Computers beat Humans at Single Character Recognition in Reading based Human Interaction Proofs (HIPs)

Kumar Chellapilla, Kevin Larson, Patrice Simard, Mary Czerwinski
Microsoft Research,
Redmond, WA 98053
{kumarc, kevlar, patrice, marycz}@microsoft.com

Abstract

Human interaction proofs (HIPs) have become commonplace on the internet for protecting free online services from abuse by automated scripts/bots. They are challenges designed to be easily solved by humans, while remaining too hard for computers to solve. Reading based HIPs comprise a segmentation problem and one or more recognition problems. Recent studies have shown that computers are better at solving the recognition problem than the segmentation problem (Chellapilla and Simard, 2004; Chellapilla et al, 2005a).

In this paper we compare human and computer single character recognition abilities through a sequence of human user studies and computer experiments using convolutional neural networks. In these experiments, we assume that segmentation has been solved and the approximate locations of individual HIP characters are known. Results show that computers are as good as or better than humans at single character recognition under all commonly used distortion and clutter scenarios used in todays HIPs.

1 Introduction

Reading-based Human Interaction Proofs[1] (HIPs) have become commonplace for protecting internet web sites against abuse by automated scripts (bots). HIPs are also known as Completed Automated Public Turing tests to tell Computers and Humans Apart (CAPTCHAs). An overview of HIPs can be found in Blum and Baird (2002), Chew, M. and Baird (2003), and Von Ahn et al (2004). The three most common usages of HIPs on the internet are a) when signing up for a free e-mail account (Google, Hotmail, Yahoo!, and others), b) when accessing some online resource such as buying tickets at Ticketmaster.com or executing a whois query at Register.com, and c) for protection against denial of service attacks (e.g. Ticketmaster.com).

The best known application of HIPs is in fighting e-mail spam. On many free email systems, a HIP must be solved to create an account. HIPs are designed to be difficult for computers to solve but easy for humans. So, they effectively impose a cost to create the account (a spammer must use his own valuable time, pay for someone else's time, or provide valuable services in exchange for solving the HIP.) This initial cost of account creation might not be high enough to deter spammers (once a spammer gets a free e-mail account they can send out spam until their activity is detected and their account is shutdown). The initial challenge can be accompanied by subsequent HIP challenges that must be solved to continue sending free e-mail. The cost imposed by the latter approach scales well. Further, spam filter analysis, e-mail usage, and other spam metrics can be used to tailor how often new HIP challenges are presented. Both costs rely on the HIP being secure and posing a problem that is difficult for today's computers. Goodman and Rounthwaite (2004) present several useful applications of HIP for addressing spam problems.

The most successful HIPs are visual and are reading based. Recent studies show that many of these rely on character recognition tasks and can be easily broken using machine learning (Chellapilla et al, 2004; Chellapilla et al, 2005). HIPs that pose a combination of segmentation and recognition tasks have been suggested to improve their security (Chellapilla et al, 2004).

In this paper, we compare human and computer recognition abilities in identifying single characters that have been segmented from HIPs. Section 2 presents a background on the segmentation and recognition problems posed by HIPs and reviews previous work in determining HIP security and human friendliness. A class of character recognition problems modeled using the distortions and clutter present in today's HIPs is presented in Section 3. Single character recognition experiments used to study human and computer

[1] These are also referred to as "Human Interactive Proofs." The term "Human Interaction Proof" is preferred in this paper as it is clearer in indicating that these are tests for human interaction.

Figure 1(a): MSN/Hotmail HIP samples.

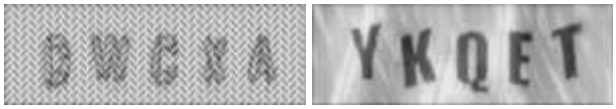

Figure 1(b): Register.com HIP samples.

Figure 1(c): Yahoo! HIP samples.

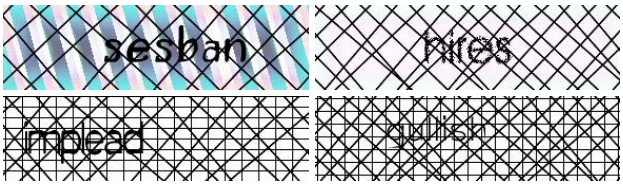

Figure 1(d): Ticketmaster HIP samples.

