



A machine learning attack against variable-length Chinese character CAPTCHAs

Xing Wu¹ · Shuji Dai¹ · Yike Guo¹ · Hamido Fujita²

Published online: 20 November 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

CAPTCHA (Completely Automated Public Turing test to tell Computer and Human Apart) is widely used as a standard security mechanism to protect resources on websites. Among various kinds of CAPTCHAs, the text-based CAPTCHA is the most popular scheme, which consists of English letters, Arabic digits and other character sets, such as Chinese characters. Due to the large quantity of Chinese characters and complicated character structure, it is difficult for bots to crack Chinese character CAPTCHAs. Thus, Chinese character CAPTCHAs have been widely applied in China. Nevertheless, effective offensive approaches are necessary to help CAPTCHA designers find security vulnerabilities to improve defense mechanisms. To deal with variable-length Chinese character CAPTCHAs with noises, an automatic attacking approach is proposed, which includes preprocessing, character segmentation and character recognition. For character recognition, two methods are proposed: MGLCR (Multi-scale Gabor and Logistic regression based CAPTCHA Recognition) and CCR (Convolutional neural network based CAPTCHA Recognition). MGLCR extracts features by multi-scale Gabor filters and classifies characters with logistic regression. CCR extracts features and recognize characters automatically with CNN (Convolutional Neural Network). Experimental results show that the proposed approaches are efficient in attacking variable-length Chinese character CAPTCHAs with noises. The pros and cons of proposed MGLCR and CCR methods are discussed, which outperform state-of-the-art methods. Besides, the proposed methods could achieve satisfactory results in breaking the mixed character CAPTCHAs which consist of English letters, Arabic digits, Chinese characters and mathematical operators.

Keywords Chinese character · CAPTCHA · Segmentation · MGLCR · CCR

1 Introduction

1.1 Background

CAPTCHA, as a network security technology, can protect passwords from being hacked violently and prevent a

large number of automatic machine requests to keep normal operation of websites [1, 39]. In order to avoid bots attacking and keep user-friendly for users, a CAPTCHA design should follow three principles [19]. Firstly, randomness is necessary in generating a CAPTCHA to ensure that the recognition task varies at each time. Secondly, a CAPTCHA should be easy for humans to recognize. Finally, it should be difficult for a machine to crack. According to the above requirements, there are various types of CAPTCHAs [34], such as text-based CAPTCHAs [38], image-based CAPTCHAs [12], sound-based CAPTCHAs [20], video-based CAPTCHAs [11], etc. Currently, the text-based CAPTCHA is the most widely used type due to low generation costs, which typically consists of letters and digits [8, 41]. To make text-based CAPTCHAs unrecognizable for the state-of-the-art of pattern recognition algorithms, noises are always added to blur CAPTCHAs images which

✉ Xing Wu
xingwu@shu.edu.cn

Shuji Dai
shujidai@hotmail.com

Yike Guo
y.guo@imperial.ac.uk

Hamido Fujita
HFujita-799@acm.org

¹ School of Computer Engineering and Science, Shanghai University, Shanghai, 200444, China

² Intelligent Software Systems Laboratory, Iwate Prefectural University, 020-0193, Iwate, Japan

include different background colors, text distortion and clutter. Straight lines and arcs, background textures and background colors are typical examples of clutter [2, 9]. However, with the development of pattern recognition technology, the traditional simple character set composed of digits and letters is incapable to guarantee the security of websites. Thus more complicated CAPTCHAs are needed to conquer the state-of-art cracking algorithms. The set of Chinese characters is much bigger than English letters and Arabic digits, which increase the complexity of automatic recognition. Furthermore the structure of Chinese characters is more complicated than English letters and Arabic digits. Thus text-based CAPTCHAs with Chinese character set have been widely used in most Chinese websites. Figure 1 displays an example of Chinese character CAPTCHA.

Research on cracking CAPTCHAs can help designers discover security vulnerabilities so as to encourage more secured and user-friendly CAPTCHAs to be designed. The standard approach to crack CAPTCHAs follows the procedure: preprocessing, denoising and binarizing the original CAPTCHA images, character segmentation and character recognition. This procedure is illustrated in Fig. 2.

During the preprocessing, different types of noises are removed to obtain the binary images [6, 36]. Image denoising methods contain spatial filters (e.g. mean filter, median filter and Gaussian filter), frequency domain filters (e.g. high-pass filter and low-pass filter) and morphological operators (e.g. open-close operations and morphological reconstruction). Morphological reconstruction is a powerful transformation method which is often used in image segmentation [29, 32, 40]. Binarization methods include local threshold methods and global threshold methods [27], among which is Otsu method [46].

In image segmentation part, positions for each character in the text-based CAPTCHA image is identified which will be segmented for recognition. Character segmentation methods include vertical projection methods [28, 43], connectivity analysis [5], etc. Besides, mean shift is a robust feature space analysis method, which could be used for clustering [3, 4], image segmentation [30], target tracking [17], etc. Actually, preprocessing and segmentation are much more difficult and time-consuming than recognitions since machine learning algorithms can efficiently solve the recognition problem but there are no general algorithms for preprocessing and segmentation problems in various scenarios.



Fig. 1 A Chinese character CAPTCHA

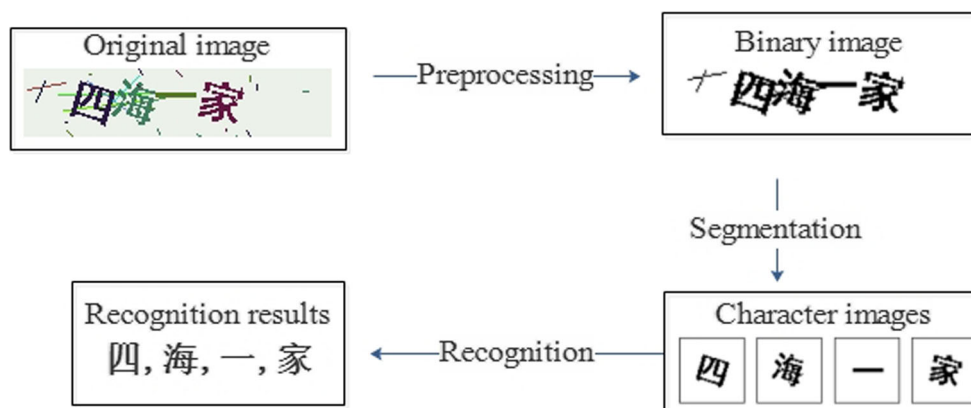
Segmented characters are classified for character recognition, in which feature vectors are extracted as the descriptions of each character images. The recognition accuracy usually depends on pattern recognition and machine learning methods, such as template matching, feature extraction and pattern classification, neural networks and deep neural networks [18]. Methods based on feature extraction and pattern classification, Gabor feature [21, 31], local binary pattern(LBP) [10, 25] and primary component analysis (PCA) [23] are mainly used to describe character images. Pattern classification methods include logistic regression, support vector machine(SVM), etc. Obviously, above mentioned pattern recognition methods rely on feature extraction results. With the rise of deep neural network, researchers have been using them to solve character recognition problem with its strong feature learning ability [26]. CNN, as a deep neural network model, has made breakthroughs in the field of pattern recognition, which is good at learning spatial features in images [15, 33, 37]. For recognizing characters, CNN is able to learn features automatically. Furthermore, CNN achieves deformation invariance, so that it is suitable to deal with the character recognition problem [24, 42].

1.2 Contribution

There are three main contributions of our research as listed below:

- i Firstly, we are committed to dealing with variable-length Chinese character CAPTCHAs with different types of noises. To the best of our knowledge, our research is the first attempt to attack variable-length Chinese character CAPTCHAs. The Chinese character set, consisting of 5,000 frequently used Chinese characters, is much bigger than the sets of 26 English letters and 10 Arabic digits. What's more, the complicated structure of Chinese characters also increases the complexity of CAPTCHAs segmentation and recognition.
- ii Secondly, two novel methods are proposed to crack variable-length Chinese character CAPTCHAs. One method is called MGLCR which is a combination of pattern recognition methods, which are tailored and combined in an elaborate order. The other one is called CCR which is inspired by CNN to extract features and recognize characters automatically. The network architecture is properly designed to meet the challenge.
- iii Finally, the proposed methods are also valid for the mixed character CAPTCHAs consisting of English letters, Arabic digits, Chinese characters and mathematical operators, which outperform state-of-the-art methods.

Fig. 2 Procedure of attacking Chinese CAPTCHAs



1.3 Structure

The remainder of this paper is organized as follows. In Section 2, the related work is discussed. Section 3 presents the proposed approaches for attacking Chinese character CAPTCHAs, including the procedure of preprocessing, segmentation and recognition. The experimental design and results are elaborated in Section 4. Finally, conclusions and future work are drawn in Section 5.

