Generation and Use of Handwritten CAPTCHAs

A. Rusu1, A. Thomas2, V. Govindaraju2

1 Fairfield University, Fairfield, CT, US
2 University at Buffalo, SUNY, Buffalo, NY, US

The date of receipt and acceptance will be inserted by the editor

Abstract. Automated recognition of unconstrained handwriting continues to be a challenging research task. In contrast to the traditional role of handwriting recognition in applications such as postal automation, bank check reading, etc., in this paper we explore the use of handwriting recognition in designing CAPTCHAs for cyber security. CAPTCHAs (Completely Automatic Public Turing tests to tell Computers and Humans Apart) are automatic reverse Turing tests designed so that virtually all humans can pass the test but state-of-the-art computer programs will fail. Machine-printed, text-based CAPTCHAs are now commonly used to defend against bot attacks. Our focus is on exploring the generation and use of handwritten CAPTCHAs.

We have used a large repository of handwritten word images that current handwriting recognizers cannot read (even when provided with a lexicon) for this purpose and also used synthetic handwritten samples. We take advantage of both our knowledge of the common source of errors in automated handwriting recognition systems as well as the salient aspects of human reading. The simultaneous interplay of several Gestalt laws of perception and the geon theory of pattern recognition (that implies object recognition occurs by components), allows us to explore the parameters that truly separate human and machine abilities.

Key words: Handwriting recognition – CAPTCHA – HIP

1 Introduction

Interpreting handwritten text is a task humans usually perform easily and reliably. However, automating the process is difficult because it involves simultaneously recognizing the symbols and comprehending the message conveyed. Although progress in optical character recognition (OCR) accuracy has been considerable, it is still inferior to that of a first grade child [20].

A review of the handwriting recognition literature [3], [10], [14], [15], [17], [23], [25] shows that while some of the computer algorithms demonstrate human-like fluency, they fail when the images are degenerated, poorly written, or without a context. Thus, there is currently a gap between human and machine abilities in reading handwriting under noisy conditions which can be explored through controllable parameters that capture aspects of handwriting such as legibility, overlapping of words, broken strokes, and the extent of overrun characters.

CAPTCHA is part of the set of protocols known as HIP’s, which allows a person to authenticate as belonging to a select group, for example human as opposed to machine, adult as opposed to child, etc. HIP’s operate over a network, without the burden of passwords, biometrics, special mechanical aids, or special training [1]. Since CAPTCHAs exploit the areas where computers are not as good as humans (yet), handwritten word image challenges are a strong candidate for these tests.

To the best of our knowledge, this is the first research effort in the design of Handwritten CAPTCHAs (Figure 1). We believe it is important to explore this avenue of research because several machine-printed, text-based CAPTCHAs (Ez-Gimpy and Gimpy-R developed by researchers at Carnegie Melon University) have already been broken. Mori and Malik (University of California at Berkeley) report automated programs that can solve Ez-Gimpy with accuracy of about 83%. The Cambridge vision group has reported 93% correct recognition rate on Ez-Gimpy, and a group from Areté Associates have reported 78% accuracy on Gimpy-R [6]. Handwritten text presents additional challenges that are rarely encountered in machine-printed text. Further, most problems faced in reading machine-printed text (for example character recognition, word segmentation, or letter segmentation) are exacerbated in handwritten text. Results of our experiments support our hypothesis that handwrit-
handwritten text images are well suited for use as CAPTCHAs. 
This is the basis for the research presented in this paper.

Fig. 1. Handwritten CAPTCHA challenges.

Handwriting recognition has been successfully used in several applications, such as postal address interpretation [26], bank-check reading [11], and forms reading [16]. These applications are all characterized by small or fixed lexicons accompanied by contextual knowledge. Recognition of unconstrained handwriting is difficult because of diversity in writing styles, inconsistent spacing between words and lines, and uncertainty of the number of lines on a page as well as the number of words in a line [24]. In addition, 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.

It must be noted that without the context of a lexicon, unconstrained cursive handwriting recognition (offline) is extremely difficult. Furthermore, the recognition accuracy drops dramatically with an increase in the lexicon size. The results in Figure 2 [28] are based on fairly well-written clean images extracted from US mail piece images. They show the execution speed and recognition accuracy of the system. Thus, generating handwritten word images which are challenging for computers programs but relatively effortless for is a worthwhile avenue to pursue. One obvious approach to increasing the difficulty for computer programs would be to increase the lexicon size. However, this may not be always practical, because it would require presenting human users with a very large lexicon in a challenge-response test. This would take up large area on the computer screen making it onerous on genuine human users. In this paper we have explored alternative ways of increasing the difficulty for automated programs.