Figure 1(e): Google HIP samples.

Figure 1: HIP samples.

recognition are presented in Section 4. Results are presented in Section 5 and conclusions are offered in Section 6.

2 Background

Figure 1 presents several examples of reading based HIPs that can be sampled on the internet.

2.1 Segmentation and Recognition challenges

Reading-based HIP challenges typically comprise a segmentation challenge followed by recognition challenges[2]. Solving the segmentation challenge requires the identification of character locations in the right order. The random location of characters, background textures, foreground and background grids or lines, and clutter in the form of arcs make the segmentation problem difficult. Image warp exacerbates the segmentation problem by reducing the effectiveness of preprocessing stages of a segmentation algorithm that attempt to estimate and remove the background textures and foreground lines, etc. Once character locations are reliably identified (in the right order) each of the characters needs to be recognized correctly giving rise to the recognition problem. The character recognition problem is made difficult through changes in scale, rotation, local and global warp, and intersecting random arcs.

[2] Solving a HIP need not require the segmentation and recognition problems to be solved separately. Humans very likely solve both problems simultaneously.

2.2 HIP security

The strength of a HIP (against a computer algorithm) is a combination of the strengths of the constituent segmentation and recognition problems. Several recent efforts have shown that weaknesses in particular HIPs can be easily exploited to break them (Mori and Malik, 2003; Thayananthan et al., 2003; Mov et al., 2004; Chellapilla and Simard, 2004). Chellapilla and Simard (2004) showed that many of the online HIPs are pure recognition tasks that can be easily broken using machine learning (in these HIPs the segmentation problem is trivial to solve). In light of these results, while the recognition challenges still pose a problem, the segmentation challenge is more important in determining HIP strength.

3 Single Character Recognition

The recognition challenges posed by the HIPs are specifically designed to fool off-the-shelf OCR systems e.g. Scansoft's OCR and several others (Coates et al, 2001). These general purpose OCR systems are designed for high quality document scans or images and are brittle to character warp and degrade rapidly in the presence of clutter. On the contrary, one can attempt to solve the recognition problem posed by a particular HIP by building a custom recognizer using machine learning. The custom recognizer is trained on distorted characters extracted from HIP samples. This approach requires a new recognizer to be built for each HIP type (e.g. Yahoo!, Google, Register, MSN, etc). This was exactly the approach adopted in (Chellapilla and Simard, 2004). Convolutional neural networks were used to build recognizers for the Mailblocks, Register, Yahoo!, Ticketmaster, and Google HIPs. Very high recognition rates (80%-95%) were obtained on these HIPs. We note that most of the real-world HIPs are designed to ensure a low human error rate.

When solving the recognition problem, the segmentation problem is assumed to be solved, i.e., we already know the number of characters in the HIP image and their locations. These locations need not be exact. Some tolerance is allowed (a few pixels) as a certain degree of translation invariance can be expected from machine learning based recognizers.

In this section we present a class of distortions and clutter that are designed to mimic those commonly used in today's reading based HIPs (Figure 1). Each distortion and clutter is parameterized and can be scaled

from very easy (little distortion/clutter) to very difficult or unreadable. In this study, to better understand human and computer abilities, the difficulty of the recognition problem is driven much higher than what would be deemed appropriate for use in a real-world HIP.

3.1 A Class of Character Recognition Problems

HIP images contain characters. Both computer and human ability to read characters from such images is dependent on the font size used and the image resolution. Clearly, at very low resolutions and font sizes the characters become illegible. Further, font size, font style (italics, bold, etc), font type (serif, non-serif, monospace, etc), the character set used (English, upper/lower case) etc all play a role in determining the difficulty of the recognition problem. In the interests of tractability, in this paper, the following choices were made in designing character recognition problems for studying human and computer abilities:

a) only upper case English characters (A-Z) and digits (0-9)[3] are used

b) a font size of 30 points is used

c) HIP images are rendered at 96 dots per inch (for humans)

d) Times New Roman font is used (serif font)

We believe these choices, though limiting, provide sufficient variety to compare and contrast human and computer recognition of HIP characters. When designing a real-world HIP, each of these choices must be carefully evaluated both in terms of their impact on HIP security and human friendliness (Chellapilla et al, 2005b).

