Handwritten CAPTCHA: Using the difference in the abilities of humans and machines in reading handwritten words

Amalia Rusu and Venu Govindaraju
Center of Excellence for Document Analysis and Recognition (CEDAR)
Department of Computer Science and Engineering, University at Buffalo
Buffalo, NY, USA
{air2,govind}@cedar.buffalo.edu

Abstract

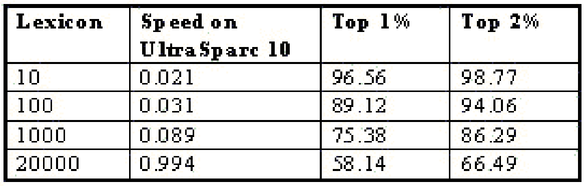
Handwritten text offers challenges that are rarely encountered in machine-printed text. In addition, most problems faced in reading machine-printed text (e.g., character recognition, word segmentation, letter segmentation, etc.) are more severe, in handwritten text. In this paper we present the application of Human Interactive Proofs (HIP), which is a relatively new research area with the primary focus of defending online services against abusive attacks. It uses a set of security protocols based on automatic tests that humans can pass but the state-of-the-art computer programs cannot. This is accomplished by exploiting the differential in the proficiency between humans and computers in reading handwritten word images.

Keywords: *Human Interactive proof (HIP), CAPTCHA, Handwriting Recognition, Word Recognition, OCR, Web security, Turing tests, SPAM, Challenge Response Protocol*

1. Introduction

Interpreting handwritten text is a task humans usually perform easily and reliably. However, automating the process is difficult, because interpreting text involves both recognizing symbols and comprehending the message conveyed. Although the progress in OCR accuracy is growing fast for various applications their accuracy is still inferior to that of a first grade child [18]. People can recognize the character components of written language in all shapes and sizes by the time they are five years old. They can recognize characters that are small or large, rotated, handwritten or machine printed. We present an application called CAPTCHA, which will exploit this differential in the reading proficiency between humans and computers when dealing with handwritten word images. A review of the handwriting recognition literature shows several algorithmic approaches have been explored, such as lexicon driven and lexicon free, parallel classifiers and combinations, pre and post processing routines, analytical and holistic methods [4, 9, 13, 14, 17, 20, 22]. Although some of the computer algorithms demonstrate human like fluency, they fail miserably when the images are degenerated or poorly written.

Our goal is to introduce ”Handwritten CAPTCHAs” as an automated recognition test that is designed to allow humans to pass with little effort but where the state-of-the-art computer programs ([28], Table 1) fail. CAPTCHA - *Completely Automatic Public Turing test to tell Computers and Humans Apart* belongs to the set of protocols called HIP (Human Interactive Proofs) which allow a person to authenticate as belonging to a select group, for example human as opposed to machine, adult as opposed to a child, etc. HIPs operate over a network without the burden of passwords, biometrics, special mechanical aids, or special training [3]. Since CAPTCHAs exploit the areas where computers are not as good as humans (yet), handwriting recognition is a candidate for these tests.



Lexicon	Speed on UltraSparc 10	Top 1%	Top 2%
10	0.021	96.56	98.77
100	0.031	89.12	94.06
1000	0.089	75.38	86.29
20000	0.994	58.14	66.49

Table 1. Speed and accuracy of a lexicon derive handwritten word recognizer [28].

Handwriting Recognition has been successfully used in several applications such as postal address interpretation [23], bank check reading [10], and forms reading [15]. These applications are all characterized by small or fixed

0-7695-2187-8/04 $20.00 © 2004 **IEEE**

lexicons afforded by contextual knowledge. Recognition of unconstrained handwriting is difficult because of the diversity in writing styles, inconsistent spacing between words and lines, and uncertainty of the number of lines in a page as well as the number of words in a line [21]. Also, most current handwritten word recognition approaches depend on the availability of a lexicon of words for matching, making the recognition accuracy dependent upon the size of the lexicon. So, for a truly general application-independent word recognizer, the lexicon would be the entire English dictionary and the accuracy of recognition would be very low.