2 Design of Handwritten CAPTCHAs

We have generated Handwritten CAPTCHA challenges and tested them with three state-of-the-art handwriting recognizers: Word Model Recognizer (WMR), Character Model Recognizer (CMR), and Accuscript (HMM) [9], [12], [28]. Our methodology is motivated by the Gestalt laws of perception and the geon theory.

2.1 Gestalt Laws of Perception

“Gestalt” in German means “shape” which in psychology implies the idea of perception in context. It is based on the observation that we often experience things that are not part of our simple sensations [13]. What we see is believed to be an effect of the whole event, which is more than the sum of the parts (Figure 3). The main idea is one of “grouping”, and the concept is similar to the holistic word recognition approaches that focus on recognizing the entire word as a group [15] without breaking it into character units. The Gestalt laws of organization that relate to our work in designing robust CAPTCHAs include a subset of the following:

1. Proximity: refers to how things tend to be grouped together by distance or location
2. Similarity: refers to how elements that are similar tend to be grouped together
3. Symmetry: refers to how things are grouped into figures according to symmetry and meaning
4. Continuity: refers to grouping by flow of lines or by alignment
5. Closure: refers to how elements are grouped together if they tend to complete a pattern allowing perception of shapes that are physically absent
6. Familiarity: refers to how elements are more likely to form groups if they appear familiar
7. Figure-ground distinction: perception involves not only organization and grouping, but also distinguishing an object from its surroundings. We perceive an object as a foreground and the area around that object as the background

For example in Figure 3e, a set of dots outlining the shape of a 'B' is likely to be perceived as a 'B', and not as a set of dots. It is also more natural for us to see the o's as a line within a field of x's (Figure 3a). In Figure 3b we are likely to see three collections of two vertical lines each as well as grouping the dots in three sets based on their proximity. Despite the law of proximity prompting us to group the brackets nearest to each other together, in Figure 3d symmetry overwhelms our perception and makes us see them as pairs of symmetrical brackets. We can see a line, for example, as continuing through another line, rather than stopping and starting as two angles (Figure 3c). The elements in an image are grouped together if we are used to seeing them together. For example we are used to seeing rectangles and squares rather than other odd shapes in Figure 3g. We also seem to have a tendency to perceive one aspect of an event as the figure or foreground and the other as the back-ground (Figure 3g). We can see two different things but not both

<table>
<thead>
<tr>
<th>Lexicon</th>
<th>Speed on UltraSparc 10</th>
<th>Top 1%</th>
<th>Top 2%</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.021</td>
<td>96.56</td>
<td>98.77</td>
</tr>
<tr>
<td>100</td>
<td>0.031</td>
<td>89.12</td>
<td>94.06</td>
</tr>
<tr>
<td>1000</td>
<td>0.089</td>
<td>75.38</td>
<td>86.29</td>
</tr>
<tr>
<td>20000</td>
<td>0.994</td>
<td>58.14</td>
<td>66.49</td>
</tr>
</tbody>
</table>

Fig. 2. Speed (in seconds) and accuracy (the percentage of correctly recognized words) of a lexicon-driven handwritten word recognizer when the lexicon contains 10, 100, 1,000, and 20,000 entries (words).
2.2 Geon Theory

We have also explored the geon theory [4] of pattern recognition (or recognition by components) as it provides cues on what is desirable to be preserved in image reconstruction, namely *edges* and *intersections*. In Figure 4 we can see the importance of junctions, crossing strokes, concavities and convexities in recognition accuracy.

![Image](image_url)

**Fig. 4.** Evidence of Geon Theory when objects are lacking some of their components. a) Recoverable objects, b) Non-recoverable objects.

The geon theory explains how moderately occluded or degraded images and new instances of objects, are successfully recognized by the human visual system. Humans are also capable of recognizing objects from various view points, even views that have never been seen before [5] (Figure 5). Objects can be recognized despite variations in size because it does not change the structure of an object (the geons and their spatial organization) (Figure 6). Also changes in position do not disrupt recognition accuracy (e.g., the book is on the desk, or letter *a* is next to letter *b*).