3.2 Character distortions and arc clutter

Character-based HIPs employ a set of character distortions to make them hard for computers. The basic character transformations include translation, rotation (clockwise or counterclockwise), and scaling. Rotation is usually less than 45 degrees to avoid converting a 6 into a 9, an M into a W or an E etc. Examples of these distortions are presented in Figures 2, 3, and 4. The parameters characterizing translation are in pixels (Figure 3), while those characterizing rotation are in degrees (Figure 4).

Figure 2: Example of Plain Text (M7F47VWC)

[3] Five characters that can be easily confused were discarded. These were I, O, Q, 0, and 1.

Figure 3: Example of Translated Text, levels 10 (5MS9FVLL), 25 (3R2YAZ9X), and 40 (C7AXBZZR)

Figure 4: Rotated Text at levels 15 (PWVDYLVH), 30 (B5PYMMLB), and 45 (GSB5776E)

Both computers and humans find HIPs, using these three transformations, easy to solve. To increase the difficulty of computer-based OCR, we introduce two kinds of warp (Deriche, 1990): Global warp and Local warp.

Figure 5: Letter M under global warp.

Global Warp: Global warp produces character-level, elastic deformations (Figure 5). It is obtained by generating a random displacement field followed by a low pass filter with an exponential decay (Deriche, 1990). The resulting displacement field is then applied to the image with interpolation. These appear to bend and stretch the given characters. The magnitude of the warp is proportional to the total displacement distance of HIP image pixels. The purpose of these elastic deformations is to foil template matching algorithms.

Figure 6: Letter M under local warp.

Local Warp: Local warp is intended to produce small ripples, waves, and elastic deformations along the pixels of the character, i.e., at the scale of the thickness of the characters, rather than the scale of the width and height of the character (Figure 6). The local warp deformations are generated in the same manner as the global warp deformations, by changing the low pass

filter cut-off to a higher frequency. The purpose of the local warp is to foil feature-based algorithms which may use character thickness or serif features to detect and recognize characters.

Clutter: Crisscrossing straight lines and arcs, background textures, and meshes in foreground and background colors are common examples of clutter used in HIPs. In this paper, we used random foreground and background arcs of different thicknesses as clutter. Foreground arcs are rendered in the same color as characters and are designed to join adjacent HIP characters together. Background arcs are rendered in the background color and as such are designed to break up characters into disconnected pieces. Both foreground and background arcs are of constant thickness. Two levels of arc thickness were chosen with thin arcs being 2 pixels wide and thick arcs being 4-5 pixels wide. The combination of thin and thick arcs were chosen to model the thin and thick portions of characters in the Times font. The number of arcs, N_{arcs}, rendered on or around the character was determined by the arc density, D, using:

$$N_{arcs} = ceil\left[WH(D/S)^2\right] \quad (1)$$

where W and H are the width and height of the HIP image, respectively. S is the font size and *ceil* is the ceiling function. One character HIPs were generated as 40 pixel x 40 pixel images centered on the character being used for recognition. These 40 x 40 character images were rendered on a 90x50 HIP image before clutter and warp were added.

4 Method

We carried out a set of seven experiments to determine the recognition rates of humans and computers (the neural network classifier) under the above distortions and clutter. These were (see Figures 7-13):

1. Local warp (+baseline1)
2. Global warp (+baseline1)
3. Thin arcs (+baseline2)
4. Thick arcs (+baseline2)
5. Non-intersecting thick arcs (+baseline2)
6. Thin background arcs (+baseline2)
7. Thick background arcs (+baseline2)

All experiments used one of two baseline settings (baseline1 or baseline2). In experiments 1 and 2, the baseline settings (baseline1) produced random translation (-20 to +20 pixels), scaling (-20 to +20 percent), and rotation (-20 to +20 degrees). The baseline setting in experiments 3-7 (baseline2) produced a random translation of -20 to +20 pixels, random scaling of -20 to +20 percent, a random rotation of -20 to +20 degrees. Further, a global warp of 75, and a local warp of 20 were also used. The warp value indicates the magnitude of the associated warp field and is proportional to the average movement of ink pixels in the HIP (Deriche, 1990).