It must be noted that without the context of a lexicon, unconstrained cursive handwriting recognition (offline) is extremely difficult for current recognition algorithms. Furthermore, the recognition accuracy drops dramatically with an increase in the lexicon size. The results in Table 1 are based on fairly well-written clean images extracted from US mail piece images. Thus, generating challenging handwritten word images where humans can read effortlessly and programs fail should be possible. One obvious way would be to increase the lexicon size (word choice list). However, this may not be always practical, as it would be difficult to present a user with a very large lexicon in a challenge-response test as it would take up most of the computer screen and also become burdensome for genuine human users. In this paper, we will describe alterative ways of ”transforming” the image to make it almost impossible for programs to read the handwritten words while the task still remains effortless for humans.

Figure 1 illustrates some of the word images that are difficult for any present day computer recognition techniques even when presented with a small list of words as a lexicon. We will use the handwritten word recognizers we have at our disposal ([8, 11, 28]) to conduct experiments and come up with parameters that can be used for automatic generation and distortion of handwritten word images.

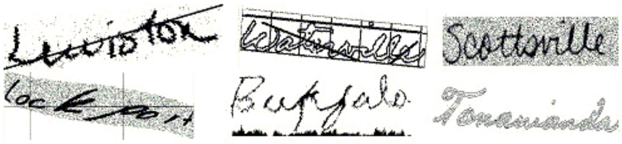


Figure 1. Examples of Handwritten CAPTCHA images where OCR systems fail.

2. Motivation

Internet spam is defined as ”unsolicited commercial bulk e-mail”, or junk mail, in other words advertisements that marketers blindly send to as many addresses as possible. It is widely accepted that the spam problem and the so-called ”Bots” have become a nuisance and must be defended against. Whereas individual anti-spam preventive measures and email address filtering may be used as a short-term solution, there is a need for more comprehensive solutions such as HIPs and CAPTCHAs.

Alta Vista web site was among the first to use CAPTCHA to block the abusive automatic submission of URLs [1]. Advanced efforts on HIPs have been made at the CMU [5, 25]. They have introduced the notion of CAPTCHA and defined its mandatory properties. Several CAPTCHA systems (e.g. Gimpy, Bongo, Pix) are available to readers on their web site [2]. Over the past three years, PARC and UC Berkeley have introduced new challenges [3, 6, 7, 16]. Mandatory Human Participation (MHP) is another kind of authentication scheme that uses a character-morphing algorithm to generate the character recognition puzzles [26]. All the CAPTCHAs currently in commercial use take advantage of superior human ability in reading machine printed text. Other algorithms use speech, facial features, and graphical Turing tests [12, 19]. To the best of our knowledge, this paper describes the first effort in ”Handwritten CAPTCHAs”.

There are four steps to authentication (Figure 2): (i) *Initialization*: the user expresses an interest in being authenticated by the server, (ii) *CAPTCHA Challenge*: the server generates a challenge in the form of a handwritten word image and issues it to the server, (iii) *User Response*: the user has to key in the right answer and return it to the server, (iv) *Verification*: the server verifies the user response and checks if it matches the right answer. It either grants access to the user or rejects the transaction.

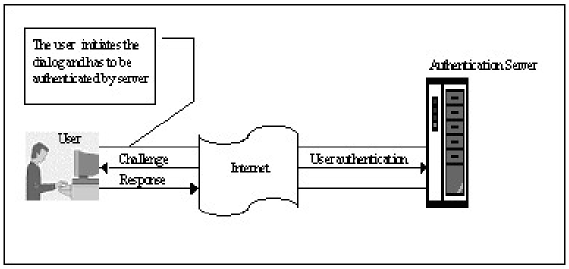


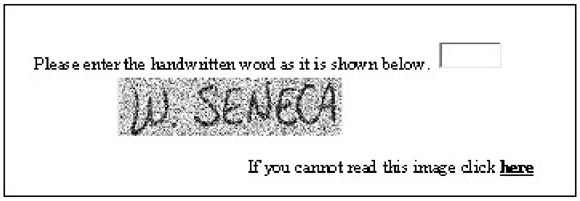
Figure 2. Automatic Authentication Session for Web Services.