2 Related work

The text-based CAPTCHA is one of the reverse Turing tests that distinguish an access by a computer program such as a crawler from an access by a human being, which uses the difference between computers' and humans' shape recognition ability. There are some researches focusing on cracking it with the help of pattern recognition methods and deep neural networks. Most of the existing CAPTCHA cracking methods work for English letters and Arabic digits, while methods for Chinese character set are rare. What's more, it's worth noting that there are no existing methods available for attacking variable-length Chinese CAPTCHAs. Thus methods introduced in this section are mainly for CAPTCHAs composed of English letters and Arabic digits, or fixed-length Chinese character CAPTCHAs [26], which play a referential role in our research.

Initially the cracking methods for CAPTCHAs based on pattern recognition are introduced. Yin et al. proposed an approach based on dense scale invariant feature transform (DENSE SIFT) and random sample consensus (RANSAC) algorithm for recognition of distorted and merged CAPTCHAs [45]. Their dataset is composed of three kinds of CAPTCHAs captured from websites, which includes 220 pictures for training set and 600 pictures for test set respectively. This method achieved good performance on CAPTCHAs with different difficulty levels. Elie Bursztein et al. introduced a method to solve CAPTCHAs

in a single step which used machine learning to address the segmentation and recognition problems simultaneously [7]. Their dataset is created by a methodology focused on unbroken real-world CAPTCHA schemes. Starostenko et al. presented a method for automatic segmentation and recognition of CAPTCHAs with variable orientation and random collapse of overlapped characters based on SVM classifier [35]. Their dataset consists of 5,000 downloaded images. A system attacking method for text-based CAPTCHAs was proposed by Hussain et al., which includes simple image processing techniques, pixel count methods and an artificial neural network [16]. They used the dataset which consists of 1,000 CAPTCHAs of Taobao, 500 CAPTCHAs of MSN and 1,000 CAPTCHAs of eBay respectively. Gabor filters are firstly used to extract character components along four different directions for a generic attack with a dataset of 1,000 images downloaded from Amazon [44]. Our MGLCR method is inspired by this research.

Additionally, some methods with neural networks for CAPTCHA segmentation are presented. Grag et al. used CNN layers followed by a dense layer and a recurrent neural network layer instead of the conventional method to segment and recognize individual characters to break text-based CAPTCHAs [14]. Their dataset is generated, which contains 20,000 images. Wang et al. combined convolutional neural network and self-adaptive algorithm to break synthetic multi-digit text-based CAPTCHA [41]. Their dataset consists of 300,000 downloaded images. More recently, Gao et al. analyzed the security of the two-layer CAPTCHA and proposed a method to attack the Microsoft two-layer CAPTCHA based on LeNet-5, which is a radical CNN model [13].

Last but not the least, the CAPTCHA recognitions with deep neural networks are discussed. In recent years, researchers have made progress in the attacking of CAPTCHAs based on deep neural networks, especially convolutional neural networks. Karthik et al. proposed a CNN method to crack the Microsoft CAPTCHAs which are composed of English letters and Arabic digits [18]. Lv et al. proposed an approach based on CNN to improve

the recognition accuracy of fixed-length Chinese character CAPTCHAs with distortion, rotation and background noises [26]. They used the dataset which is generated by CAPTCHA generator for the training set of 5,000 Chinese character CAPTCHAs and the test set of 15,000 Chinese character CAPTCHAs. However the segmentation method for the CHAPTCHAs is not included in this paper and the proposed method deals with the fixed-length CAPTCHA only, which contains four Chinese characters. Compared with it, the proposed MGLCR and CCR methods can deal with variable-length Chinese character CAPTCHAs due to the proposed segmentation method discussed in Section 3.2, which is of practical usage. The proposed network structure is similar to LeNet-5 [22] while the network structure in the proposed CCR method is designed for Chinese character CAPTCHAs.

3 The proposed approach

As shown in Fig. 2, an automatic attacking approach includes three steps: preprocessing, character segmentation and character recognition. Accordingly, the proposed approach is introduced in three parts. Section 3.1 discusses the preprocessing. Section 3.2 introduces the character segmentation. Section 3.3 presents two recognition methods: the MGLCR(Multi-scale Gabor and Logistic regression based

CAPTCHA Recognition) and the CCR(Convolutional neural network based CAPTCHA Recognition) methods.

3.1 Preprocessing

In the proposed methods, the preprocessing is used to binarize and denoise CAPTCHA images, which is based on Gaussian smoothing, morphological reconstruction and Otsu method. The preprocessing can be adapted to different types of noises. Particularly, it is able to effectively binarize a CAPTCHA image with gradient background color. The flow chart of proposed preprocessing is shown in Fig. 3.

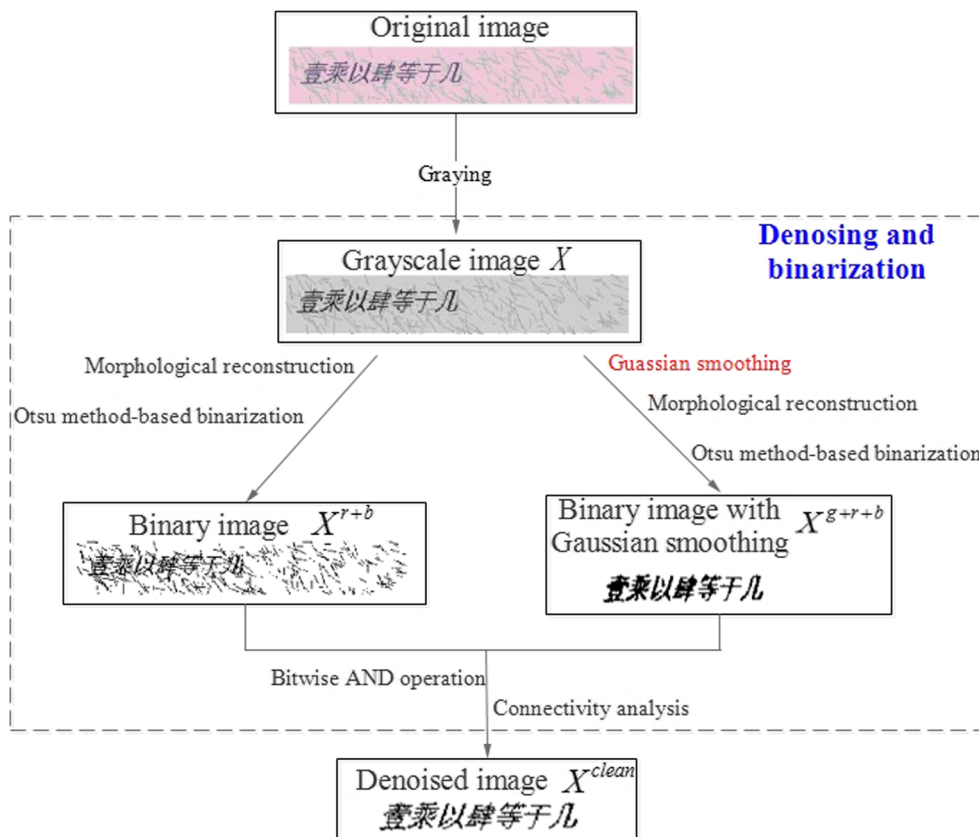
As shown in Fig. 3, the CAPTCHA image is grayed firstly, then Gaussian smoothing and morphological reconstruction are used to denoise the grayscale image. Furthermore Otsu method is applied to binarize the image, which takes the noises of the CAPTCHA image as the background and the characters as the foreground.

It should be noted that, the denoising is an optional step according to the characteristics of different types of CAPTCHAs.

3.1.1 Graying

The graying method aiming at converting RGB to greyscale is designed to perform a weighted average operation. The

Fig. 3 Flow chart of preprocessing



weight of each RGB channel should follow the recommended grayscale conversion formula.

$$V = 0.299R + 0.587G + 0.114B \tag{1}$$

where R , G and B are the RGB three-channel pixel values of the CAPTCHA image and V is the gray values of the grayscale CAPTCHA image.

A grayscale image is denoted as $X = x_i, i = 1, \dots, |X|, |x_i \in [0, 255]$ after graying.

3.1.2 Denoising and binarization

A global threshold obtained from Otsu method is not valid for a CAPTCHA image with gradient background color, as shown in Fig. 4c. To cope with this problem, Gaussian smoothing and morphological reconstruction are adopted to remove noises and to average gradient background color. As shown in Fig. 3, the denoising and binarization part has two branches before bitwise AND operation. Two branches both have morphological reconstruction and Otsu method-based binarization parts. Nevertheless the right branch has an additional Gaussian smoothing part which is adopted before morphological reconstruction. The Gaussian smoothing part works to denoise straight lines, but the side effect is that characters are also blurred. To obtain a denoised image which combines advantages of two branches, a bitwise AND operation is performed for these two branches. With denoising and binarization, the image with gradient background color can be effectively processed as shown in Fig. 4d.