In each experiment, the associated distortion/clutter parameter was varied from very easy (small) to very difficult (large). To better understand human and computer abilities, the range of parameter setting studied in this paper is much wider than those that would be used when designing real-world HIPs. Figures 7-13 present several examples of characters used in the computer and human user studies.

4.1 Computer Experiments

The computer based recognition engine is a convolutional neural network (Simard et al, 2003) that has been widely used for building single character recognizers for document processing. It yielded the best known error rate of 0.4% on the MNIST database consisting of handwritten digits (0-9). It uses little memory, and is very fast for recognition.

In each experiment, a total of 110,000 random characters were sampled using the distortion and clutter settings. 90,000 characters were used for training and 10,000 were used for validation. Test error was computed over the remaining 10,000 characters. Thirty one characters from {A-Z, 0-9} were chosen. Five characters that can be easily confused were discarded. These were I, O, Q, 0, and 1. Characters were rendered in Times Roman font at a font size of 30 points.

Distortion and clutter (when present) were added to HIPs in the following order a) characters were rendered at random locations (with translation and rotation), b) clutter was added c) global and local warps were applied.

LW 20

LW 40

LW 60

LW 80

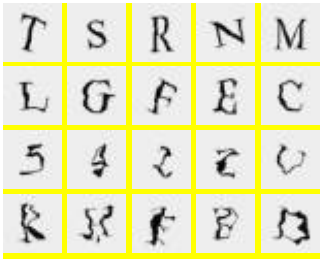

Figure 7: Five sample characters for local warp settings of 20, 40, 60, and 80.

GW 120

GW 180

GW 240

GW 300

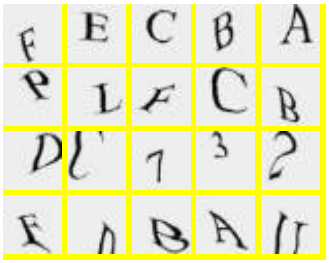

Figure 8: Five sample characters for global warp settings of 20, 40, 60, and 80.

FG 0.5

FG 1.0

FG 1.5

FG 2.0

FG 2.5

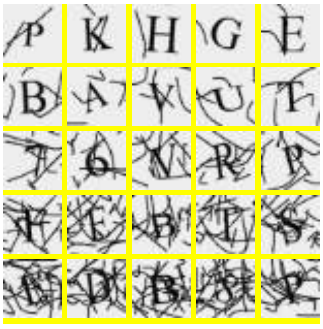

Figure 9: Five sample characters for thin foreground arc densities of 0.5, 1.0, 1.5, 2.0, and 2.5.

FGThick 0.5

FGThick 1.0

FGThick 1.5

FGThick 2.0

FGThick 2.5

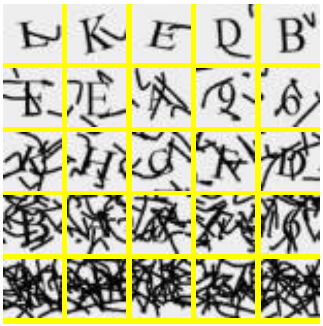

Figure 10: Five sample characters for thick foreground arc densities of 0.5, 1.0, 1.5, 2.0, and 2.5.

FG Thick(NI) 0.5

FG Thick(NI) 1.0

FG Thick(NI) 1.5

FG Thick(NI) 2.0

FG Thick(NI) 2.5

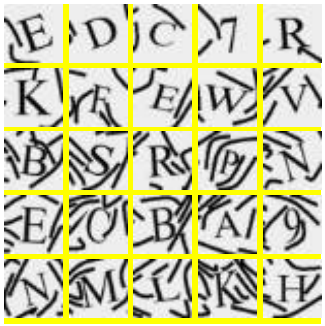

Figure 11: Five sample characters for thick non-intersecting (TNI) foreground arc densities of 0.5, 1.0, 1.5, 2.0, and 2.5.