In 1950, Alan Turing raised the question: ”Can machines think?”. The Imitation Game has been modified many times since then so that it became simply a problem of deciding whether the contestant is human or machine [24]. Recently, new formulations of the reverse problem with the following specifications exist: (i) the judge is a machine instead of a human, (ii) the goal is that virtually all human users will be recognized and pass the test, whereas no computer program can pass.

3 Technical Approach

Our focus is on automatic generation of CAPTCHA challenges (Figure 3). We have conducted experiments to investigate human recognition of a set of distorted hand printed image samples to gain an insight into human reading abilities. Holistic features [14] were first investigated since they are widely believed to be inspired by psychological studies of human reading.

For automatically generating CAPTCHA images transformations are applied to a randomly chosen handwritten word image from a database of over 4,000 handwritten US city name images. Alternatively, we can construct any handwritten word image by gluing together characters randomly chosen from a set of 20,000 handwritten character images of isolated upper and lower case alphabet. We have identified the following transformations that defeat current handwriting recognition systems. We essentially considered all the normalization operations that word recognizers use prior to recognition and simply introduced them on purpose. Also, given our knowledge of how much of the distortions a word recognizer can tolerate, we are able to generate images that cannot be easily normalized or rendered noise free by present computer programs.



Please enter the handwritten word as it is shown below.

If you cannot read this image click **here**

Figure 3. Example of interface and handwritten CAPTCHA to confirm registration.

1. *Noise*: Add lines, grids, arcs, circles, and other ”background noise”; Use random convolution masks, and special filters (i.e. multiplicative/impulsive noise, blur, spread, wave, median filter, etc)

2. *Segmentation*: Delete ligatures or use letters and digits touching with some overlap to make segmentation difficult; Use stroke thickening to merge characters

3. *Lexicon*: Use lexicons with ”similar” entries, large lexicons, or no lexicons; Use words with confusing and complex characters such as ”w” and ”m”

4. *Normalization*: Create images with variable stroke width, slope, and rotations; Randomly stretch or compress portions of word image

Since the process of image generation must be completely automated, and images and the associated set of planned distortions must be chosen at random, there is little risk of repetition. Also, the norms of CAPTCHA generation dictate that the method of generating these images must be public knowledge giving those that want to ”break” the CAPTCHAs a fair shot.

For deformation, we use the following algorithm:

Input. Original (random) handwritten image

Output. Deformed handwritten image

Method. Given a test image, run the deformation algorithm once.

1. Randomly choose the number of transformations (up to three).

2. Randomly establish the transformations corresponding to the given number. Some rules apply: (i) no transformation can be applied more than once to the same image (multiple times, it drastically degenerates the image and affects human reading abilities), (ii) if just one transformation was chosen then add one more, unless the first transformation is to add background noise.

3. A priori order is assigned to each transformation. Sort the list of chosen transformations based on their prior order. We have ordered them based on our experimental results and common sense. For example, applying noise to an image and then blurring or spreading it has undesired effect on word readability rather than doing it the other way round. In the second case the image preserves some of the original features and the word consistency would not be altered by meshing letters with backgrounds as in the first case. We found this ordering to be helpful for humans, but still difficult for recognizers.

4. Apply each transformation in sequence and generate the output-deformed image. After each transformation the image is updated, so that the effect is cumulative.

