Pattern Recognition 42 (2009) 3365 – 3373

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Synthetic handwritten CAPTCHAs

Achint Oommen Thomas*, Amalia Rusu, Venu Govindaraju

CUBS, CEDAR, State University of New York at Buffalo, 201 Bell Hall, Amherst, NY 14280, USA

ARTICLE INFO

Article history:
Received 9 August 2008
Received in revised form 30 November 2008
Accepted 19 December 2008

Keywords:
HIP
CAPTCHA
Handwriting generation

ABSTRACT

CAPTCHAs (completely automated public Turing test to tell computers and humans apart) are in common use today as a method for performing automated human verification online. The most popular type of CAPTCHA is the text recognition variety. However, many of the existing printed text CAPTCHAs have been broken by web-bots and are hence vulnerable to attack. We present an approach to use human-like handwriting for designing CAPTCHAs. A synthetic handwriting generation method is presented, where the generated textlines need to be as close as possible to human handwriting without being writer-specific. Such handwritten CAPTCHAs exploit the differential in handwriting reading proficiency between humans and machines. Test results show that when the generated textlines are further obfuscated with a set of deformations, machine recognition rates decrease considerably, compared to prior work, while human recognition rates remain the same.

1. Introduction

In 1950, Alan Turing described a test to distinguish humans from machines. This test, known as the Turing test [1], was designed to be administered by a human who would ask questions, to a human and a machine trying to pose as a human, and try to ascertain which is which by the answers received. In a reverse Turing test, a machine asks the questions with the aim of distinguishing between humans and machines. The CAPTCHA (completely automated public Turing test to tell computers and humans apart) [2] is an example of a reverse Turing test. Today, there are a number of online services which allow people to contribute content and interact online in some manner. Many of these services require that they be accessed only by humans. CAPTCHAs have come into the spotlight in cyber security applications for use in automated human verification online. Spam control for blogs and automated account sign-up by bots are some of the applications that require testing if the entity accessing a service is a human or an automated machine. There is economic incentive in posing as a human online. Consider a website like Ticketmaster.com which sells event tickets online. The website allows only a limited number of tickets to be bought using one account, to prevent activities like ticket hoarding which leads to ticket price inflation. A hacker could write a web-bot to sign up for hundreds of accounts and buy out a large number of tickets. These tickets could later be sold at a premium price, illegally. In such situations, it is imperative to know whether the entity creating the account is human or machine.

Most CAPTCHAs in use today are text-based. An image consisting of a series of printed text characters are rendered, distorted and obfuscated to varying degrees. The distorted image is then presented to a user. If the user correctly guesses the characters present in the CAPTCHA in the right order, he/she is granted access to some service. Circumventing the challenge posed by a CAPTCHA is an area that malicious hackers are actively looking into. Several printed text based CAPTCHAs have already been broken as reported in [3].

2. Background

Researchers are working on techniques to allow for automatically distinguishing between humans and machines. The general area is termed as human interactive proofs (HIPs) of which CAPTCHAs are a type of HIP. A number of different genres of CAPTCHAs exist; visual, auditory and semantic. The most commonly used genre is the visual CAPTCHA. Under the visual CAPTCHA, various types are present. http://www.captcha.net/cgi-bin/esp-pix [4] asks users to mark all images out of a set of images that contain similar objects. http://www.toallwhoseekit.net/cgi-bin/sq-pix [5] asks users to mark the region in an image containing some object where the object name is presented on screen to the user. However, the most popular kind of visual CAPTCHA is the text CAPTCHA. Text CAPTCHAs are popular since automatic recognition of degraded, noisy, distorted text with background clutter is still a challenging task for machines, but is a task that humans perform with relatively more ease. Many of the text CAPTCHAs proposed in the literature exploit this shortcoming.

* Corresponding author. Tel.: +1 716 472 6899; fax: +1 716 645 2377.
E-mail address: aothomas@buffalo.edu (A.O. Thomas).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.12.018

PessimalPrint [6], whose designers had extensive experience with OCR technology and understood its limitations takes advantage of known weaknesses of OCR technology. Other CAPTCHAs [7–9] are based on heuristics from Gestalt theory. ScatterType [7] uses the Gestalt principle of proximity. Fragments of the message are scattered within a short distance of each other; the relatively small distance allows the human perceptual system to connect the pieces so humans can read the message. Machines still find this to be a big challenge. For more background on CAPTCHAs, the reader can refer [10].