![Image](image_url)

**Fig. 5.** Object recognition is size invariant.

![Image](image_url)

**Fig. 6.** Object recognition is rotational invariant.

Geon theory researchers [19] explain word perception to be based on the following rules:

---

**Fig. 3.** Several examples for Gestalt laws of perception: a) similarity, b) proximity, c) continuity, d) symmetry, e) closure, f) familiarity, g) figure-ground, and h) memory.
1. Words contain specific visual elements (e.g., connected lines at different orientations)
2. Visual elements are presented in orderly fashion (e.g., lines connected in specific ways denote specific letters)
3. Combination of letters follows specific rules (e.g., rules of English language, letters combined in the form of consonant-vowels)
4. Words convey meaning

2.3 Transforms based on Gestalt and Geon Theories

Several examples of handwritten word images on which OCR systems fail (but humans can recognize easily) are shown here. We have successfully designed transforms to deform these handwritten word images by using the Gestalt and Geon theories. This ensures that the challenges are human readable but beyond the abilities of current handwriting recognizers (Section 5). The transforms are as follows:

1. **Image transformation method**: Create horizontal or vertical overlaps (Figure 7a).
   
   **Gestalt laws that help humans in recognizing the deformed images**: proximity, symmetry, familiarity, continuity, figure-ground.

2. **Image transformation method**: Add occlusions by circles, rectangles, lines, etc., (Figure 7b).
   
   **Gestalt laws that help humans in recognizing the deformed images**: closure, proximity, continuity, familiarity.

3. **Image transformation method**: Add occlusions by waves from left to right (Figure 7c).
   
   **Gestalt laws that help humans in recognizing the deformed images**: closure, proximity, continuity.

4. **Image transformation method**: Add occlusions using the same pixels as the foreground pixels (Figure 7d).
   
   **Gestalt laws that help humans in recognizing the deformed images**: familiarity, figure-ground.

5. **Image transformation method**: Use empty letters, broken letters, edgy contour, fragmentation, etc., (Figure 7e).
   
   **Gestalt laws that help humans in recognizing the deformed images**: closure, proximity, continuity, figure-ground.

6. **Image transformation method**: Split the image in parts and displace (Figure 7f).
   
   **Gestalt laws that help humans in recognizing the deformed images**: closure, proximity, symmetry.

7. **Image transformation method**: Split the image in parts and spread (i.e., mosaic effect) (Figure 7g).
   
   **Gestalt laws that help humans in recognizing the deformed images**: closure, proximity, continuity, symmetry.

8. **Image transformation method**: Add extra strokes (Figure 7h).
   
   **Gestalt laws that help humans in recognizing the deformed images**: familiarity, figure-ground.

9. **Image transformation method**: Change word orientation, stretch, compress (Figure 7i).

**Gestalt laws that help humans in recognizing the deformed images**: Memory, internal metrics, familiarity of letters and letter orientation.

![Fig. 7](image-url)

**Fig. 7.** The truth words are: a) Lockport, Silver Creek, Young America, W. Seneca, New York. b) Los Angeles, Buffalo, Kenmore. c) Young America, Clinton, Blasdell, d) Albany, Buffalo, Rockport. e) Buffalo. f) Syracuse, Tampa, Amherst, Kenmore. g) Buffalo, Hamburg, Waterville, Lewis- ton. h) Binghamton, Lockport, Rochester, Bradfordon. i) W. Seneca

3 Generation Algorithm

We describe here our methodology for automatic generation of random and "infinitely many" distinct handwritten CAPTCHAs (Figure 8).
We have used handwritten US city name images available from postal applications (CEDAR CDROM, Figure 9). Transformations are applied to randomly chosen handwritten images from this set.

We have also constructed handwritten word images (actual and unreal) by gluing together characters randomly chosen from a set of 20,000 handwritten character images of isolated upper and lower case alphabet letters (Figure 10). Character size, height, stroke width, slope, etc., require prior normalization before concatenating to assure consistent aspect ratio for the entire word.

We have also used the handwriting distorter described in [22] for generating (potentially) an infinite number of different synthetic samples from handwritten words. Our design exploits the knowledge of the common source of errors in automated handwriting recognition systems and also takes advantage of the cognitive aspects of human reading by incorporating the Gestalt laws of perception and geometry. Following is the algorithm (Figure 11):

**Algorithm Handwritten CAPTCHA Generation**

*Input:* Original (randomly selected) handwritten image (existing US city name image or synthetic generated word image with length 5 to 8 characters or meaningful sentence).

*Output:* Handwritten CAPTCHA image.

**Method:**

- Randomly choose the number of transformations.
- Randomly establish the transformations corresponding to the given number. Some rules apply. For example, no transformation can be applied more than once to the same image (multiple times, it drastically degenerates the image and affects human reading abilities).

- Assign a priori order 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 an 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 remains difficult for recognizers.
- Apply each transformation in sequence and generate the output-deformed image.
- Update the image after each transformation, so that the effect is cumulative.

**end Algorithm.**

We have demonstrated our method of generating the CAPTCHAs with the described transformations cannot be easily "reverse engineered" (Section 3.1) by hackers.