We examined the sources of errors of recognition algorithms and found that segmentation errors (i.e. over-segmentation, inability to segment), recognition errors (i.e. confusing with a similar entry in lexicon), and image quality are the most common. Typical cases of failure are images with background noise. We have found this to be the most powerful transformation in our algorithm because it is easily reproducible and the accuracy of the system drops on noisy images. On the other hand, the extra components such as arcs, lines, grids, etc. produce incorrect segmentation and recognition errors, thus significantly reducing the performance of recognizers. The other transformations that we considered (blur, spread, wave, median filter, etc) are efficient when combined in groups as evidenced by our experimental results.

4 Testing and Evaluation

We have used image files in TIFF and HIPS formats. We have generated the test images to be recognized by the two most advanced word recognizers available at CEDAR, Word Model Recognizer (WMR) [11] and HMM Recognizer [8]. Both recognizers match a word image against a lexicon (Figure 4). For each image, we have produced a deformed version by applying successive transformations using the above described algorithm. For every image the corresponding truth word is always present in the lexicon, and the lexicon is created so as to contain all the truths of test images.

We have completed the following two experiments.

1. First involves a database of 4,127 city name images and randomly generated distorted images based on the existing deformation algorithm. They are all handwritten city-words (cursive, hand printed, or the others) which are manually extracted from mail pieces. In general, each image contains only one word that corresponds to a US city name (Figure 5).

2. Second uses 3,000 word images generated based on a random combination of characters, with one word per image and random word length between 5 and 10 (Figure 6). The characters are chosen at random from a database of 20,529 characters, which were previously extracted from city name images (Figure 7). In addition to samples in Figure 1 and Figure 3 for which both recognizers failed to recognize the correct word, there are other examples recorded for further review. Majority of these handwritten images are readable by humans.

We have implemented an automated version of the algorithm and a number of transformations (up to three) are applied to each image. We performed tests by running WMR and HMM recognizers on more than 7,000 word images, most of them US postal word images of unconstrained writing styles. The corresponding error rates are shown in Table 2.

A pertinent observation for the computer program recognition results is that the lexicon plays a central role in determining the right word in most of the cases. In handwritten word recognition as described by most researchers in the literature, the performance of a recognizer depends on the quality of input image as well as other factors, such as the lexicon entries and lexicon size. It is intuitively understood that word recognition with larger lexicons is more difficult [8, 11]. Another accurate measure to categorize the difficulty of a word recognizer task is the similarity between lexicon entries, defined as the distance between handwritten words, and called ”Lexicon Density” [27, 29]. One of the co-authors has previously shown how the performance of a recognizer is a function of lexicon density and the results of performance prediction models change as lexicons density changes [27, 29]. Therefore, for CAPTCHA purposes, the design can involve introducing difficulty at the word image level or at the associated lexicon level. In order to utilize the lexicon level challenge, we have considered a few images that were successfully recognized by the two word recognizers in the previous test. In that instance, a lexicon of size 10 was chosen randomly. In Figure 8 we show what happens when the lexicon of size 10 is deliberately chosen to increase the confusion (density).

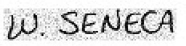
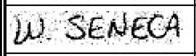
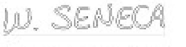
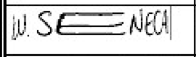
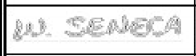
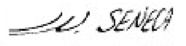
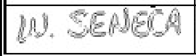
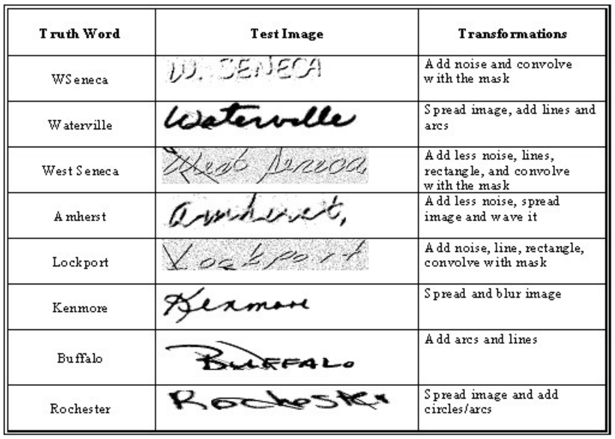
Image	Transformation	WMR	HMM
	Add noise	Recognizes	Fails
	Add noise, apply median filter	Recognizes	Fails
	Run chain code, empty letters	Fails	Fails
	Linear transformation	Recognizes	Recognizes
	Edge detector multiple times	Fails	Fails
	Affine transformation	Fails	Recognizes
	Empty letters, edge detector	Fails	Fails

Figure 4. Testing results for various transformations applied to the same test image.