For the left branch in Fig. 3, morphological reconstruction and binarization are included. During morphological reconstruction process, the grayscale image is used as the mask image I . The number 102 is subtracted from the gray value of each pixel in the grayscale image to get the seed image J . Both the mask image and the seed image have a range of pixel values of $0, 1, \dots, 255$. The grayscale reconstruction is performed according to the following formula:

$$\begin{aligned} \forall p \in D, \gamma^{rec}(J, I)(p) \\ = \max\{k \in [0, N - 1] | p \in \gamma^{rec}(T_k(J), T_k(I))\} \end{aligned} \tag{2}$$

where $T_k(\cdot)$ represents a binary image obtained by binarizing a grayscale image at threshold k .

Actually, the output of morphological grayscale morphology is an image after morphological expansion of the local extreme points, as shown in Fig. 6(b), denoted as $X^{dilated}$. Then $X^{dilated}$ is subtracted from the grayscale image X to get the difference image X' . Then, the binarization based on Otsu method is conducted on X' .

The Otsu method selects a global threshold to binarize the image. The image gray level is L and the binarization threshold is k . With the binarization threshold, the image pixels are divided into two classes: the foreground and the background which are denoted as f and b respectively. The variances of two classes are denoted as σ_f^2 and σ_b^2 respectively. The sizes of two classes are denoted as ω_f and ω_b respectively. The average gray values of two classes are denoted as μ_f and μ_b respectively. Then the intra-class variance and inter-class variance are defined as follows:

$$\begin{aligned} \sigma_W^2 &= \omega_f \sigma_f^2 + \omega_b \sigma_b^2 \\ \sigma_B^2 &= \omega_f \omega_b (\mu_f - \mu_b)^2 \end{aligned} \tag{3}$$

where σ_W^2 is the intra-class variance and σ_B^2 is the inter-class variance. These variances should obey the formula described as follows.

$$\sigma_W^2 + \sigma_B^2 = \sigma^2 \tag{4}$$

where σ^2 is the variance of all pixels.

Therefore, minimizing the intra-class variance is equal to maximize the inter-class variance. The threshold k is obtained by minimizing the intra-class variance or maximizing the inter-class variance.

The global binarization threshold α is calculated from the difference image X' . With this threshold, each pixel in the grayscale image is divided into foreground class or background class. The background pixel values are set to 255 and the foreground pixel values are set to 0.

$$x_i^b = \begin{cases} 0 & x_i < \alpha, \\ 255 & x_i \geq \alpha. \end{cases} \tag{5}$$

Fig. 4 Usage of morphological reconstruction on an CAPTCHA image with gradient background color. **a** Original image; **b** Grayscale image; **c** Binarized image; **d** Binarized image with morphological reconstruction



where x_i^b is the i -th pixel of the binary image, x_i is the i -th pixel of the grayscale image. Eventually, the binary image is described as follows:

$$X^b = x_i^b, i = 1, \dots, |X^b| |x_i^b \in 0, 255 \tag{6}$$

where X^b is binary image.

It should be noted that it is necessary to standardize the foreground and background pixel values after binarization. After that the foreground characters are black, and the background is white. If the background color is darker than the foreground color, the binary image after the preprocessing will have a white foreground and a black background. For unification, the foreground and background pixel values of substandard binary images are exchanged based on the assumption that the number of foreground character pixels is less than the number of background pixels.

After binarizing X^r , a binary image X^{r+b} is obtained. As shown in Fig. 5d, the main noises in the output image have been removed.

For the right branch in Fig. 3, Gaussian filtering($\sigma = 0.7$) is conducted firstly to get image X^g . After that, morphological reconstruction is used to find local maximum points. After morphological expansion of these local extreme points, the image $X^{g+dilated}$ is obtained. Then $X^{g+dilated}$ is subtracted from the grayscale image X to get the difference image X^{g+r} . Then binarization based on Otsu method is conducted on X^{g+r} . Binarization procedure is same as the left branch. After binarization, the binary image X^{g+r+b} is obtained.

Thus, two binary difference images X^{r+b} and X^{g+r+b} are obtained. X^{r+b} can maintain the original structure of the characters and avoid the noises of gradient background color, which still have lines and arcs. As a result of the Gaussian smoothing, clutter can be removed partly, but the characters are also blurred in X^{g+r+b} , which will bring difficulties to the recognition part. Thus, bitwise AND operation on X^{g+r+b} and X^{r+b} is performed to combine advantages of these two kinds of binary images. The denoised

image is denoted as X^{clean} . Figure 3 shows two binary images and the denoised image X^{clean} .

$$X^{clean} = X^{g+r+b} \text{ AND } X^{r+b} \tag{7}$$

After denoising and binarization steps, the image may still have some lines or arcs, thus connectivity analysis is needed for further denoising. The binary image X^{bin} is divided into eight connected regions and the area of each region is calculated. Then the regions with area less than 9 are filtered out.

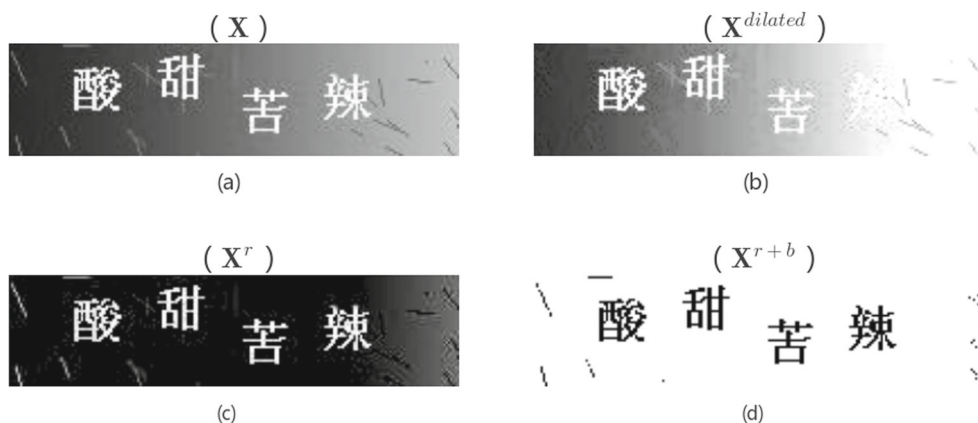
3.2 Segmentation

After preprocessing, character segmentation will be carried out on denoised and binary images. The segmentation procedure includes data dimension classification, clustering, clustering correction and morphological open operation. During segmentation procedure, mean shift clustering is used on both spatial coordinates and the color space coordinates of character pixels. The obtained cluster center represents the center point of each character. Because non-parametric mean shift is a density estimation method, it is unnecessary to specify the length of characters in advance when clustering the CAPTCHA images with variable-length characters. Besides, the segmentation method based on mean shift clustering not only takes into account the spatial information, but also considers the color space information. Therefore, it is adaptive when there are different colors between characters and clutters, even between characters and characters. In addition, different clustering strategies are used for different types of CAPTCHAs to archive satisfactory results.

3.2.1 Data dimension classification

Before clustering, classifying the pattern of the CAPTCHA images is necessary so as to determine the data dimension required for clustering. It should be noted that the image pattern here refers to the color pattern of the original

Fig. 5 Procedure of morphological reconstruction. **a** grayscale image; **b** image after morphological expansion of the local extreme points; **c** difference image; **d** binary image



CAPTCHA image, so that the color information is preserved to enhance the segmentation effect.

Basically, there are three color patterns for CAPTCHA images: grayscale image, color image with uniform character color and color image with non-uniform character colors. Examples of three color patterns are displayed in Fig. 6. For the first two patterns, the spatial coordinates (r, c) and the gray value v of the character pixel are used for clustering, denoted as (r, c, v). For the third pattern, the set of data points is denoted as (r, c, l, a, b), in which r, c are the spatial coordinates and (l, a, b) are values of LAB color space.

3.2.2 Clustering

The feature space depends on dimension of data points. After the dimension has been fixed, mean shift is used to cluster the data points by classifying character pixels and finding cluster centers.

Bandwidth used in nonparametric mean shift is estimated automatically by the following steps:

- (1) Taking all the foreground pixel sets in the binary image as the data set $X_{fg}^b = x | x \in X^b, x = 0$ for estimation.
- (2) Taking 20% of the foreground pixels as the K parameter of the K-nearest neighbor method.
- (3) Calculating the furthest distance d_i to the nearest K pixels for each point x^i in X_{fg}^b .
- (4) Calculating the average farthest distance $\frac{1}{|X_{fg}^b| \sum_1^{|X_{fg}^b|} d_i}$ of all points in X_{fg}^b .