BG 0.5

BG 1.0

BG 1.5

BG 2.0

BG 2.5

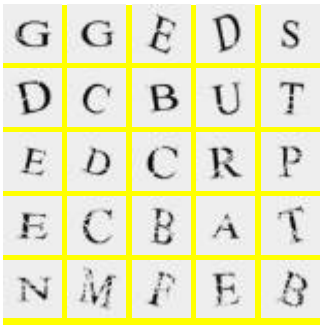

Figure 12: Five sample characters for each background arc densities of 0.5, 1.0, 1.5, 2.0, and 2.5.

BG Thick 0.5

BG Thick 1.0

BG Thick 1.5

BG Thick 2.0

BG Thick 2.5

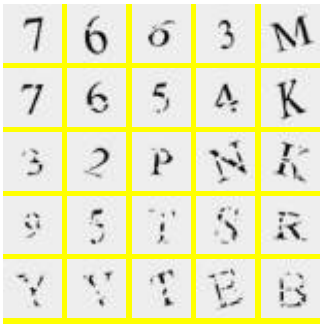

Figure 13: Five sample characters for each thick background arc densities of 0.5, 1.0, 1.5, 2.0, and 2.5.

4.2 HIP User studies

We carried out a user study that attempted to closely match the computer experiments. The study participants were asked to recognize the same distortion conditions that were used with the computer recognition. The studies were designed to be run electronically, allowing participants to do the HIP recognition tasks from the comfort of their own offices. 44 users were recruited to participate in this set of experiments. All were employees at a large software company. Average age of the participants was 33.7 (range of 21-58 years of age), 15 were female, and all had normal or corrected-to-normal vision. All but thirteen of the participants had at least an undergraduate education (though seven responded “other” which could have included a PhD). Participants were compensated by holding a raffle, with one of the participants winning an x-box video game console.

Participants were asked to identify characters in various distortion conditions, the same characters that were used for computer testing in the above experiments. On each screen, participants saw 10 characters per condition selected randomly from a set of 100 test characters. The conditions were the same as those

described in the above section. Each character appeared within a box, and it was clear that there was only one character per box. The participants responded by typing their answer into a text field below the character. The experiments were self-paced, and the participants would move on to the next set of 10 characters whenever they were ready. On average participants spent 19 seconds to recognize 10 characters. Accuracy was defined as the percentage of characters recognized correctly.

5 Results

The first two experiments investigated single character recognition accuracies in the presence of warp. The baseline settings (baseline1) produced only translation, rotation, and scale variations. Experiments three through seven investigate recognition abilities in the presence of foreground and background clutter in the form of thin and thick arcs. The baseline settings (baseline2) produce not only translation, rotation, and scaling, but also a small amount of global (75) and local warp (20).

Local warp (experiment 1): The local warp was incremented in 4 steps from 20 to 80, as shown in Figure 7. The local warp value indicates the magnitude of the local warp field and is proportional to the average movement of ink pixels in the HIP. The computer and human accuracies in the presence of local warp are presented in Figure 14. Human participants had a very high accuracy rate with levels of local warp up to level 40, and poor accuracy at level 60 and above. In contrast, the computer accuracies start out the same as that for humans at a local warp of 20, but stay above 99.5% up to a local warp of 60. Even at a local warp of 80, the convolutional neural network is able to recognize over 96% of the characters correctly.

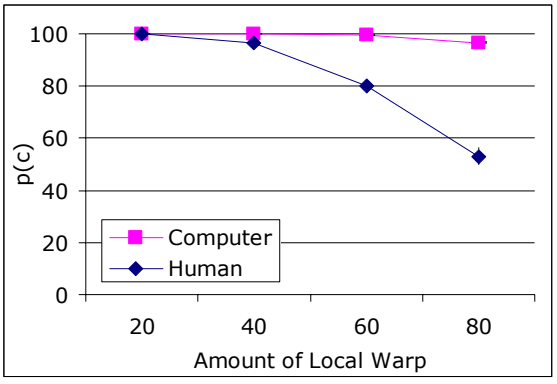

Figure 14: Human and computer accuracy rates (percent) under local warp of 20, 40, 60, and 80.

