separation and projection techniques to segment the CAPTCHAs with line cluttering and character warping [11]. Gao et al. provided an efficient analysis of text based CAPTCHAs and improved their earlier attack on hollow CAPTHCAs. They attacked 19 websites in Alexa including two versions of reCAPTHCA. They attacked the hollow CAPTHCAs, connected and disconnected CAPTHCAs and achieved success rates ranging from 12 to 88.8%. [8]. Fang et al. proposed a community driven model which is based on complex networks to segment the connected characters in text based CAPTCHAs they achieved a success segmentation rate ranging from 33 to 98% on connected and disconnected CAPTCHAs [7]. Chandavale and Sapkal presented snake segmentation algorithm and modified projection based segmentation algorithm to cut the individual characters for disconnected and connected characters respectively [4]. Their algorithms worked well for disconnected and overlapped characters with accuracy of 98% but for connected

characters the accuracy is 42%. However, for some connected and overlapped characters their algorithms failed during segmentation.

3 Our proposed method

Our proposed method is based on standard approach of CAPTCHA preprocessing, segmentation and recognition. In this method we first prepare the dataset of CAPTCHA images, apply preprocessing on these images, and partially segment the images firstly on the basis of open and closed characters and finally on the basis of recognition.

3.1 Dataset creation

Unlike MNIST, CIFAR and other datasets, no standard dataset for CAPTCHA images is available online. Therefore we downloaded CAPTCHA images from Taobao and other websites (discussed in section 3.3). Taobao CATPCHA is based on popular CCT (Crowding Characters Together) mechanism. These CAPTCHAs are alphanumeric, i.e. lower and upper case letters and numbers from 0 to 9. However alphabets such as I, L and O while number 0 and 1 are not used in their dataset. Every CAPTCHA image consists of 4 characters and all characters are connected with each other as shown in Fig. 2(a).

Once the data set is created we apply above mentioned steps, i.e. preprocessing, segmentation and recognition as illustrated in the system diagram in Fig. 3.

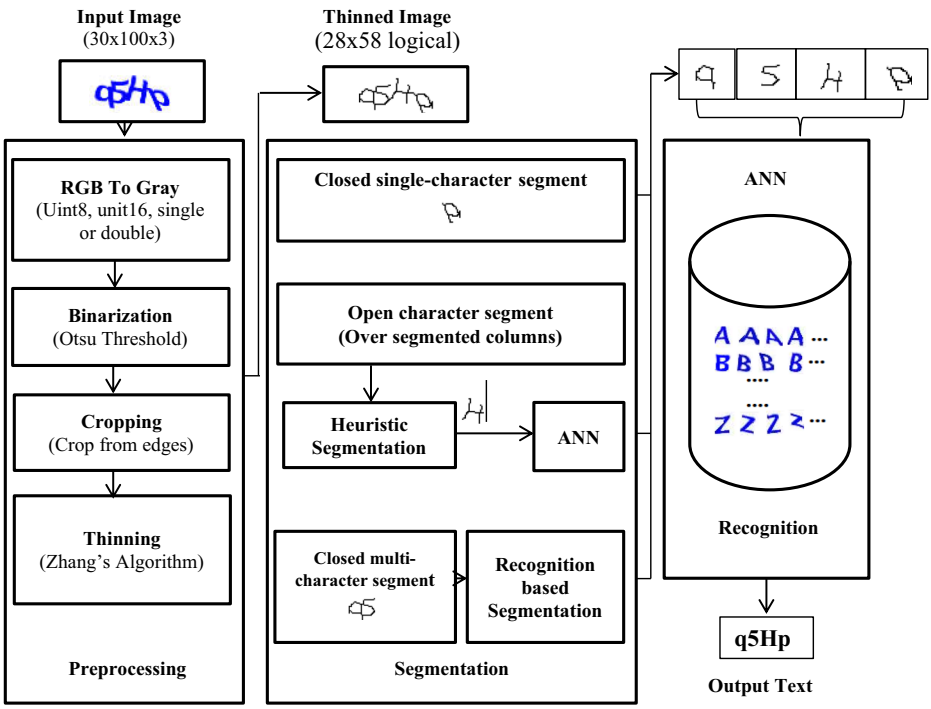


Fig. 3 System block diagram

3.2 Preprocessing

The preprocessing step includes gray scale conversion, binary conversion, cropping and thinning of the image. The obtained image is converted to gray scale using Eq. 1.