The estimated bandwidth is used as the clustering window diameter in the mean shift method. Calculations for each window are listed as follows:

- (1) Calculating the mean shift vector for the center of the window:

$$ms = \left[\sum_{i=1}^n x_i g \left(\left\| \frac{x - x_i}{h} \right\|^2 \right) \right] / \left[\sum_{i=1}^n g \left(\left\| \frac{x - x_i}{h} \right\|^2 \right) \right] - x \tag{8}$$

Fig. 6 CAPTCHA images of different color patterns. **a** grayscale image; **b** color image with uniform character color; **c** color image with non-uniform character colors



(a)



(b)



(c)

- where $g(\dots)$ is a Gaussian kernel function, h is the bandwidth and x is the vector of the window center.
- (2) Moving the window center along the mean shift vector and the end point is the new window center.
 - (3) If the window center does not converge, it will repeat step 1. Otherwise, the algorithm outputs an extreme point for each window and a number of parallel windows finally converge to obtain extreme points as the center of the cluster. All points converging to the same extreme point belong to a class.

3.2.3 Clustering correction

Figure 7 displays two segmentation cases without clustering correction, in which an original image, a binary image after preprocessing and the result of segmentation are listed. The orange stars marked in the binary image indicates the clustering center of each character, which is calculated by the mean shift method. In the segmentation results, different colors represent different pixel classes. Due to the adhesion between characters, it is easy to misclassify the adjacent characters into the same class as shown in Fig. 7a. What's more, some noises are misinterpreted as characters as shown in Fig. 7b. Thus, clustering correction is needed to optimize the results of character segmentation.

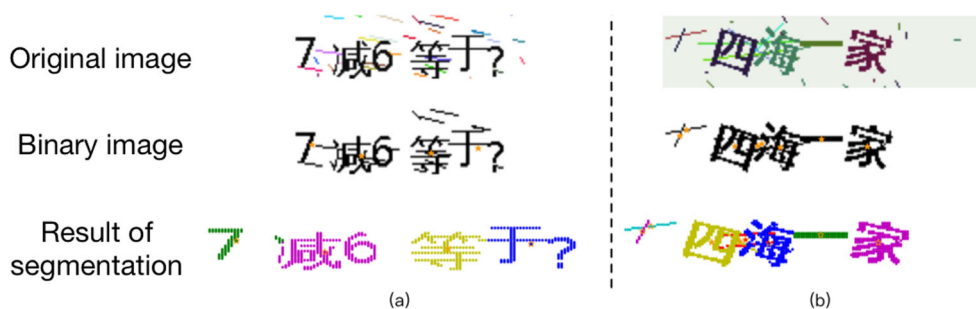
In clustering correction, if the number of pixels in a class is less than 10 or less than 5% of total pixels, the class is neglected. Furthermore, if the pixel spatial distribution is too long or too wide, the mean shift clustering is implemented for this class again. Examples of clustering correction is shown in Fig. 8.

3.2.4 Morphological open operation

Since the segmentation process does not take full account of the character connectivity, there are many specks and holes in the segmented Chinese characters, as shown in Fig. 9, which is not conducive to subsequent feature extraction and character recognition.

In order to solve this problem, a segmented character image is implemented with morphological open operation. The results are displayed in Fig. 10, which are more suitable for feature extraction and character recognition.

Fig. 7 Segmentation cases without clustering correction. **a** case of adjacent characters misclassified into one class; **b** case of noises misinterpreted as characters



3.3 Recognition

In this part, two kinds of character recognition methods are proposed. One is based on multi-scale Gabor filter and logistic regression, which is denoted as MGLCR (Multi-scale Gabor and Logistic regression based CAPTCHA Recognition). The other one is based on CNN, which is denoted as CCR (Convolutional neural network based CAPTCHA Recognition).

3.3.1 MGLCR

MGLCR relies on the multi-scale Gabor filter to extract features and classify characters with logistic regression. The texture features of different characters are quite different in the frequency domain, thus the frequency descriptor is used for the description of texture features. It is a set of Gabor filters with different scales, rotation directions and frequencies that is chosen to describe an image. By convoluting images with these Gabor filters, the frequency distribution of each point is calculated.

Before extracting character features, it is necessary to convert pixel value to the number within [0, 255]. The converted image is denoted as X_{char} . Feature extraction steps based on Gabor filters are described as follows:

- (1) Based on four direction parameters $\theta_1 = 0, \theta_2 = \frac{\pi}{4}, \theta_3 = \frac{\pi}{2}, \theta_4 = \frac{3\pi}{4}$, three scale parameters $\sigma_1 = 1, \sigma_2 = 2, \sigma_3 = 4$ and two frequency parameters $f_1 = 0.05, f_2 = 0.25$, 24 Gabor filters are generated as shown in Fig. 11, denoted as $Gabor_{\theta,\sigma,f}$.

- (2) The convolution results of four directions are obtained by convoluting a character image with four filters which have same scale and frequency parameters but different direction parameters. The multiplication result of four filtering responses is the proximate isotropic decomposition of the image at this scale.

$$C(x, y, \sigma, f) = \prod_{i=1}^4 Gabor_{\theta,\sigma,f} \times X_{char} \tag{9}$$

- (3) Multi-scale Gabor responses are calculated as follows:

$$C(x, y) = \sum_i^3 \sum_j^2 C(x, y, \sigma_i, f_j) \tag{10}$$

In order to make the eigenvectors of character images have same dimension, PCA (principal component analysis) is used to preserve the main components in the response results and the dimensionality is reduced to 128 dimensions.

The character recognition is a typical machine learning classification process. It is divided into two parts: training and testing. During the training phase, character features are extracted on the training set to train the logistic regression classifier. During the testing phase, character features are extracted on the test set and characters are classified with previously trained classifier.

The logistic regression is a multi-class classifier, which is equivalent to applying a logistic function to ordinary linear regression so that it can only get [0,1] range of values. Assuming that the output of the function is the probability that a data class belongs to $y = 1$. The closer the value is to

Fig. 8 Examples of clustering correction. **a** clustering correction for adjacent characters; **b** clustering correction for noises

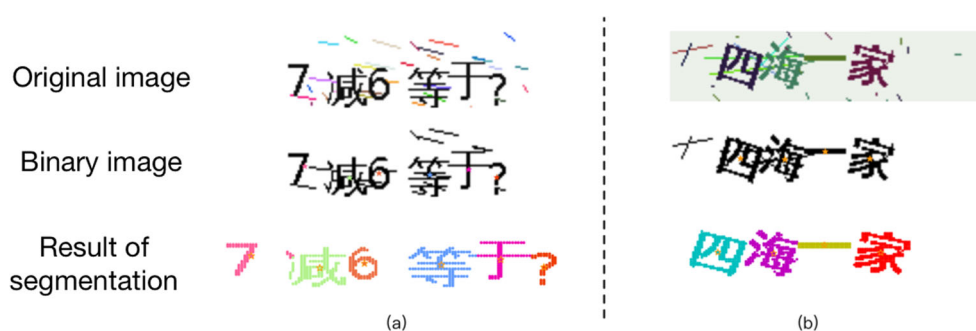


Fig. 9 Segmented characters with specks and holes



1, the more likely it belongs to $y = 1$, while the closer the value is to 0, the more likely it belongs to $y = 0$.

$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}} \quad (11)$$

where $h_{\theta}(\dots)$ is the logistic regression model, θ is the model parameter.

Since the character recognition is a multi-class task, multiple logistic regression is adopted, i.e., softmax regression. The loss function is defined as:

$$\mathcal{L}(X, Y, \Theta) = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^m 1(y_i = k) \log(\text{Pr}(\hat{y}_i = k | x_i, \Theta)) \quad (12)$$

where X is a batch of input character images, Y is the set of character labels corresponding to these images and Θ is the model parameter. N is the number of images, m is the total number of classes. $1(\dots)$ is a characteristic function and $1(y_i = k)$ means that if the i -th tag y_i belongs to the class k , the function value is 1, otherwise it is 0. $\text{Pr}(\hat{y}_i = k | x_i, \Theta)$ means that when inputting x_i , the estimated output \hat{y}_i is the probability of class k . This probability is obtained by the following formula:

$$\text{Pr}(\hat{y}_i = k | x_i, \Theta) = \frac{e^{\theta_k^T x}}{\sum_{k=0}^m e^{\theta_k^T x}} \quad (13)$$

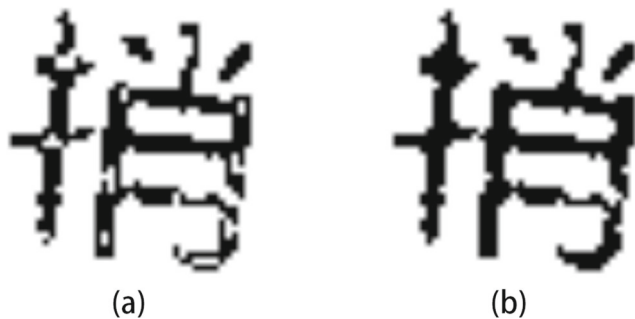


Fig. 10 Segmented characters after filling holes. **a** original segmented character; **b** segmented character after filling holes

With the multi-scale Gabor, features can be extracted from segmented character images. The following figure shows the convolution result of a Chinese character image with 24 Gabor filters.