![Image](image-url)

**Fig. 8.** Human readable but OCR failing image challenges.

**Fig. 9.** Handwritten US city name images collected or available from postal applications.

**Fig. 10.** Isolated upper and lower case handwritten characters used to generate word images, real or nonsense.

**Fig. 11.** Handwriting CAPTCHA puzzle generation.

We have examined the sources of errors of computer recognition algorithms. Segmentation errors (over- and under-segmentation), recognition errors (confusions between lexicon entries), and image quality are the most common. We have considered all the normalization operations that word recognizers use prior to recognition and have introduced the related distortions deliberately. Also, given our knowledge of the extent of the distortions a word recognizer can tolerate, we are able to generate images that cannot be easily de-noised by such pre-processing algorithms.

Following transformations have been applied because current recognizers account for them in the pre-processing step.

1. **Noise:** Add lines, grids, arcs, circles, and background noise; use random convolution masks, and special filters (e.g., multiplicative/impulsive noise, blur, spread, wave, median filter, etc.) (Figure 12).
2. **Segmentation:** Segmentation errors have been exploited in machine-printed CAPTCHAs. This is even more effective in handwriting ([7]) as the gaps between words are not of constant size. Our approach is to
delete ligatures or use letters and digits touching with some overlap to make segmentation difficult. We also use stroke thickening to merge characters (Figure 13).

3. **Lexicon**: Use lexicons with similar entries, large lexicons, or no lexicons. Use words with confusing and complex characters such as ‘w’ and ‘m’ (Figure 14). If we allow the use of non-sense words then there are at least 160 similar words for a word of length 6 within an edit distance of 1.

4. **Normalization**: Create images with variable stroke width, slope, and rotations, randomly stretch or compress portions of a word image (Figure 15).

It is intuitively understood that word recognition with larger lexicons is more difficult [9], [12]. Another way to categorize the difficulty of a word recognizer task is by the similarity between lexicon entries, defined as the distance between handwritten words. (Lexicon Density, [27], [29]). Although the idea of generating random lexicons with higher density is expected to provide additional handwritten CAPTCHAs, this direction of research was not pursued, since preliminary results raised human usability issues (Figure 16).

Consider the results in Figure 2 that are based on clean, US mail piece images. We ensured that all the lexicon entries are real words and the true word is always present in order to make a fair comparison with human ability, which relies heavily on context.

Which of the following sequence of characters better describe the image:

A. Orlando
B. Rochester
C. Tonawanda

Fig. 12. Several transformations that affect image quality.

Fig. 13. Segmentation errors are caused by oversegmentation, merging, fragmentation, ligatures, scralls, etc. To make segmentation fail we can delete ligatures, use touching letters/digits, merge characters for over segmentation or to be unable to segment.

Fig. 14. Increasing lexicon challenges such as size, density, and availability cause problems to handwriting recognizers.

Fig. 15. Transformations that affect the image features.

Fig. 16. Multiple choice handwritten CAPTCHA.

**Controlling Distortions** We remove features or add non-textual strokes or noise to a handwritten image in a systematic fashion based on Gestalt segmentation and grouping principles in order to pose difficulties for ma-
chine recognition, while preserving the overall letter legibility for human reading.

1. Create horizontal or vertical overlaps: use smaller distance overlaps for same words and bigger distance overlaps for different words. Try to avoid the situations presented in Figure 17.

![Fig. 17. Confusing results: a) if the overlaps are too large both humans and machines could recognize a wrong word (e.g., Williamsville where in reality the truth word is Williamsville), b) machines can read the image if the overlaps are too small (the truth words is Lockport).](image)

2. Add occlusions by circles, rectangles, lines, and random angles. Keep the occlusions small so that they do not hide the letters completely but are correlated with the image stroke width and size (Figure 18).

![Fig. 18. Word images that have been recognized by machine due to size uncorrelations. The truth words are: Cheektowaga, Young America.](image)

3. Add occlusions by waves from left to right on the entire image, with various amplitudes and wavelength or rotate them by an angle. Choose areas with more foreground pixels (e.g., on bottom part of the text image, not too low and not too high, to avoid the situations in Figure 19).

If the occlusions are applied on ascenders or descenders of characters that could generate inter-character ambiguities that decrease the performance of any handwriting recognizer. For example, a q with its descender chopped off will appear more as an a. Occlusions and line removal in general are still open problems in handwriting recognition. Moreover, even if in particular instances they may seem to work, the methods are still not able to improve the recognition with more than few percents.