$$Y = 0.2989*R + 0.5870*G + 0.1140*B \quad (1)$$

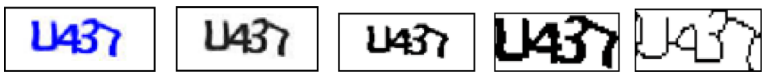
Gray scale image is further converted to binary image using Otsu threshold method [13]. The image is then cropped from edges in order to remove the extra white space along all four sides of the images which significantly reduces the size and hence increases the speed and simplifies later operations on the image. In order to remove the space surrounding the image we simply calculated the sum of all pixels over all columns and rows. We then found the left, right, up and down space which does not contain any foreground pixels and remove that space. The cropped image is thinned by using Zhang's thinning algorithm [17] to obtain its skeleton. Thinning greatly reduced the number of character pixels without removing any inherent information. It is important for later operations performed in our algorithm on the image such as pixel count per column. The main steps of preprocessing are shown in the Fig. 4.

3.3 Segmentation

There are two types of characters in the said CAPTCHA images. First type of characters is termed as 'closed characters' while other as 'open characters'. Both types along with numbers ranging from 0 to 9 are shown in Table 1.

First type of characters, i.e. the closed characters contain full loop/semi loop and each character contains at least two pixels in each column except the ligature between two characters while other type, i.e. open characters contain multiple single pixel columns within the same character as shown in Fig. 5.

In Algorithm 1, we have exploited the condition of minimum number of pixels within a column to segment the closed characters. We have measured the number of pixels in each column and if this number found less than or equal to 1 then that column is assumed as Possible Segment Column (*PSC*). A list of *PSCs* is obtained in this way containing both correct and incorrect segments. Although a number of incorrect segments are produced in open characters but closed characters are properly segmented by using this method as shown in Fig. 6(b). However two problems arise here; first is the over segmentation of open characters and second is the complex ligatures between closed characters. In order to tackle the problem of over segmentation of open characters we proposed two solutions: first solution is the estimation of minimum horizontal distance for next segmentation column within the same character in order to find the True Segmented Columns (*TSC)*. The second solution is the use of a neural classifier for correct and incorrect segments (discussed later in this section). We experimentally estimated the distance 'd' as 6 for next cut because it is the minimum character



(a) (b) (c) (d) (e)

Fig. 4 Main steps of preprocessing **a** Original image, **b** Gray scale image, **c** Binary image, **d** Cropped image, **e** Thinned image

Table 1 Open and closed characters

Type	Closed Characters	Open characters
Lower case	a, b, c, d, e, f, g, o, p, q, s, z	h, i, j, k, l, m, n, r, t, u, v, w, x, y
Upper case	B, C, D, E, F, G, O, P, Q, R S, Z	A, H, I, J, K, L, M, N, I, T, U, V, W, X, Y
Numeric	0, 2, 3, 4, 5, 6, 8, 9	1, 7

size in Taobao CAPTCHAs so we are most likely to get a complete character in case of a small character or part of the big character although it can vary depending on the type of CAPTCHA as in case of Microsoft CAPTCHAs and ebay CAPTCHAs (discussed later in this section) or other type of CAPTHCAs. This step significantly reduced the number of potential cuts of over segmented open characters as shown in Fig. 6(c).