Figure 12 reveals that the larger the scale parameter σ is, the fuzzier the features will be. The smaller the scale parameters σ is, the more detailed the features will be. The filters with different directions make the feature extraction more robust, so as to achieve rotation invariance.

Each response obtains a 128-dimensional vector after the dimension reduction. Then the vectors are joined. 24 Gabor filters perform feature extraction and obtain a 3072-dimension vector, which will describe the features of a character image.

The Gabor feature extraction is carried out on the character images of training set and the obtained feature vectors are used for training the logistic regression classifier, by which character images are predicted.

3.3.2 CCR

The proposed CNN is used to extract features and recognize character images automatically. Compared with the LeNet-5, the proposed network has additional structures such as batch normalization and dropout as shown in Fig. 13, which improves the performance of networks.

The marks, like $46 \times 46@64$, represent the size and number of feature maps. As shown in Fig. 13, the network architecture consists of five convolution layers, two max pooling layers and one fully connected layer. All convolution layers in the network are composed of 64 convolution filters with 3×3 size. Predicted results are outputted through the softmax classifier.

Each max pooling layer is followed by a batch normalization layer and a dropout layer. The batch normalization layer is used to normalize the middle layer characteristics of the network, which can significantly improve the convergence efficiency and the generalization ability of the network. Besides, the dropout layer randomly sets a part of

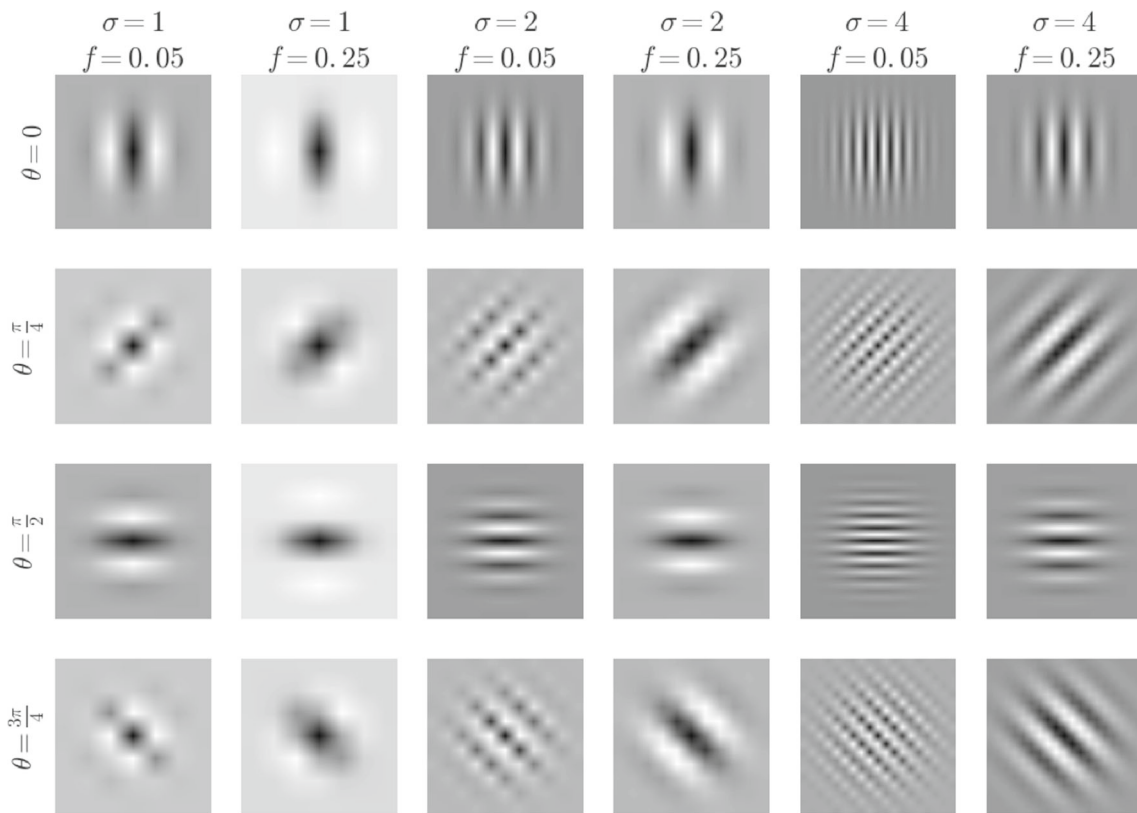


Fig. 11 Gabor filters

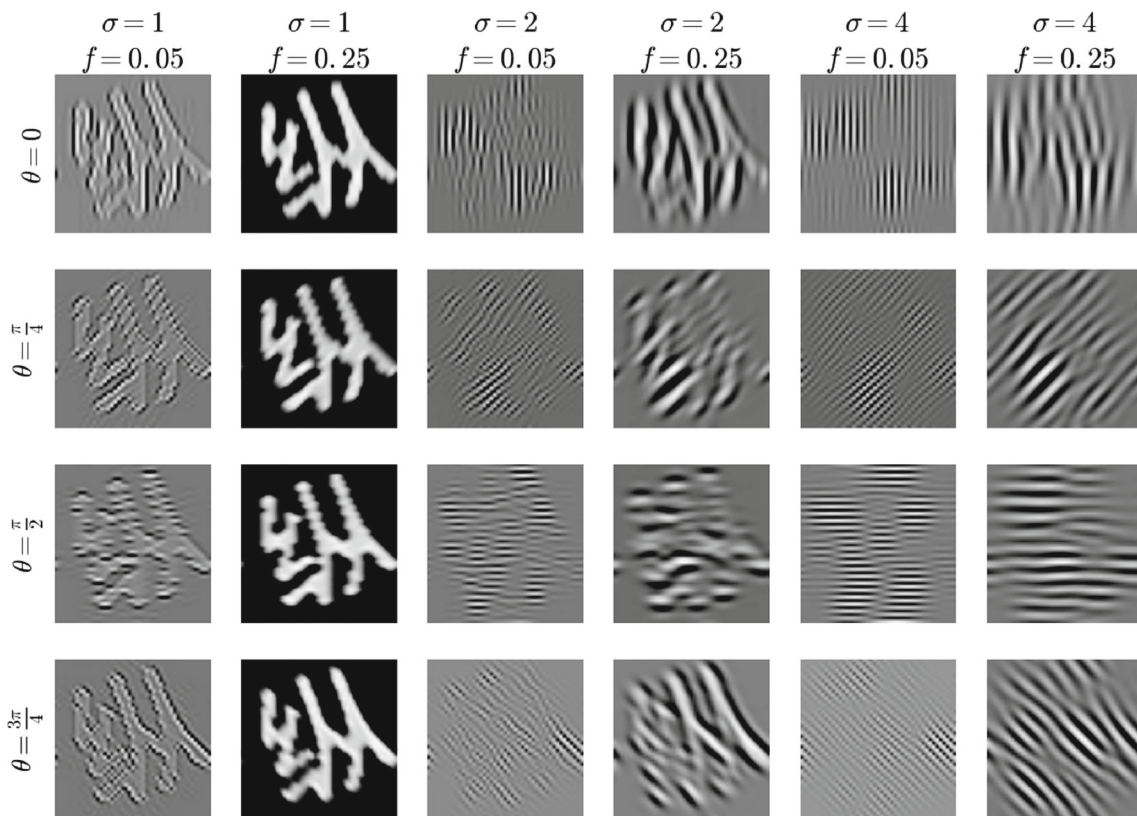


Fig. 12 Results of character images under multi-scale Gabor

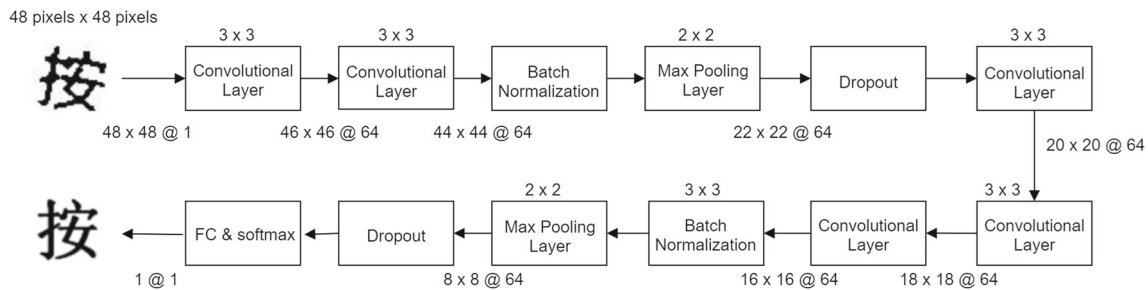


Fig. 13 Network architecture of the CCR method

the neuron's output value to zero in the training process, which can improve the generalization ability of the network and avoid over-fitting.