Test Images	Number of test images	WMR Error Rate	HMM Error Rate
City Names	4,127	91%	96%
Random Nonsense Words	3,000	88%	97%

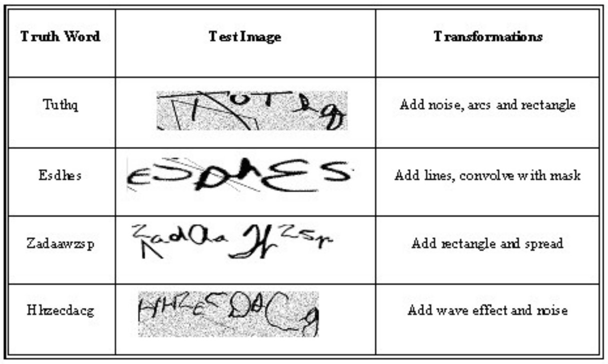
Table 2. Error rates of handwriting recognizers.

Although currently in its manual version, the idea of generating random lexicons with higher density is expected to provide additional handwritten CAPTCHAs, this phase has not been completely researched. For instance, we would



Truth Word	Test Image	Transformations
WSeneca		Add noise and convolve with the mask
Waterville		Spread image, add lines and arcs
West Seneca		Add less noise, lines, rectangle, and convolve with the mask
Amherst		Add less noise, spread image and wave it
Lockport		Add noise, line, rectangle, convolve with mask
Kenmore		Spread and blur image
Buffalo		Add arcs and lines
Rochester		Spread image and add circles/arcs

Figure 5. City name images that defeat both WMR and HMM recognizers.



Truth Word	Test Image	Transformations
Tuthq		Add noise, arcs and rectangle
Esdhes		Add lines, convolve with mask
Zadaawzsp		Add rectangle and spread
Hlzecdacg		Add wave effect and noise

Figure 6. Random ”nonsense” word images that defeat both WMR and HMM recognizers.

like all the lexicon entries to be ”real” words in order to make it a fair comparison with human ability which relies heavily on context. In fact, when the writing is unconstrained or very loosely constrained, comprehension cannot follow recognition in a strictly sequential manner, because comprehension must occasionally assist recognition. In these cases we need the comprehension process to help disambiguate uncertainty caused by variability in the input patterns.

We have administered tests to 12 graduate students in our department. The test consists of 15 handwritten images as follows: 10 handwritten word images of city names, and 5 random ”nonsense” words formed by concatenating handwritten characters. The images are chosen from the set of word images that defeat both recognizers. As expected, the first set of city names did not pose any problem for the humans given the context. For the ”nonsense” words, students faced some difficulties and confused the following characters: ”g” vs. ”q”, ”r” vs. ”n”, and ”e” vs. ”c”. Clearly, the overall error rate of 18% for humans versus 100% for both recognizers shows the superiority of human abilities when reading handwritten text images.