Algorithm 1:
Suppose a CAPTCHA image represented by *I* such that
$I_{i,j}$ *I* , i = 1,2,3,….h; j = 1,2,3, ….w , where '*h*' and '*w*' are height and width of *I* respectively.
(a) Compute number of foreground pixels for each column.
$n_j =$ $_{i,j}$, j=1,2,…,w
(b) Identify the possible segment columns (*PSC*)
m = 1
For *j* = 1To *w*
If n_j
PSC(m) = j
m = m +1
end
(c) Identify the True segmented columns (*TSC*) //removing the consecutive columns within the same character
TSC = PSC(1)
For *k = 2To length(PSC)*
if *PSC(k) TSC(end) + d* //*d* is the threshold value
TSC(end+1) = PSC(k)
End

Although a huge number of over segmented columns are reduced in this step but still there were some incorrect segments in open characters as shown in Fig 6(c). This over

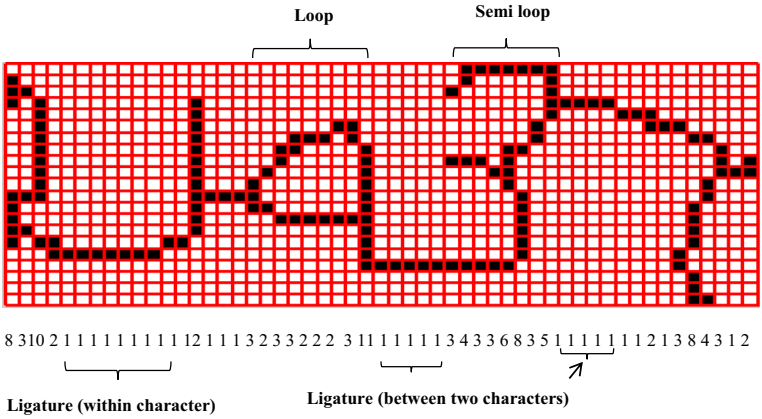


Fig. 5 Pixel distribution of open and closed characters with ligatures, loops and semi loops

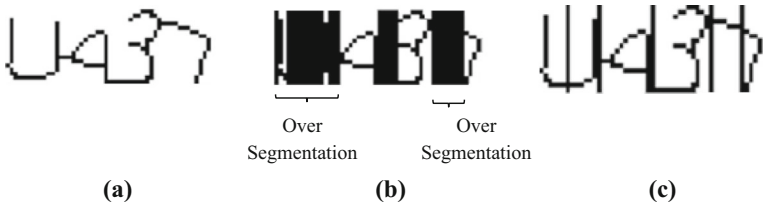


Fig. 6 **a** Thinned image, **b** Over segmented open characters and properly segmented closed characters, **c** After horizontal distance measurement and removing consecutive segments

segmentation can be further reduced by using a trained neural network. A simple artificial neural network with back propagation is trained with minimum burden of classification of correct and incorrect segments. It is important to notice that the burden of classification of correct and incorrect segments is minimized because most of the consecutive segments are already removed by using minimum distance for consecutive columns.

The coordinates of all segmentation points generated by proposed segmentation algorithm are detected by using a program in Matlab 8.3. They contain patterns of correct and incorrect classes stored in a training file. The network is trained on approximately 6400 patterns taken from 1000 CAPTCHA images. The network is trained on preprocessed and normalized data multiple times on different hidden layers, epochs and learning rates. The trained network retained the correct segments and discarded the incorrect segments with high accuracy as depicted in Fig. 7.

Our proposed method performed well to segment the closed characters. With the use of above mentioned heuristics and neural classifier, the open characters are also segmented with high accuracy. The closed characters which are connected at more than one point with adjacent characters and some overlapping characters were still not segmented properly as shown in Fig. 8.