Global warp (experiment 2): The global warp was increased in 4 incremental steps from 120 to 300, as shown in Figure 8. The global warp value indicates the magnitude of the global warp field and is proportional to the average movement of ink pixels in the HIP. As shown in Figure 15, both humans and computers do well when the global warp field is less than 200 and gradually deteriorate. However, while computers do marginally better, both computer and human accuracies stay above 85%.

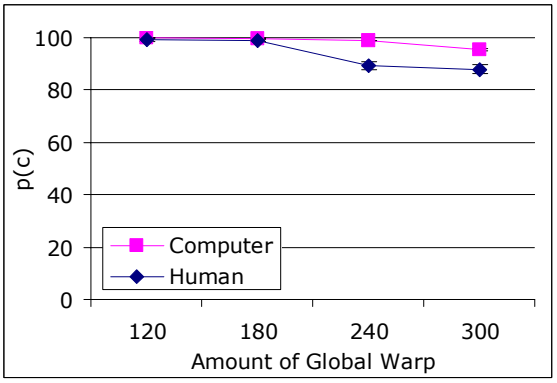

Figure 15: Human and computer accuracy rates (percent) under global warp at 120, 180, 240, and 300.

Thin arcs (experiment 3): In this condition, HIP characters are mixed with clutter in the form of thin intersecting arcs. Note that the arcs are rendered uniformly over the image and do not necessarily intersect the character. However, most do. The arc density (Eq. 1) was varied in five steps from 0.5 to 2.5, as shown in Figure 9. Human participants had a high accuracy rate up to an arc density of 1.5, after which human performance deteriorated quickly. However, computer accuracies remain high (above 95%) throughout the tested arc density range.

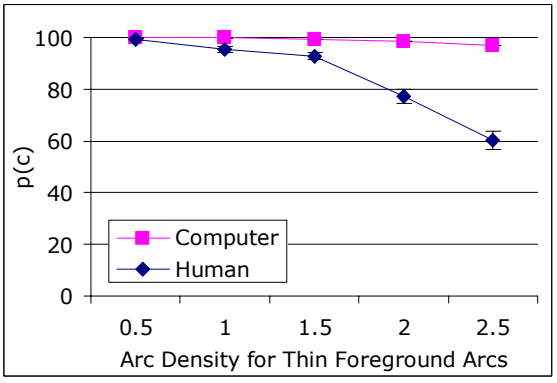

Figure 16: Human and computer accuracy rates (percent) in the presence of thin foreground arcs.

Thick arcs (experiment 4): Figure 17 presents recognition accuracies in the presence of thick intersecting arcs. The arc density (Eq. 1) was varied in five steps from 0.5 to 2.5. Though computer and human

accuracies are high at an arc density of 0.5, they rapidly deteriorate as characters become unrecognizable. Computer recognition is better, but also suffers significant deterioration with increasing arc density.

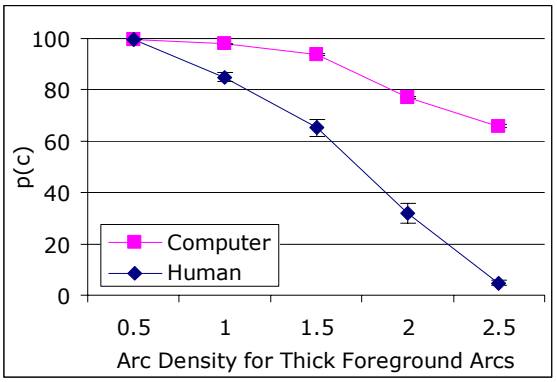

Figure 17: Human and computer accuracy rates (percent) in the presence of thick foreground arcs.