Additionally, except the fully connected layer, ReLU is adopted as the activation function in other layers, which can avoid gradient disappearance. At the same time, because of the fact that the output values are zeros of some neurons, ReLU function can avoid over-fitting. Besides, the ReLU function is simpler than the sigmoid function so as to alleviate the complexity of the network.

The categorical cross entropy between the outputs and the ground truth is calculated by the following formula.

$$\mathcal{L}(X, Y; \Theta) = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^m 1(y_i = k) \log(\text{Pr}(\hat{y}_i = k | x_i; \Theta)) \quad (14)$$

Obviously, the loss function is the same as the logistic regression described in MGLCR method. X is a batch of characters images, and Y is the set of character labels corresponding to these images and Θ is the network parameter. N is the number of images, and m is the total number of classes. $1(\bullet)$ is a characteristic function, in which $1(y_i = k)$ means that if the i -th tag y_i belongs to the class k , the function value is 1, otherwise it is 0. $\text{Pr}(\hat{y}_i = k | x_i; \Theta)$ means that when inputting x_i , the estimated output \hat{y}_i is the probability of class k . This probability is obtained by the following formula:

$$\text{Pr}(\hat{y}_i = k | x_i; \Theta) = \frac{\exp(o_i)}{\sum_{k=0}^m \exp(o_i)} \quad (15)$$

where o_i is the output of the last layer in the network.

During the training phase, character images are predicted by feed-forward propagation firstly. Then the loss between the estimated values and the ground truth is minimized by back-propagation. Until convergence, the trained network is able to recognize characters.

Automatic feature extraction and classification can be accomplished with the proposed CCR method. In the training procedure of CCR method, simple stochastic gradient

descent with learning rate of 0.01, weight decay of 0.00001 and momentum of 0.9 is applied for optimizing the training loss. The size of mini-batch in training is 64. After trained with around 800 epochs, the model was ready for predicting characters.

4 Experiments

4.1 Dataset

To prove the general efficiency of proposed methods, twenty subclasses of CAPTCHA images with different type of noises are tested, which are divided into four classes. Figure 14 lists all kinds of CAPTCHAs within four classes and twenty subclasses. Class One includes groups of English letters or Arabic digits or mixtures of English letters and Arabic digits. Although, the proposed method is for Chinese character CAPTCHAs, Class One is still included in the dataset to check the general validity of proposed methods. Class Two contains CAPTCHAs with variable-length Chinese characters. Class Three includes mathematical formulas consisting of Chinese characters only. Class four contains other types of mathematical formula which is the combination of Chinese characters, Arabic digits and mathematical operators. In view of the above four classes of CAPTCHAs, a CAPTCHA generator is developed to generate training sets with labels. The CAPTCHA generator has two sets of parameters: one set is the color, rotation angle, position, size and font of characters, and the other set is the background colors, density of straight lines and arcs and random layout of noises.

According to 2,000 common Chinese characters, 26 capital letters of English, 26 lowercase letters of English, 10 digits and 5 operational symbols, training sets of 60,000 CAPTCHAs are generated, which include 20,000 CAPTCHAs for Class One, 20,000 CAPTCHAs for Class Two, 10,000 CAPTCHAs for Class Three and 10,000 CAPTCHAs for Class Four. For each class, half of CAPTCHAs have no noises such as straight lines and arcs

Class One	
Class Two	
Class Three	
Class Four	

Fig. 14 CHAPTCHAs of four classes

and the other half have noises. Furthermore, background colors are randomly generated for all classes. The generated CAPTCHAs are displayed in Fig. 15.

Test set consists of 10,000 images from the website of Chinese National Enterprise Credit Information Publicity System, where are 2500 CAPTCHAs for each class.

4.2 Experimental results of preprocessing

With the proposed preprocessing method, experiments are carried out on the test set. Figure 16 depicts 10 successful

preprocessing cases, in which the left column lists original images and the right one lists denoised images after preprocessing. Experimental results show that the proposed preprocessing method can denoise CAPTCHA images efficiently with crossed straight lines, background textures and gradient background colors from Fig. 16a to Fig. 16j. What's more, proposed preprocessing method is self-adaptive to do local binarization on a CAPTCHA with gradient background as shown in Fig. 16d. Last but not the least the proposed preprocessing method could deal with circular noises as shown in Fig. 16a.

Fig. 15 Examples of synthetic CAPTCHA images

Class One		
Class Two		
Class Three		
Class Four		

Fig. 16 Successful cases of preprocessing



Unfortunately the proposed preprocessing method is not applicable with intersecting thick straight lines as shown in Fig. 17b whose original CAPTCHA is Fig. 17a. Especially, it is difficult to remove the straight lines with high-density layouts. It is thick straight lines that could not be filtered by the denoising method with morphological operators, which will probably cause difficulties in subsequent segmentations.

Besides thick straight lines with high-density layouts, the distribution of background color also has a side effect on the denoising results. If the background color is similar to the foreground color, it could lead to failure as shown in Fig. 17 (d) whose original CAPTCHA is Fig. 17c.

4.3 Experimental results of segmentation

The segmentation results are presented with the proposed self-adaptive mean shift. A good segmentation should follow two principles. On the one hand the amount of segmented characters should be correct. On the other hand, the segmented character can be recognized easily. Based

on above principles, 10,000 images for all and 500 images for each subclass (shown in Fig. 14) are tested. The segmentation accuracies are listed on Fig. 18.

According to the Fig. 18, the proposed segmentation method performed well in the CAPTCHAs with groups of English letters or Arabic digits, whose accuracy is about 100%. If the CAPTCHAs have gradient background colors, the proposed segmentation method can still achieve a high accuracy around 96.8%. As the background color is similar to the foreground character color, the segmentation accuracy decreases slightly to 90%. It is unfortunate that the accuracy of CAPTCHAs consisting of thick straight lines with high-density layouts is about 50% which is obviously lower than other subclasses. Figure 19 displays the segmentation results, which reveals that variable-length CAPTCHAs can be successfully segmented by the proposed method without the parameter of text length. Furthermore proposed segmentation method could also achieve good results in CAPTCHAs with colorful characters because it can distinguish the color space distribution in clustering.

Fig. 17 Failure cases of preprocessing

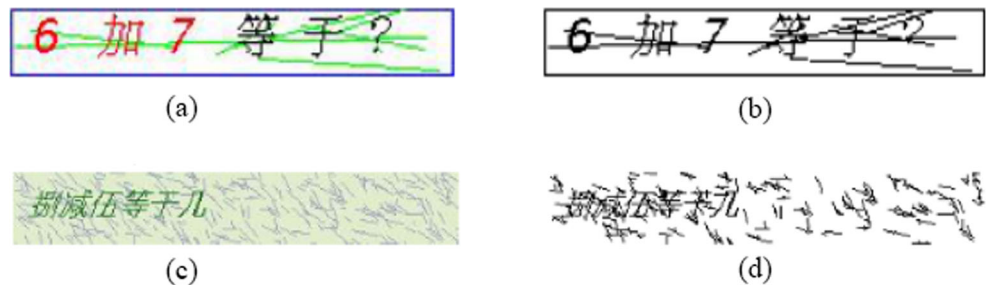


Fig. 18 Segmentation accuracies of all classes CAPTCHAs

	CAPTCHA in 20 subclasses	Accuracy		CAPTCHA in 20 subclasses	Accuracy
(a)		100.0%	(b)		100.0%
(c)		98.6%	(d)		95.3%
(e)		44.9%	(f)		100.0%
(g)		98.9%	(h)		94.6%
(i)		88.2%	(j)		45.6%
(k)		94.8%	(l)		78.2%
(m)		99.8%	(n)		87.5%
(o)		94.6%	(p)		92.2%
(q)		55.6%	(r)		88.4%
(s)		85.0%	(t)		41.6%