In order to solve the problem of complex ligatures between two closed connected characters, we measured the horizontal distance of all segments and once a segment found bigger than the threshold value μ then that segment is labeled for further processing. After various experiments we analyzed this threshold value as 12 (Note that minimum character size in Taobao CAPTCHAs is 6 and maximum character size as 18), which is the average character size for Taobao CAPTCHA (though it varies with other types of CAPTCHAs and it can be adjusted according to the type of CAPTCHA).

A neural network is trained using scaled conjugate gradient algorithm with backpropagation. Cross entropy is used for calculating the performance of the network with given targets and outputs. Approximately 4000 characters were used to train this neural network. This data is randomly divided into training, validation and testing datasets as 70, 15 and 15% respectively. This network is used to segment the bigger segments (containing heavily connected and overlapped characters) based on recognition

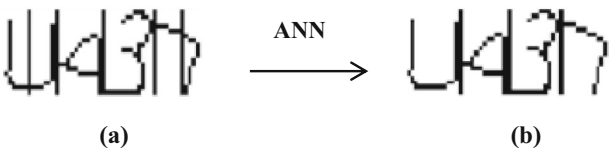


Fig. 7 **a** Over segmented Image, **b** ANN discarding the incorrect segment segments

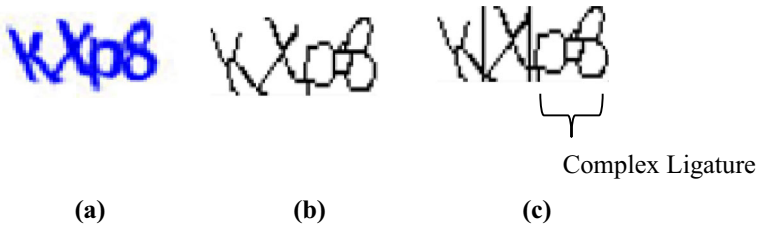


Fig. 8 **a** Original image, **b** Thinned Image, **c** Complex ligatures between two closed characters

confidence. This neural classifier has served both in recognition of properly segmented characters as well as non-segmented characters by using recognition based segmentation.

The neural network is constructed by the feature vectors obtained by using local and global geometric features of character skeleton [2, 10]. This algorithm first defines the universe of discourse of a character skeleton, i.e. the shortest matrix which fits the entire character skeleton, the image is then divided into 9 equal zones and the features of each window are calculated. Particular pixels are defined as the starters, intersections and minor starters. The line segments in each zone are classified as horizontal, vertical, right diagonal or left diagonal line. After classification of line segments, the feature vectors are formed for each zone based on the following information in each zone:

I. Number of horizontal lines
II. Number of vertical lines.
III. Number of Right diagonal lines.
IV. Number of Left diagonal lines.
V. Normalized Length of all horizontal lines.
VI. Normalized Length of all vertical lines.
VII. Normalized Length of all right diagonal lines.
VIII. Normalized Length of all left diagonal lines.
IX. Normalized Area of the Skeleton.

The area of skeleton, Euler number, regional area and eccentricity are also calculated. Further details on this geometric feature extraction method can be found in [2, 10].

The neural network trained on individual characters is used for recognition based segmentation. The labeled big segments containing 2 or more than two characters (rare cases) are identified with neural confidence and then segmented based on the obtained results. This recognition based segmentation method is discussed as follows:

As shown in Algorithm 2, we measured the size of all segments w_n, if size of any segment is found bigger than the threshold value μ then it is assumed as possible double character segment, otherwise it is kept as correct segment containing a character and recognized with the use of neural classifier. For the big segment we calculated k sub windows such that $k = \mu \pm j$, Where $j \in \mathbb{N}$. The feature vectors f_k for all values of k are sent to the pre trained neural classifier. The results are saved as the score returned by the classifier in terms of probability p_k for all values of k. The maximum value V from all values of p_k, is computed at m^{th} column where V is the highest score returned by classifier in term of probability of a particular character. We cut the recognized character with highest confidence at m. The $(m + 1)^{th}$ column is labeled as the next segmentation

point. Finally the remaining part of the segment w_n is compared with threshold value μ, If it is bigger than μ then we repeat from step ii, otherwise results are displayed.