4. Add occlusion using the same pixels as the foreground pixels, arcs, or lines, with various thicknesses. Curved strokes could be confused with parts of a character. Use asymmetric strokes such that the pattern cannot be learned (unwanted results in Figure 20).

![Fig. 19. The area where the occlusions are applied has to be carefully chosen. We show examples here that do not pose enough difficulty to computers and therefore they have recognized the words Albany and Silver Creek.](image)

5. Use empty letters, broken letters, edgy contours, and fragmentation. Break characters so that general image processing techniques cannot reconstruct the original image. Use various degrees of fragmentation (Figure 21).

![Fig. 20. Example of handwritten image that was recognized by one of our testing recognizers. The truth word is Lewiston.](image)

6. Split the image in two parts on horizontal and displace the parts in opposite directions. Learn reasonable position for horizontal displacement to adjust and decide the range based on human/machine results.

7. Split the images in parts, either by a vertical/horizontal line or by diagonals, and spread the parts apart (i.e., mosaic effect). Symmetry in displacement helps image reconstruction for humans.

8. Add occlusion using the same pixels as the foreground pixels. Extra strokes are confused with character components when they are about the same size, thickness, and curvature as the handwritten characters.
9. Change word orientation even for just a few letters.
   Use variable rotation, stretching, and compressing.

Figure 22 shows a set of images that have been successfully recognized by humans but on which state-of-the-art handwriting recognizers failed (more examples have been included in Section 2.3, Figure 7). We note that all the methods described here would work for machine-printed text images as well. However, the advantage of using handwriting is that most handwritten text challenges are more difficult.

![Images of handwritten text examples](image)

**Fig. 22.** Examples of handwritten image transformations that are easy for humans to interpret but OCR systems fail: a) extra strokes, b) occlusions by black waves, c) vertical and horizontal overlaps, d) occlusions by circles, e) occlusions by white waves, f) fragmentation, g) stroke displacement, h) mosaic effect. The truth words are: Liverpool, Angola, Kenmore, Bradenton, Jamestown, Boston, Chicago, Niagara, Denver, America, Niagara, Longmont, Valley, Newark, Kansas, Albany.

### 3.1 Automated Reversing of Distortions

We have developed several methods to attack the proposed handwritten CAPTCHAs and used pre-processing techniques to reverse the above transformations as follows. This was done by a different group of programmers who were given the list of transformations applied. This is in keeping with the idea of HIPs where the algorithm for the creation of the CAPTCHAs is open public knowledge.

- **Gaps**

  These transformations can be reverted in particular circumstances. For example if we consider adding a gap in the word image and separating the word into 2 unconnected halves, then by connecting these two disconnected parts, the new word image that is obtained is very similar to the original word image. For example, consider the word image in Figure 23a. There is a continuous gap that runs horizontally across the word. Due to this break in the continuity of the pixel sequence, recognizers fail on the image. Reverse Engineering of the transformation is shown in Figure 23b.

![Gap transformation examples](image)

**Fig. 23.** Gap transformation: a) before pre-processing, b) after pre-processing. The truth word is: Buffalo.

However, it becomes more difficult to revert the transformations when the handwriting has a slant.

- **Mosaic effect**

  In some cases the transformation splits the word image into 2 halves and separates each part by a constant gap. It is relatively easy to revert this kind of transformation by simply joining the lower half of the word image to the upper half. However, the reverting procedure is less successful if there is a discontinuity in the strokes caused by removing some part of the original image pixel streams and using displacements.
- **Waves**
  Another simple form of transformation that can be applied is a horizontal wavy line that runs across the word image without affecting the human readability of the word. The wave line that overlaps the word image makes it difficult for the recognizer to identify the correct word.

  It can be observed that the human readability of the word is not hampered, and at the same time while the added noise makes it difficult for a word recognizer. If the reverse engineering software has this information, it can usually make the reversing task possible (Figure 24). However, when the pixel thickness of the wavy line is the same as the pixel thickness of the character strokes then the reversing task becomes even more difficult.

![Fig. 24. Wave transformation: a) before pre-processing, b) after pre-processing. The truth words are: WSeneca, Young America.](image)

- **Overlapping**
  Another form of transformation is overlapping of a word image over itself with a slight shift in the X or Y direction. We tried the reversal operation and it works in some cases (Figure 25).

![Fig. 25. Overlapping transformation: a) before pre-processing, b) after pre-processing. The truth word is: Uslf](image)

The quality of pre-processed word image depends upon a number of parameters such as the pixel thickness of characters, the distance between the two overlapped images, etc. It was relatively easy to refine the image when the two overlapping images were well separated. But when the separation was not large enough (small overlaps), it was difficult to successfully filter the image (Figure 26).