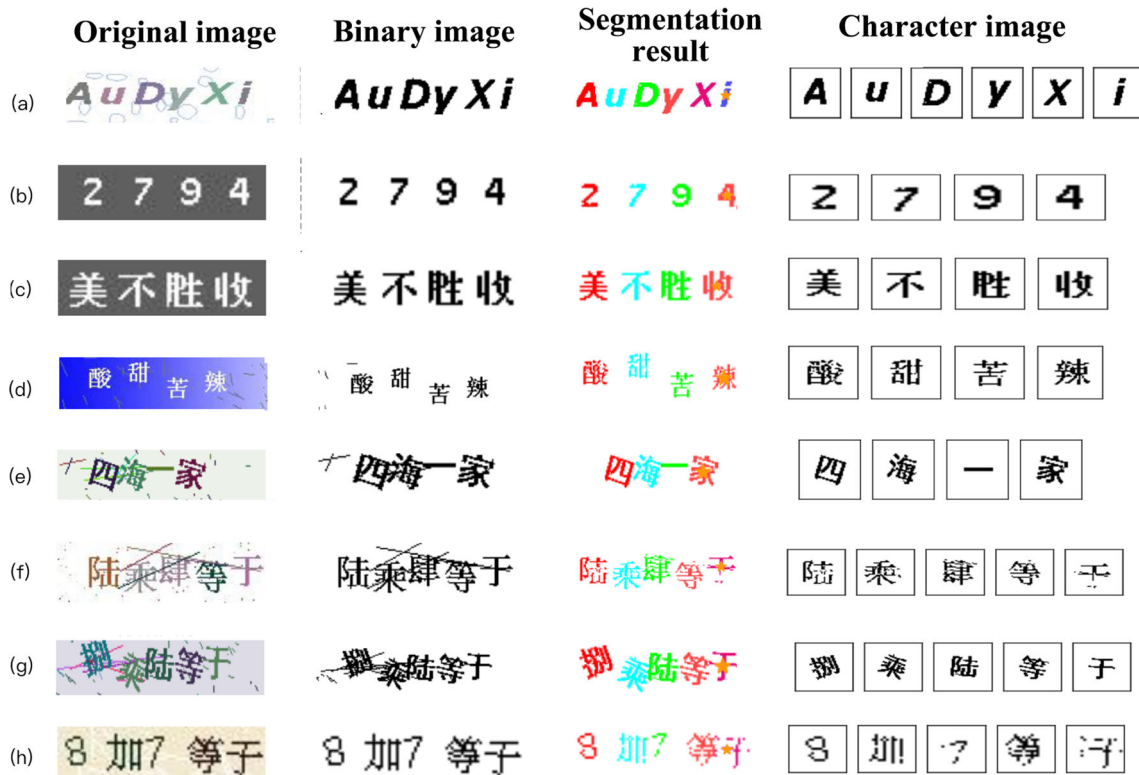


Fig. 19 Successful segmentation results on training set

4.4 Experimental results of recognition

After segmentation, 453,280 segmented character images from 60,000 CAPTCHAs are obtained. The ratio of training set and test set is 7:3. The recognition results of proposed MGLCR and CCR methods are compared by precision and recall rates. Traditionally, equations of precision and recall rate are defined as follows:

$$Precision = \frac{N_{tp}}{N_{tp} + N_{fp}} \quad (16)$$

$$Recall = \frac{N_{tp}}{N_{tp} + N_{fn}} \quad (17)$$

where N_{tp} is the number of true positive cases, N_{fp} is the number of false positive cases and N_{fn} is the number of false negative cases.

The above (16) and (17) concern binary classification whereas CAPTCHA classification is a multiclass problem. Thus the equations for precision and recall rate should be deformed as shown in (18) and (19).

$$Precision_i = \frac{M_{ii}}{\sum_{j=1}^n M_{ji}} \quad (18)$$

$$Recall_i = \frac{M_{ii}}{\sum_{j=1}^n M_{ij}} \quad (19)$$

where M_{ij} is the corresponding value in confusion matrix and $Precision_i$ and $Recall_i$ is the precision and recall of the i th class.

The recognition precisions and recall rates of MGLCR and CCR are compared in the Table 1, which lists the average values of all four classes. From the table, we can see that both MGLCR and CCR achieve good results on Class One, Class Three and Class Four. However the MGLCR only get 82.3% precision on Class Two with variable-length Chinese character CAPTCHAs, which is much lower than CCR. There are pros and cons of proposed MGLCR and CCR methods. MGLCR is less time-consuming and suits for the CAPTCHAs with small dataset of characters such as English letters and Arabic digits. Whereas CCR has relatively longer training time but it performs well

Table 1 Comparison between two recognition methods

		MGLCR	CCR
Class One	Precision	98.5%	99.4%
	Recall rate	98.4%	99.6%
Class Two	Precision	82.3%	92.6%
	Recall rate	81.0%	93.4%
Class Three	Precision	90.3%	93.7%
	Recall rate	92.5%	94.8%
Class Four	Precision	97.1%	97.4%
	Recall rate	97.8%	98.2%

in all cases no matter how large the dataset is and how complicated the CAPTCHA character structure is.

Besides making comparisons between MGLCR and CCR methods, state-of-the-art methods are also evaluated. For ease of presentation, the method breaking Microsoft CAPTCHAs with CNN is called M-CNN [18] and the CNN-based method cracking fixed-length Chinese character CAPTCHAs is called C-CNN [26]. The datasets used by [18, 26] and our proposed method are different. The segmentation in [18] was done assuming equal distance among the characters, so the image was divided in six equally distributed segments, which could not meet the challenge in our proposed scenarios. The segmentation method for the CHAPTCHAs in [26] is not included in their paper. To enable a fair comparison of previously mentioned three methods, the proposed Class-One and Class-Two CAPTCHAs are segmented with our proposed segmentation method into single characters and classified into two separate datasets for experiments.

On one hand, M-CNN applies LeNet-5 to break Microsoft CAPTCHAs which are groups of English letters or Arabic digits or mixtures of English letters and Arabic digits. Thus the dataset for the comparative experiments is composed of 20,000 letter-digit CAPTCHAs from Class One. The ratio of training set and test set is 7:3. The Table 2 lists the precisions and recall rates of M-CNN and the proposed MGLCR and CCR. The precisions and recall rates of the proposed MGLCR and CCR are higher than M-CNN. The reason could be that M-CNN uses a similar structure of LeNet-5 instead of a carefully designed network structure. Besides, the number of training epochs can also affect the results.

On the other hand, C-CNN can crack fixed-length Chinese character CAPTCHAs, which is similar to proposed CCR. Thus the dataset for the comparative experiments is composed of 10,000 Chinese character CAPTCHAs from Class Two and all the length of the CAPTCHA is four. The ratio of training set and test set is also 7:3. The comparative results of C-CNN and the proposed MGLCR and CCR are listed in Table 3. As can be seen from the table, the proposed CCR has higher precision than C-CNN and they achieve the similar recall rates. However the recognition results based on the good segmentation results with our proposed method, which is not included in [26]. Unfortunately MGLCR performs not so well in fixed-length Chinese character CAPTCHAs with intersecting thick straight

Table 2 Comparison on fixed-length letter-digit CAPTCHAs

	M-CNN	MGLCR	CCR
Precision	91.1%	98.8%	*99.6%
Recall rate	91.4%	98.7%	*99.7%

Table 3 Comparison on fixed-length Chinese character CAPTCHAs

	C-CNN	MGLCR	CCR
Precision	94.2%	83.2%	*94.6%
Recall rate	*95.3%	82.1%	95.2%

lines. Thus, the methods based on CNN are more recommended than the method based on multi-scale Gabor to crack Chinese character CAPTCHAs.

5 Conclusions

A new attempt is made on attacking variable-length Chinese character CAPTCHAs, which includes preprocessing, character segmentation and character recognition.

Although variable-length Chinese character CAPTCHAs can be cracked within acceptable time, there are some future works can be done to improve the performance.

- (1) Removing intersected thick straight lines more effectively.
- (2) Exploring a more effective segmentation method for sticky characters.
- (3) Improving the recognition precision on the CAPTCHAs with similar foreground color and background color.

Hints for better design of the text-based CAPTCHA include that increasing the density of straight lines and arcs and improving the similarity of the foreground color and the background color of CAPTCHAs. However the balance between user friendliness and the security of CAPTCHAs should be noticed. Furthermore other types CAPTCHAs, like interactive CAPTCHAs or sound-based CAPTCHAs are alternative options.

Acknowledgements This paper is supported by National Natural Science Foundation of China under grant No. 61303094, also supported by the Science and Technology Commission of Shanghai Municipality under grant No. 16111107800 and No. 16511102400, by Innovation Program of Shanghai Municipal Education Commission under grant No. 14YZ024 and by Shanghai Key Laboratory of Financial Information Technology.