Algorithm 2.
Let w_n is the size of n^{th} segment such that $w_n \in I$ (Image) and $n \in \mathbb{N}$
We assumed threshold value for a bigger segment as μ

I. Compare the segment w_n with μ
   If $(w_n > \mu)$
   $k = \mu \pm j$ where $j \in \mathbb{N}$
II. Calculate feature vectors f_k for all values of k
III. Save P_k such that P_k is the probability of k^{th} segment returned by classifier
IV. Calculate the maximum value V from all values of P_k at any column m.
   $V = max(P_k)$
V. Label the $(m+1)^{th}$ column for next segment
VI. Compare the remaining part of the segment w_n with threshold value μ.
   If $(size(w_n)-m > \mu)$
   Go to Step II.

3.4 Experiments on other datasets

In order to verify the generic nature of our algorithm we performed similar experiments with slight variations on other data sets like MSN and eBay CAPTCHAs. MSN v.2016 CAPTCHAs consist of 6 to 8 characters where all the characters are closely connected together. Reduced number of characters is used in MSN CAPTCHAs for usability purposes and it consists of only 23 classes instead of 62. Foreground characters are blue while background is white. On the other hand eBay CAPTCHAs consist of 6 digits and contain only numbers from 0 to 9 and some digits are connected and/or overlapped with each other. Many colors are used for the foreground while background color is always kept as white as already shown in Fig. 1. We attacked the above said CAPTCHAs with our proposed algorithm and achieved promising results on both these CAPTCHAs. The results on these CAPTCHAs and Taobao CAPTCHA are discussed in section 4.

Table 2 Average time distribution of all the processing steps on Taobao, MSN and eBay CAPTCHAs

Processing Steps	Time elapsed in each step (ms)		
	Taobao	MSN	eBay
Image Acquisition	3.62	5.13	4.52
Gray scale conversion	1.32	1.43	1.25
Binarization	1.82	2.31	1.90
Cropping	1.22	2.21	1.85
Thinning	17.09	22.27	19.83
Segmentation (heuristic + ANN based)	77.23	89.19	85.10
ANN Recognition	11.2	15.26	13.25
Total	113.5	137.8	127.7

Table 3 Results of our proposed algorithm

CAPTCHA	Success Recognition Rate SRR (%)	Segmentation Success Rate SSR (%)	Overall Precision OP(%)	No. of images in a dataset
Taobao	96.5	59.25	51.3	1000
MSN	91.25	51.5	27.1	500
eBay	97.5	62	53.2	1000

4 Experimental results and analysis

Average time distribution of our attack on all datasets is given in Table 2. The results have been obtained on a Computer with Core i3, 2.2 GHz, 6 GB RAM, with windows 7×64 operating system.

Overall Precision (OP) is calculated using Eq. 2.

$$OP(\%) = SSR\%(SRR\%)^N \tag{2}$$

Where SSR(%) is the Segmentation Success Rate, SRR(%) is Success Recognition Rate and N is number of characters in an image [15]. OP calculates the overall success rate, i.e. segmentation and then recognition. OP depends on the segmentation accuracy of the algorithm, accuracy of the classifier/ANN and number of characters in a challenged image. The SSR is the rate at which a segmentation algorithm can segment the number of characters in a set of CAPTCHAs [11]. For example if a segmentation algorithm can segment 250 characters from 100 images in Taobao CAPTCHAs, the segmentation success rate will be $250/400 = 0.625$ or 62.5%. SRR is the rate at which a classifier can predict the individual characters. These results are shown in Table 3.