- **Ares / Jaws**
  It is quite difficult to remove the noisy curves that overlap a word image. The thickness of the curves and the thickness of the characters in the image is nearly the same. Also the continuity of these noisy curves is quite similar to the strokes in the handwritten word, which makes it difficult to separate it out the noise.

- **Fragmentation**
  Fragmentation also is not a method that pre-processing techniques could successfully reverse. The methods we have tried have repeatedly failed to correctly revert the images.

- **Background noise**
  One possible image deformations that can be applied to make the image more complex is adding background noise (Figure 27).

![Fig. 26. Overlapping transformation with bad reverse: a) before pre-processing, b) after pre-processing. The truth words are: Matthews, Paso.](image)

![Fig. 27. Background noise that cannot be reverted. The truth words are: Los Angeles, Silver Creek, Young America.](image)

When the image pixels that form the handwritten characters are uniformly blended with the background pixels (as in Figure 27), it becomes very difficult to filter out the noise and retrieve the actual handwritten word image. Also, if random strokes are added
to the image (whose pixel thickness is similar to the thickness of the characters in the image), the uncertainty of differentiating the actual character strokes from the noisy strokes increases.

Figure 28 is an example that can be reverted by noise removal techniques.

![Fig. 28. Background noise that can be reverted. The truth word is: Wits](image)

We have run the state-of-the-art recognizers on the reverted images. However it is practically impossible to have a program which accounts for all transforms. Although in particular instances certain pre-processing methods may seem to work, it is nearly impossible to generally and successfully apply them on images that have been deformed at random with one (or more than one) transformation proposed in this paper. Moreover, combining more than one deformation complicates the revert process.

### 4 Image Complexity

In addition to the errors caused by image quality, image features, segmentation, and recognition, we have also explored the influence of image complexity on handwriting recognition (or how hard it is to read handwriting) and compared humans’ versus machines’ recognition rates. In [21] we have investigated the influence of handwritten image complexity and Gestalt laws of perception on this gap.

However, in our attempt to quantify the strength of human reading abilities, we have obtained inconclusive results. In our experiments, neither image density nor perimetric complexity has shown to predict the efficiency of humans in handwritten word recognition ([21]). On the other hand, Gestalt and geon components have been shown to play an important role in the relative identification of characters and at the same time pose problems to machine recognition.

Unlike the results reported by [18] for letter identification, in our experiments perimetric complexity does not correlate well with human recognition accuracy as seen on 1058 distinct handwritten sample images tested on human subjects (Figure 29). However, these results support the importance of other factors involved in human handwriting recognition such as the role of Gestalt principles, as well as preservation of geon components such as intersections and edges, and having prior knowledge of the context. Similar non-correlations between the perimetric complexity and human recognition efficiency have been reported in [2] for machine-printed word images. The researchers in [2] try to predict legibility of ScatterType challenges using features that can be automatically extracted from the images such as perimetric image complexity. However the metric that worked well on another machine-printed CAPTCHA [8] failed to predict legibility in that case. We have found similar non-correlations for handwritten CAPTCHAs.

![Fig. 29. Humans recognition accuracy vs. perimetric complexity as a percent of correct answers per bin (with a total range for perimetric complexity of 100 equal bins; the complexity range is [0,.20,000]) ([21]).](image)

### 5 Handwriting-based HIP System

Our HIP system has three main components: (i) the actual CAPTCHA challenge (handwritten word image) that is presented to the user, (ii) user response or the answer to the challenge, and (iii) a method to validate the user response and report success or failure. The three components have been implemented and tested online (Figure 30). The modules can be used independently to secure any online application. The process is entirely automatic so that it is easily deployable and there is no risk of image repetition, which enforces higher security.

We have used several sets of image files in TIFF and HIPS formats. We have generated handwritten CAPTCHAs and performed test legibility on human volunteers and the state-of-the-art handwriting recognizers available at CEDAR (WMR, CMR, and Accuscript) [9], [12], [28].

For each image, we have produced a deformed version by applying successive transformations according to the handwritten CAPTCHA generation algorithm in Section 3. We assume that a valid lexicon is provided and that for every image the corresponding truth word is always present. We ran tests on lexicons of size 4,000 and 40,000 (the entire list of US city names).