Thick non-intersecting arcs (experiment 5): In this condition, HIP characters are mixed with clutter in the form of thick arcs. However, unlike the case of thick intersecting arcs, these thick arcs do not intersect the characters. At best, they can touch the characters in the HIP image. The arc density (Eq. 1) was once again varied in five steps from 0.5 to 2.5, as shown in Figure 11. As shown in Figure 18, both humans and computers find this scenario quite easy with accuracies never straying too far from 100% throughout the arc density range.

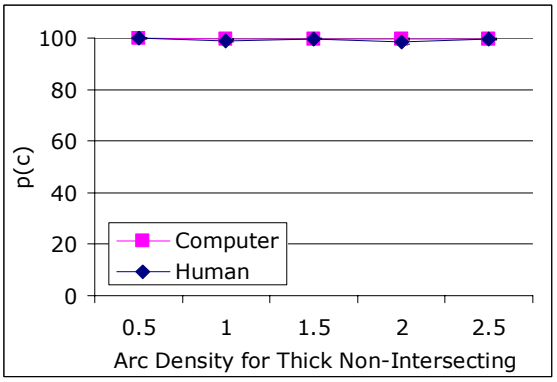

Figure 18: Human and computer accuracy rates (percent) in the presence of thick non-intersecting foreground arcs.

Thin background arcs (experiment 6): Unlike the foreground arcs that connect neighboring characters (in a multi-character HIP), background arcs are designed to break up characters in to two or more pieces. The background arc density (Eq. 1) was varied in five steps from 0.5 to 2.5, as shown in Figure 12. As shown in Figure 19, both humans and computers find this scenario quite easy and achieve near 100% accuracy rates. Computer and human performances are nearly indistinguishable.

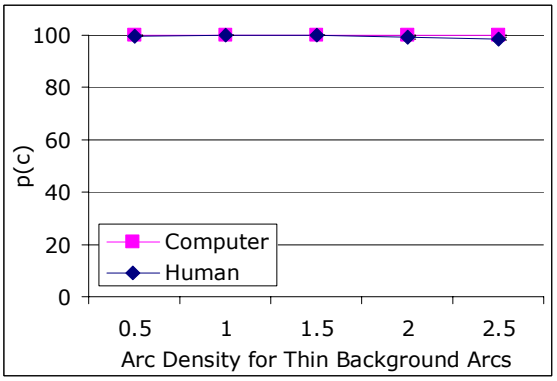

Figure 19: Human and computer accuracy rates (percent) in the presence of thin background arcs.

Thick background arcs (experiment 7): In comparison with thin background arcs, thick background arcs breakup the characters much more rapidly. The arc density (Eq. 1) was varied in five steps from 0.5 to 2.5, as shown in Figure 13. As shown in Figure 20, both humans and computers rarely make errors when the arc density is less than 1.5. As the arc density increases past 1.5, human performance drops mildly, while computer accuracy remains almost unchanged.

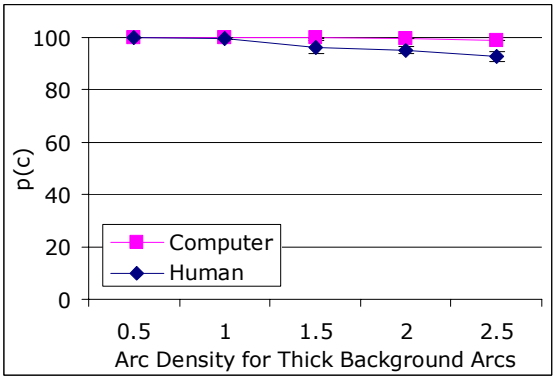

Figure 20: Human and computer accuracy rates (percent) in the presence of thick background arcs.

In all seven experiments, computers did as well as or better than humans at recognizing HIP characters in the presence of distortion and clutter. Thin background arcs and thick non-intersecting foreground arcs were the easiest for both humans and computers. Humans and

computers did well and achieved near 100% accuracies. Thick background arcs and global warp presented low to medium difficulty. For low levels, both humans and computers did well (near 100% accuracy). With increasing difficulty computers did only marginally better than humans. Local warp and thin intersecting foreground arcs presented moderately challenging recognition tasks for both computers and humans. At low levels, accuracy was high (above 95%). However, with increasing difficulty, while human performance degraded significantly, computer performance only dropped marginally (less than 5%). Thick foreground arcs posed the most difficult problem. Both human and computer recognition rates quickly dropped as characters became unreadable. However, computers still did much better than humans in recognition accuracy. As seen in Figure 10, the difficulty of the thick intersecting arcs arises from a loss in character content.