The CAPTCHA is ideally assumed to be broken less than 0.01% of the time by automatic scripts [16], it can be seen from results that it is far beyond this imagination. It can be observed in Table 4 that as overall, our method offers improved results as

Table 4 A comparison of our algorithm with other algorithms

Author	Same Datasets	Overall Precision (OP)
Gao [8]	Taobao	42.8%
	eBay	49.8%
	MSN	16.2%
Fang [7]	Taobao	33%
Bursztein [3]	eBay	51.39%
Proposed algorithm	Taobao	51.3%
	eBay	53.2%
	MSN v. 2016	27.1%
Author	Other Datasets	Overall Precision (OP)
Starostenko [15]	reCAPTCHA v.2011	40.4%
Ahmad [6]	Google	33%
Huang [11]	Yahoo	60.57%
	MSN v.2008	41.13%

Fig. 9 **a** Overlapped characters: 'X' and 'd' in MSN, **b** 'v' and 'j' in Taobao CAPTCHA

compared to other methods evaluated on the same datasets. Table 4 also presents results from some other datasets including reCAPTCHA v.2011, Google, Yahoo and an old version of MSN.

As shown in Table 4, on Taobao and eBay CAPTCHAs our proposed method has provided better precision than other methods. On MSN CAPTCHAs although the precision is lower than Taobao and eBay CAPTCHAs but it is still far beyond the security imagination of an ideal CAPTCHA [16]. This version of MSN CAPTCHA is not yet attacked by any other researcher to the best of our knowledge. Although promising results on above discussed CAPTHCAs were achieved but still there were few cases where our algorithm could not segment the said CAPTHCAs. This was mainly due to heavily overlapped characters as shown in Fig. 9.

Based on our attack results and analysis of the said CAPTCHAs, we strongly propose following design guidelines for better security measures:

- *Offer multiple schemes:* offering multiple schemes for the same web resource can be useful. Every time the user tries to access the same resource can be offered a different randomized scheme.
- *Improvements in the overlapping principles:* overlapping of the characters is the most powerful resistant mechanism against segmentation attacks. Individual characters can be overlapped at random directions and can offer serious resistance against segmentation attacks such as ours.
- *Characters of local languages*: Depending on the users of a particular region, they can be offered different types of CAPTCHAs containing characters from the contents written in the local languages. It will not only improve the security but will also help in digitization of the books written in local languages.
- *Randomize the length of CAPTCHAs*: Using fixed number of characters (like Taobao and eBay CAPTCHAs) is insecure. It provides too much knowledge to the attackers. Randomizing the number of characters with increased length can be helpful.
- *Multiple characters position*: Instead of using the fixed position for characters randomizing their positions can avoid the calculated guess.

5 Conclusion and future work

In this research work we have attacked the CAPTCHAs which are based on popular CCT mechanism. In our proposed method simple image processing techniques including thresholding, thinning and pixel count are used to exploit the open and closed characters, the closed characters are nicely segmented using pixel count per column method while open characters are over segmented. In order to reduce the over segmentation, firstly a

heuristic based method and secondly a neural classifier is used. These two approaches largely reduced the over segmented columns. However few closed characters were not segmented due to multiple connections between adjacent characters. Therefore a recognition based segmentation method is used to segment these characters. Using the said methods, results were improved up to 51.3%. Similar experiments were performed on other datasets like MSN and eBay CAPTHCAs with success rate of 27.15 and 53.2% respectively. Few characters were not segmented correctly due to heavy overlapping between adjacent characters. In the end we proposed design guidelines based on the weaknesses explored in the current design.

In the future various other types of CAPTCHAs will be analyzed to verify their strengths against automated attacks. The CAPTCHAs with heavily overlapping characters will be analyzed for devising better segmentation mechanism.