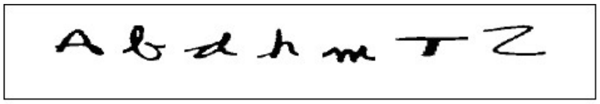
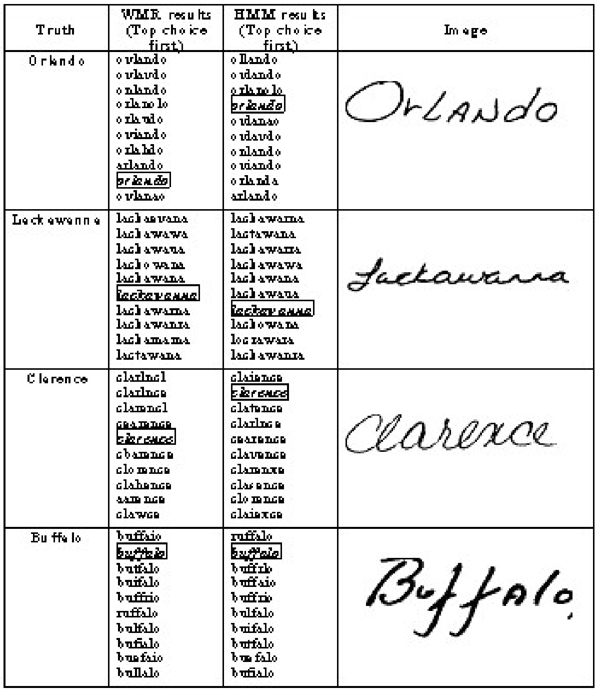


Figure 7. Examples of handwritten characters used to generate random ”nonsense” words.



Truth	WMR results (Top choice first)	HMM results (Top choice first)	Image
Orlando	ovlando ovlavdo onlando orlano lo orlavdo oviando orlaldo arlando orlando ovlanao	ollando ovdando orlano lo orlando ovdanao ovdavdo onlando oviando orlanda arlando	
Lackawanna	lachaevana lachawawa lachawana lacho wana lachawana lackawanna lachawama lachawanra lachamama lactawana	lachawama lactawana lachawara lachawawa lachawana lachawana lackawanna lachowana locrawara lachawanra	
Clarence	clarlncl clarlnce claranel cearance clarence cbarance clo mnce clahence aam nce clawce	claiance clarence clatence clarlnc e cearance clavence claranxe clavence clo mnce claiexce	
Buffalo	buffaio buffalo buttalo buifalo buffrio ruffalo bulfalo bufialo bus faio bullalo	ruffalo buffalo buffrb buffaio buffrio bulfalo buifalo butfalo bus falo bufialo	

Figure 8. Handwritten CAPTCHAs using lexicons with similar entries. The images are not deformed here just to show the effect of this method without image transformations. Even in this situation, the recognizers did not produce the correct results as top choice.

5 Summary and Future Work

In this paper, we have presented Handwritten CAPTCHAs, and evaluated the performance of our ”Handwritten CAPTCHA” generation algorithm. Experimental results on two handwritten word recognizers show the gap in the ability between humans and computers in handwriting recognition. We also conducted user studies and human survey on handwritten CAPTCHAs and the analysis of results correlates strongly with our hypothesis.

Next we will consider CAPTCHAs based on handwritten sentence reading and understanding. There are open questions on how long ”Handwritten CAPTCHAs” will resist automatic attacks, how robust is our proposed algorithm for image transformation and degradation, or how easily an image deformation can be reversed and the original image retrieved, as well as concerns based on the future technology development of computer vision systems that could eventually fill the gap in ability between humans and machine reading.

References

[1] Altavistas add url site: altavista.com/sites/addurl/newurl.

[2] The captcha project homepage: http://www.captcha.net.

[3] H. Baird and K. Popat. Human interactive proofs and document image analysis. *Proc. IAPR 2002 Workshop on Document Analysis Systems*, August 2002.

[4] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. *Proc. The Eighth IEEE International Conference on Computer Vision*, 1:454–461, July 2001.

[5] M. Blum, L. von Ahn, J. Langford, and N. Hopper. The captcha project: Completely automatic public turing test to tell computers and humans apart. *http://www.captcha.net*, November 2000.

[6] M. Chew and H. Baird. Baffletext: A human interactive proof. *Proc. SPIE-IST Electronic Imaging, Document Recognition and Retrieval*, pages 305–316, January 2003.