Various models of human-like writing generation are available in the literature [11–17]. Many of the existing approaches are on-line based since it is convenient to change the trajectory and shape of the letters based on the on-line information such as pen-down, pen-up, and velocity profiles. Lin and Wan [11] describe an approach to synthesize handwriting in a user's style. They collect handwritten character samples using a GUI interface and then build the textlines in a bottom-up fashion. Their technique adapts to a user's specific style. Wang et al. [12] presented a learning based approach to synthesizing cursive handwriting of a user by combining shape and physical models. A delta-log normal model based conditional sampling algorithm was used to produce the handwriting. Guerfali and Plamondon [13] describe the delta-log normal model which has been used for the generation and modelling of rapid movements to generate curvilinear strokes. An optimal selection of parameters for generation of these strokes will result in appropriate symbols being generated so as to conform to different characters in the alphabet. Kokula [14] performs script font ligature generation on-the-fly by optimizing a parametric curve between two characters. Researchers are also applying various character and image level perturbations directly on real character images in cases where online character information is absent. In [15,16], a perturbation model for generating synthetic textlines from existing cursively handwritten textlines by humans is presented. The perturbation model uses a continuous nonlinear function to geometrically transform points along the original textlines based on the value of the function. Mori et al. [17] developed a character generation method based on point correspondence between patterns. Their method automatically generates new character samples that appear to be written naturally.

3. Motivation

Automated recognition of high resolution printed text is all but a solved problem [18]. However, automated recognition of unconstrained handwriting continues to be a challenging research task. This fact can be exploited to develop human verification systems for cyber security applications. By replacing the printed text content in today's text CAPTCHAs with handwritten content, it would intuitively appear that the recognition task would be made more difficult for machines. To be suitable for online applications, a machine must be able to generate a challenge as well as score the response to it. Also, it must be possible to automatically generate infinitely many distinct artificially handwritten samples. To our knowledge, the only other use of handwritten CAPTCHAs is in the work done in [8,19]. However, in that work, the challenge images used were city names segmented out from postal envelopes. The approach of obtaining handwritten word images by segmenting out word images from handwritten documents (postal envelopes, handwritten letters, etc.) will seriously limit the size of the dataset from which challenges can be generated. It will also not include random strings of characters or combinations of phonemes. A finite dataset is a flaw while designing CAPTCHAs, as an adversary with access to the dataset can use dictionary/lexicon based brute force attacks to circumvent the system. The other approach would be to build textlines of handwritten words on-the-fly. This makes it possible to generate infinitely many distinct challenges limited only by the length of the textline. We have thus chosen to build a handwriting generator and base our generation technique on pre-existing character images taken from varied sources to reduce dependency on specific writer styles.

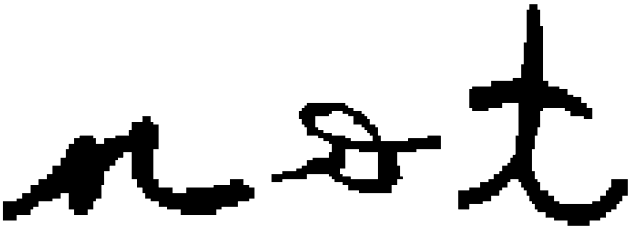

Fig. 1. Sample characters from the character dataset.

4. Generation method

In this section, we describe a method for the generation of cursive English handwritten textline samples that uses pre-existing character images. The generation algorithm consists of several steps: (i) character auto-scaling, (ii) automatic baseline determination, (iii) ligature endpoint detection, (iv) ligature parameterization, (v) ligature joining, (vi) skeleton perturbation, and (vii) skeleton thickening.

The high level algorithm to achieve this task can be described as follows. We first construct a preliminary image which is a concatenation of individual character templates/representations. Character templates are one pixel wide representations (skeletons) of the original character image. The preliminary image contains individual character templates strung together to form a string. Since the textlines have to be close to human handwriting style, important aspects like character baseline alignment, scaling of character sizes and ligature joins have to be considered. We present original techniques for automatic baseline detection and ligature handling. We also present a character auto-scaling technique. Once we have the preliminary image, we apply a set of geometric, image level perturbations that distorts the preliminary image in a random fashion. The perturbations can be parameterized and this allows us to pick random values over a range of values. The technique follows the model presented in [15]. Finally, the distorted image is thickened. Thickening can be controlled so that different parts of the image are thickened by different amounts.

A dataset of over 20,000 character images[1] which contains multiple handwritten samples of each English character has been used. The characters have been segmented out manually from actual pieces of US mail. For a large fraction of the cases, the beginning and ending ligatures are also present with the character. Fig. 1 shows some examples of the character images.

We first construct a preliminary image, which is a concatenation of individual character templates. Character templates are one-pixel wide representations (skeletons) of the original character image. We use Blum's medial axis transform [20] to generate the character templates.

4.1. Character auto-scaling

To form the preliminary image, we need to concatenate individual characters to form the required textline. We perform auto-scaling of the characters so that all characters maintain their correct relative sizes with respect to each other. This means that, for instance, a 'p'

[1] Dataset collection and character segmentation done by CEDAR, University at Buffalo, NY, USA.