We have conducted several experiments with both human subjects and machine. The handwritten CAPTCHA tests are graded pass or fail, where pass is granted when all the characters of the word are correctly recognized, and fail otherwise.
5.1 Various Transformations on Real Words - US City Names

Method

Procedure. The first experiment involves a database of 4,127 city names images. They are all handwritten citywords (cursive and hand-printed, with unconstrained writing styles) manually extracted from mail pieces. Each image contains one or two words that correspond to a U.S. city name. We have implemented an automated version of the deformation algorithm, and a number of transformations (up to three) are applied to each image. The transformations considered are: adding lines, grids, arcs, background noise, applying convolution masks and special filters, using variable stroke width, slope, rotations, stretching, compressing. We performed tests by running WMR and Accuscript recognizers on the same images.

We have also administered tests on 12 voluntary (graduate students) in our department. The same set of 10 handwritten word image was tested on all subjects. The images were chosen randomly from images that are not recognized by our recognizers.

Results and Discussion. The corresponding accuracy rates for recognizers are shown in Table 1. Figure 31 illustrates some of the word images that are difficult for any current computer recognition techniques even when presented with a small lexicon of words.

By examining the set of recognized images we have found that the majority of them are deformed by only one transformation, such as blur, spread, or wave, which makes these transformations alone inefficient. In the recognized set there were just very few images with background noise, such as salt-and-pepper noise, and in all cases the character image pixels were more prominently than the noisy pixels so that they have been distinguished easier.

Generally we observe that adding background noise is the most powerful transformation because it is easily reproducible and the accuracy of the system drops significantly 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 the recognizers. The other transformations that we have considered (blur, spread, wave, median filter, etc) are efficient when applied in groups.

As expected, the set of city names did not pose any problem for humans given the context (Table 2). The protocol of study involving human participants was reviewed and approved by the Social and Behavioral Sciences Institutional Review Board at the University at Buffalo.

<table>
<thead>
<tr>
<th>Truth Word</th>
<th>Test Images</th>
<th>Transformations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waterville</td>
<td>383</td>
<td>9.26%</td>
</tr>
<tr>
<td>Accuscript</td>
<td>182</td>
<td>4.41%</td>
</tr>
</tbody>
</table>

Table 1. The accuracy of handwriting recognizers for the first experiment on US City names images.

<table>
<thead>
<tr>
<th>Test Images</th>
<th>Humans</th>
<th>WMR</th>
<th>Accuscript</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>82%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table 2. The accuracy of human readers for the first experiment on US City names images.

In order to utilize the lexicon level challenge, we also 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 32 we show what happens when the lexicon (of size 10) is simulated to increase the confusion (density).
Fig. 32. Handwritten CAPTCHAs using lexicons with similar entries. In order to show the effect of this method without using image transformation, the images were not deformed. Even in this situation, the recognizers did not produce the correct results as top choice.

5.2 Various Transformations on Nonsense Words

Method

Procedure. 3,000 random nonsense word images were generated randomly by combination of characters, with one word per image and a random word length between 5 and 10 (Figure 33). The characters were chosen randomly from a database of over 20,000 characters, which were previously extracted from city name images (Figure 6). We have run WMR and Accuscript recognizers on these images. We have also used a subset of 100 images that recognizers cannot read correctly and tested them on human subjects.

Results and Discussion. A majority of these synthetic handwritten word images are readable by humans. However, human subjects confused the following characters: “g” vs. “q”, “r” vs. “n”, and “e” vs. “e”. Perhaps using real word images can help eliminate some of the errors humans have done in the case of ambiguous characters. The overall error rate of 20% for humans versus 100% for all recognizers shows the superiority of human abilities when reading handwritten text images even without the aid of context. Recognizers’ accuracy is presented in Table 3.

Table 3. The accuracy of handwriting recognizers for random non-sense words.

<table>
<thead>
<tr>
<th>Test Images</th>
<th>WMR</th>
<th>Accuscript</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random non-sense words</td>
<td>12.04%</td>
<td>3.19%</td>
</tr>
</tbody>
</table>

5.3 Transformations Related to Gestalt and Geon principles

Another set of experiments deals with the methods described in association with the Gestalt laws of perception.

Method
Table 4. The accuracy of WMR for all image transformations.