References

1. Ahn LV, Blum M, John L (2004) Telling humans and computers apart automatically. Commun ACM 47(2): 56–60
2. Blumenstein M, Verma B, Basli H (2003) A novel feature extraction technique for the recognition of segmented handwritten characters. In: Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference (pp. 137–141). IEEE
3. Bursztein E, Martin M, Mitchell J (2011) Text-based CAPTCHA strengths and weaknesses. In: Proceedings of the 18th ACM conference on Computer and communications security, pp. 125–138. ACM
4. Chandavale AA, Sapkal A (2012) A new approach towards segmentation for breaking CAPTCHA. In: International Conference on Security in Computer Networks and Distributed Systems (pp. 323–335). Springer Berlin Heidelberg
5. Chellapilla K, Larson K, Simard PY, Czerwinski M (2005) Building segmentation based human-friendly human interaction proofs (HIPs), Human Interactive Proofs pp. 1–26. Springer, Berlin Heidelberg
6. El Ahmad AS, Yan J, Tayara M (2011) The robustness of Google CAPTCHA's. Computing Science, Newcastle University
7. Fang K, Bu Z, Xia ZY (2012) Segmentation of CAPTCHAs based on complex networks. In: International Conference on Artificial Intelligence and Computational Intelligence (pp. 735–743). Springer Berlin Heidelberg
8. Gao H, Wang X, Cao F, Zhang Z, Lei L, Qi J, Liu X (2016) Robustness of text-based completely automated public turing test to tell computers and humans apart. IET Inf Secur 10(1):45–52
9. Gao H, Wang W, Fan Y, Qi J, Liu X (2014) The Robustness of "Connecting Characters Together" CAPTCHAs. J Inf Sci Eng 30(2):347–369
10. Gaurav DD, Ramesh R (2012). A feature extraction technique based on character geometry for character recognition. arXiv preprint arXiv:1202.3884
11. Huang SY, Lee YK, Bell G, Ou ZH (2010) "An efficient segmentation algorithm for CAPTCHAs", with line cluttering and character warping. Multimed Tools Appl 48(2):267–289
12. Mori G, Malik J (2003) Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA. In: Computer Vision and Pattern Recognition, (Vol. 1, pp. I-134). Proceedings of IEEE Computer Society Conference IEEE
13. Otsu N (1975) A threshold selection method from gray-level histograms. Automatica 11:285–296
14. Simard PY (2004) Using machine learning to break visual human interaction proofs. Adv Neural Inf Proces Syst 17:265–272
15. Starostenko O, Cruz-Perez C, Uceda-Ponga F, Alarcon-Aquino V (2015) Breaking text-based CAPTCHAs with variable word and character orientation. Pattern Recogn 48(4):1101–1112
16. Yan J, El Ahmad AS (2008) A low-cost attack on a microsoft CAPTCHA. In: Proceedings of the 15th ACM conference on Computer and communications security (pp. 543–554) ACM
17. Zhang TY, Suen CY (1984) A fast parallel algorithm for thinning digital patterns. Commun ACM 27(3): 236–239
18. Zhang H, Wen X (2014) The recognition of CAPTCHA based on fuzzy matching. In: Foundations of Intelligent Systems (pp. 759–768). Springer Berlin Heidelberg

Rafaqat Hussain is pursuing his Ph.D. in School of Computer Science and Engineering, University of electronics Science and Technology of China. His areas of interest include Image Processing, Machine learning, and Artificial Intelligence. He is awarded MS in 2014. He is serving as Assistant Professor in Shah Abdul Latif University, Pakistan.

Hui Gao received his PhD degree in computing science from the University of Groningen (the Netherlands) in 2005. In 2006, he joined data mining lab at the School of Computer Science and Engineering, UESTC. Currently he is working as Professor in the Department of Computer Science. His research areas include Data Mining, Formal verification, Image Processing and Analysis and Computational Intelligence.

Riaz Ahmed Shaikh has completed his Ph.D. from University of Electronic Science and Technology, Chengdu, Sichuan, China in 2016. He is currently working as Lecturer in Department of Computer Science, Shah Abdul Latif University, Pakistan. His current research areas are Image Processing, Computer Vision and Artificial Intelligence.