6 Conclusion

The recognition problem posed by HIPs was modeled using a sequence of character transformations such as translation, rotation, scaling, warp (local and global), and clutter (thin and thick foreground and background arcs). Using this model, seven experiments were designed to assess human and computer abilities in solving the recognition problem posed by today's HIPs. User studies were done to assess human accuracy. Convolutional neural networks were trained using machine learning to recognize characters in each of these experiments. Experimental results comparing human and computer recognition of HIP characters indicate that computers a) do as well as humans on the easy problems, b) are marginally better at low and medium difficulty scenarios, and c) beat humans at high distortion and clutter settings. Overall, the results clearly indicate that the recognition problem posed by HIPs is easier for computers than humans. In light of this new result, HIP strength against computer attacks is essentially determined by the strength of the segmentation problem. As a result, HIP designers can no longer rely on the recognition problem(s) to contribute much strength to the HIP. We note that several choices made while designing the recognition problem also affect the ability to segment a HIP. The next generation HIPs must rely on posing a strong segmentation problem and leveraging human segmentation ability to provide HIP security.

Acknowledgements

We would like to acknowledge Chau Luu for her help with developing the website for the user studies and collecting user data.

References

Chellapilla K., and Simard P. (2004), "Using Machine Learning to Break Visual Human Interaction Proofs (HIPs)," *Advances in Neural Information Processing Systems 17*, Neural Information Processing Systems (NIPS'2004), MIT Press.

Chellapilla K., Larson K., Simard P., and Czerwinski M. (2005a), "Designing Human Friendly Human Interaction Proofs (HIPs)," in Conference on Human factors In computing systems, CHI 2005. ACM Press.

Chellapilla K., Larson K., Simard P., and Czerwinski M. (2005b), "Building Segmentation Based Human-friendly Human Interaction Proofs (HIPs)," in Proc. of the Second Intl' Workshop on Human Interactive Proofs, HIP2005. Springer-Verlag.

Chew, M. and Baird, H. S. (2003), "BaffleText: a Human Interactive Proof," Proc., 10th IS&T/SPIE Document Recognition & Retrieval Conf, Santa Clara, CA, Jan. 22.

Coates A. L., Baird H. S., and Fateman R. J. (2001), "Pessimal Print: A Reverse Turing Test," *Sixth International Conference on Document Analysis and Recognition (ICDAR '01)*, Sep. 10 - 13, Seattle, WA.

Deriche R. (1990), "Fast Algorithms for Low-Level Vision", IEEE Trans. on PAMI, 12(1), pp. 78-87.

Goodman J. and Rounthwaite R. (2004), "Stopping Outgoing Spam," Proc. of the 5th ACM conf. on Electronic commerce, New York, NY. 2004.

Blum M. and Baird H (2002), *First Workshop on Human Interactive Proofs*, Palo Alto, CA, January 2002.

Mori G, Malik J (2003), "Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA," *Proc. of Comp. Vision and Pattern Rec. (CVPR) Conf.*, IEEE Computer Society, vol.1, pages:I-134 - I-141, June 18-20, 2003

Simard, P.,Y., Steinkraus, D., Platt, J. (2003) "Best Practice for Convolutional Neural Networks Applied to Visual Document Analysis," *International Conference on Document Analysis and Recognition (ICDAR)*, IEEE Computer Society, Los Alamitos, pp. 958-962, 2003.

Von Ahn L, Blum M, and Langford J. (2004) "Telling Computers and Humans Apart (Automatically) or How Lazy Cryptographers do AI." *Comm. of the ACM* ,47(2):56-60.

Von Ahn L, Blum M, and Langford J, *The Captcha Project.* http://www.captcha.net