References

1. Abdalla K, Kaya M (2017) An evaluation of different types of captcha: effectiveness, user-friendliness, and limitations International Journal of Scientific Research in Information Systems and Engineering (IJSRISE) 2(3)
2. Al-Fannah NM (2017) Using aesthetic judgements to distinguish between humans and computers. arXiv:1704.02972
3. Anand S, Mittal S, Tuzel O, Meer P (2014) Semi-supervised kernel mean shift clustering. *IEEE Trans Pattern Anal Mach Intell* 36(6):1201–1215
4. Arfan Jaffar M (2017) A dynamic fuzzy genetic algorithm for natural image segmentation using adaptive mean shift. *J Exp Theor Artif Intell* 29(1):149–156
5. Benchaou S, Nasri M, El Melhaoui O (2017) Features extraction for offline handwritten character recognition. In: Europe and MENA cooperation advances in information and communication technologies, Springer, pp 209–217
6. Burger W, Burge MJ (2016) Digital image processing: an algorithmic introduction using Java. Springer, Berlin
7. Bursztein E, Aigrain J, Moscicki A, Mitchell JC (2014) The end is nigh: generic solving of text-based captchas. In: WOOT
8. 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, ACM, pp 125–138
9. Chandavale AA, Sapkal A (2012) Security analysis of captcha. In: International conference on security in computer networks and distributed systems, Springer, pp 97–109
10. Cote M, Albu AB (2015) Robust texture classification by aggregating pixel-based lbp statistics. *IEEE Signal Process Lett* 22(11):2102–2106
11. Eswaran S, Ashok A, Krishnan RH (2017) Graphical passwords effects of tolerance password, image choice and otp login security. *International Journal of Research and Engineering* 4(1):31–34
12. Gao H, Cao F, Zhang P (2016) Annulus: a novel image-based captcha scheme. In: Region 10 conference (TENCON), 2016 IEEE, IEEE, pp 464–467
13. Gao H, Tang M, Liu Y, Zhang P, Liu X (2017) Research on the security of microsoft's two-layer captcha. *IEEE Trans Inf Forensics Secur* 12(7):1671–1685
14. Garg G, Pollett C (2016) Neural network captcha crackers. In: Future technologies conference (FTC), IEEE, pp 853–861
15. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
16. Hussain R, Gao H, Shaikh RA (2016) Segmentation of connected characters in text-based captchas for intelligent character recognition. *Multimedia Tools and Applications* 76(24):25547–25561
17. Jeong J, Yoon TS, Park JB (2017) Mean shift tracker combined with online learning-based detector and kalman filtering for real-time tracking. *Expert Syst Appl* 79:194–206
18. Karthik CP, Recasens RA (2015) Breaking microsoft's captcha. Tech. rep., Tech. Rep
19. Kaur R et al (2016) A non ocr approach for math captcha design based on boolean algebra using digital gates to enhance web security. In: International conference on Wireless communications, signal processing and networking (wiSPNET), IEEE, pp 862–866
20. Khan M, Shah T, Batool SI (2016) A new implementation of chaotic s-boxes in captcha. *SIViP* 10(2):293–300
21. Khan S, Hussain M, Aboalsamh H, Bebis G (2017) A comparison of different gabor feature extraction approaches for mass classification in mammography. *Multimedia Tools and Applications* 76(1):33–57
22. LeCun Y et al (2015) Lenet-5, convolutional neural networks. <http://yann.lecun.com/exdb/lenet>
23. Li K, Wu Y, Song S, sun Y, Wang J, Li Y (2017) A novel method for spacecraft electrical fault detection based on fcm clustering and wpsvm classification with pca feature extraction. *Proceedings of the Institution of Mechanical Engineers Part G: Journal of Aerospace Engineering* 231(1):98–108

24. Li P, Peng L, Wen J (2016) Rejecting character recognition errors using cnn based confidence estimation. *Chin J Electron* 25(3):520–526
25. Liu L, Lao S, Fieguth PW, Guo Y, Wang X, Pietikäinen M. (2016) Median robust extended local binary pattern for texture classification. *IEEE Trans Image Process* 25(3):1368–1381
26. Lv Y, Cai F, Lin D, Cao D (2016) Chinese character captcha recognition based on convolution neural network. In: 2016 IEEE congress on evolutionary computation (CEC), IEEE, pp 4854–4859
27. Najafi MH, Salehi ME (2016) A fast fault-tolerant architecture for sauvola local image thresholding algorithm using stochastic computing. *IEEE Trans Very Large Scale Integr VLSI Syst* 24(2):808–812
28. Nguyen KC, Nakagawa M (2016) Text-line and character segmentation for offline recognition of handwritten japanese text. *IEICE Technical Report* 115(517):53–58
29. Pastore J, Brun M, Bouchet A, Ballarin V (2017) Color morphological reconstruction as a segmentation tool for microscope cell images. In: VII Latin american congress on biomedical engineering CLAIB 2016, bucaramanga, santander, colombia, october 26th–28th, 2016, Springer, pp s312–315
30. Qiu Y, Ming D, Zhang X (2016) Object oriented land cover classification combining scale parameter preestimation and mean-shift segmentation. In: 2016 IEEE international geoscience and remote sensing symposium (IGARSS), IEEE, pp 6332–6335
31. Shi G, Yu W, Xiao Z (2016) A method for license plate recognition in vehicle based on gabor feature and synergetic neural network. *Journal of Hebei University(Natural Science Edition)* 36(2):210–217
32. Shi J, Wang Y, Xu D, Yan C, Chen T, He Y, Tang L, Nie M, Duan P, Yan D et al (2017) Terahertz imaging based on morphological reconstruction. *IEEE J Sel Top Quantum Electron* 23(4):1–7
33. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
34. Singh VP, Pal P (2014) Survey of different types of captcha. *International Journal of Computer Science and Information Technologies* 5(2):2242–2245
35. 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
36. Szabó K. Z., Jordan G, Petrik A, Horváth Á., Szabó C. (2017) Spatial analysis of ambient gamma dose equivalent rate data by means of digital image processing techniques. *J Environ Radioact* 166:309–320
37. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9
38. Thakur A, Chaware R, Nikhil S, Islam SH (2015) A reading oriented overlapping text based captcha. In: 2015 international conference on trends in automation, communications and computing technology (i-TACT-15), vol 1. IEEE, pp 1–6
39. Thangavelu S, Purusothaman T, Gowrison G (2017) Action based color captcha approach based on human cognitive factor in web applications. *Asian Journal of Research in Social Sciences and Humanities* 7(2):784–795
40. Torok L, Liatsis P, Viterbo J, Conci A et al (2017) k-ms. *Pattern Recogn* 66(C):392–403
41. Wang Y, Huang Y, Zheng W, Zhou Z, Liu D, Lu M (2017) Combining convolutional neural network and self-adaptive algorithm to defeat synthetic multi-digit text-based captcha. In: 2017 IEEE international conference on Industrial technology (ICIT), IEEE, pp 980–985
42. Xiao X, Yang Y, Ahmad T, Jin L, Chang T (2017) Design of a very compact cnn classifier for online handwritten chinese character recognition using dropout and global pooling. arXiv:1705.05207
43. Xu S, Li M, Zheng RR, Michael S (2017) Manchu character segmentation and recognition method. *J Discret Math Sci Cryptogr* 20(1):43–53
44. Yan J (2016) A simple generic attack on text captchas
45. Yin L, Yin D, Zhang R, Wang D (2014) A recognition method for distorted and merged text-based captcha. *Pattern Recognit Artif Intell* 27(3):235–241
46. Zhang Y, Yan H, Zou X, Tao F, Zhang L (2016) Image threshold processing based on simulated annealing and otsu method. In: Proceedings of the 2015 Chinese intelligent systems conference, Springer, pp 223–231



Xing Wu received his PhD in Computer Science at the Shanghai JiaoTong University. He is an associate professor at the School of Computer Engineering and Science, Shanghai University. His research interests include issues related to artificial intelligence and data mining. He is the author of a great deal of research studies published at national and international journals, conference proceedings as well as book chapters.



Shuji Dai received her master degree in School of Computer Engineering and Science, Shanghai University. Her current research interests include data mining and deep learning. She has published some research papers on international journals and conferences.



Yike Guo is a professor of Computing Science in the Department of Computing at Imperial College London. He is the founding Director of the Data Science Institute at Imperial College. Professor Guo also holds the position of dean of the School of Computer Engineering and Science, Shanghai University. Professor Guo has published over 200 articles, papers and reports.



Hamido Fujita is the professor and director of Intelligent Software Systems Laboratory at Iwate Prefectural University. He is researching on designing new software techniques based on cognitive interaction between human and machines, and intelligent decision making based on Medical Knowledge for medical decision making application. He is the author of a great deal of research studies published at national and international journals, conference proceedings as well as book chapters.