[7] A. Coates, H. Baird, and R. Fateman. Pessimal print: a reverse turing test. *Proc. IAPR 6th International Conference on Document Analysis and Recognition*, pages 1154–1158, September 2001.

[8] J. T. Favata. Character model word recognition. *Proc. Fifth International Workshop on Frontiers in Handwriting Recognition*, pages 437–440, September 1996.

[9] T. K. Ho, J. J. Hull, and S. N. Srihari. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–75, January 1994.

[10] S. Impedovo, P. Wang, and H. Bunke. Automatic bankcheck processing. *Machine Perception and Artificial Intelligence*, World Scientific 28, 1997.

[11] G. Kim and V. Govindaraju. A lexicon driven approach to handwritten word recognition for real-time applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):366–379, 1997.

[12] G. Kochanski, D. Lopresti, and C. Shih. A reverse turing test using speech. *Proc. International Conference on Spoken Language Processing*, September 2002.

[13] L. Li, T. K. Ho, J. J. Hull, and S. N. Srihari. A hypothesis testing approach to word recognition using dynamic feature selection. *Proc. The 11th IAPR International Conference on Pattern Recognition*, pages 586–589, August 1992.

[14] S. Madhvanath and V. Govindaraju. The role of holistic paradigms in handwritten word recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2), February 2001.

[15] S. Madhvanath, V. Govindaraju, V. Ramanaprasad, D. Lee, and S. Srihari. Reading handwritten us census forms. *Proc. Third International Conference on Document Analysis and Recognition*, pages 82–85, 1995.

[16] G. Mori and J. Malik. Breaking a visual captcha. *Computer Vision and Pattern Recognition*, 2003.

[17] H. S. Park and S. W. Lee. An hmmrf-based statistical approach for off-line handwritten character recognition. *IEEE Proceedings of the 13th International Conference on Pattern Recognition*, 2:320–324, August 1996.

[18] S. Rice, G. Nagy, and T. Nartker. *Optical Character Recognition: An Illustrated Guide to the Frontier*, Kluwer, May 1999.

[19] Y. Rui and Z. Liu. Artifacial: Automated reverse turing test using facial features. *Proc. The 11th ACM international conference on Multimedia*, November 2003.

[20] G. Saon and A. Belaid. Off-line handwritten word recognition using a mixed hmm-mrf approach. *Proc. The Fourth International Conference on Document Analysis and Recognition*, 1:118–122, August 1997.

[21] A. Senior and A. Robinson. An off-line cursive handwriting recognition system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):309–321, 1998.

[22] M. Shridhar, G. Houle, and F. Kimura. Handwritten word recognition using lexicon free and lexicon directed word recognition algorithms. *Proc. The Fourth International Conference on Document Analysis and Recognition*, 2:861–865, August 1997.

[23] S. Srihari and E. Keubert. Integration of hand-written address interpretation technology into the united states postal service remote computer reader system. *Proceedings of Fourth International Conference on Document Analysis and Recognition*, pages 892–896, August 1997.

[24] A. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

[25] L. von Ahn, M. Blum, and J. Langford. Telling humans and computers apart (automatically) or how lazy cryptographers do ai. *Technical Report TR CMU-CS-02-117*, February 2002.

[26] J. Xu, R. Lipton, I. Essa, M. Sung, and Y. Zhu. Mandatory human participation: A new authentication scheme for building secure systems. *ICCCN*, 2003.

[27] H. Xue and V. Govindaraju. On the dependence of handwritten word recognizers on lexicons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1553–1564, December 2002.

[28] H. Xue and V. Govindaraju. A stochastic model combining discrete symbols and continuous attributes and its applications to handwriting recognition. *International Workshop on Document Analysis and Systems*, pages 70–81, 2002.

[29] H. Xue, V. Govindaraju, and P. Slavik. Use of lexicon density in evaluating word recognizers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):789–800, June 2002.