<table>
<thead>
<tr>
<th>Transformation</th>
<th>WMR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L = 4,000$</td>
</tr>
<tr>
<td>All Transformations</td>
<td>12.69%</td>
</tr>
<tr>
<td>Empty Letters</td>
<td>0.89%</td>
</tr>
<tr>
<td>Small Fragmentation</td>
<td>0.00%</td>
</tr>
<tr>
<td>High Fragmentation</td>
<td>0.48%</td>
</tr>
<tr>
<td>Displacement</td>
<td>19.75%</td>
</tr>
<tr>
<td>Mosaic</td>
<td>14.34%</td>
</tr>
<tr>
<td>Jaws/Arcs</td>
<td>3.12%</td>
</tr>
<tr>
<td>Occlusion by circles</td>
<td>33.93%</td>
</tr>
<tr>
<td>Occlusion by waves</td>
<td>15.43%</td>
</tr>
<tr>
<td>Black Waves</td>
<td>16.36%</td>
</tr>
<tr>
<td>Vertical Overlap</td>
<td>27.88%</td>
</tr>
<tr>
<td>Horizontal Overlap (Small)</td>
<td>21.33%</td>
</tr>
<tr>
<td>Horizontal Overlap (Large)</td>
<td>12.93%</td>
</tr>
<tr>
<td>Overlap Different Words</td>
<td>3.80%</td>
</tr>
<tr>
<td>Flip-Flop</td>
<td>0.46%</td>
</tr>
</tbody>
</table>

Table 5. The accuracy of Accuscript recognizer for all image transformations.

<table>
<thead>
<tr>
<th>Transformation</th>
<th>Accuscript</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L = 4,000$</td>
</tr>
<tr>
<td>All Transformations</td>
<td>3.60%</td>
</tr>
<tr>
<td>Empty Letters</td>
<td>0.06%</td>
</tr>
<tr>
<td>Small Fragmentation</td>
<td>0.18%</td>
</tr>
<tr>
<td>High Fragmentation</td>
<td>0.00%</td>
</tr>
<tr>
<td>Displacement</td>
<td>8.84%</td>
</tr>
<tr>
<td>Mosaic</td>
<td>8.99%</td>
</tr>
<tr>
<td>Jaws/Arcs</td>
<td>3.58%</td>
</tr>
<tr>
<td>Occlusion by circles</td>
<td>32.34%</td>
</tr>
<tr>
<td>Occlusion by waves</td>
<td>10.56%</td>
</tr>
<tr>
<td>Black waves</td>
<td>1.37%</td>
</tr>
<tr>
<td>Vertical Overlap</td>
<td>12.64%</td>
</tr>
<tr>
<td>Horizontal Overlap (Small)</td>
<td>2.93%</td>
</tr>
<tr>
<td>Horizontal Overlap (Large)</td>
<td>2.42%</td>
</tr>
<tr>
<td>Overlap Different Words</td>
<td>4.43%</td>
</tr>
<tr>
<td>Flip-Flop</td>
<td>0.70%</td>
</tr>
</tbody>
</table>

We tried various displacements of overlap in the vertical and horizontal direction. We noticed that by increasing the displacement in the horizontal direction, the error rate for machines increases but it also poses problems for humans since visual segmentation becomes difficult. The flip-flop transform is not relevant due to the nature of our recognizers. The accuracy for these cases is very small with our test recognizers, but we do not count these results yet since our current recognizers are not trained on these types of images. Some efficient methods based on our results are: duplicate the word along the vertical axis (vertical overlaps) or add black occlusions such as waves, lines, arcs, or any stroke that can be confused with parts of a character. While computers have major difficulties in recognizing them, humans have little difficulty for such images. The Gestalt law of differentiating between background and foreground holds in this case, and humans easily connect the characters that are overlapped and are able to eliminate the background noise.

For methods that hide parts of images, we experimented with several ways of placing the occlusions (middle of image, or determine the part of the image based on where the majority of black pixels are present) and also varying the size of occlusions (wave amplitude, wavelength, circle radius, or number of circles per image). In our tests, we were concerned with the overall results for each kind of transformation, getting a sense of which ones work based on the Gestalt assumptions and humans results, and further varying the parameters for each transformation. We have considered image complexity (i.e., perimetric complexity and image density) as a factor that can be manipulated through the transformation parameters to achieve the maximum gap between human and machine accuracy.

Based on our results, the most efficient transformations are letter fragmentations (i.e., small and high fragmentation in Table 4, Table 5, and Table 6). The Gestalt laws of closure and continuity hold strongly in this case, and humans easily fill the gaps or continue the characters that are broken apart. One might expect low accuracy for handwriting recognizers when jagged strokes are added to the original images. Jagged strokes and arcs, as well as regularly spaced and sized graphics, or short drawings, can be misclassified as text and lead to segmentation failure. For the splitting transforms, we used cuts in the middle, in the lower part and upper part of the word. Generally they have similar effect on both human and machine recognition.

Due to the randomness of some parameters in our transformations, we may end up with images with just small areas affected by occlusions and mostly covering parts of the background. Most of the images correctly recognized by the recognizers fall in this category. Fig-
Figure 34 shows several images that were correctly recognized but where the transformation chosen did not modify the original image significantly and therefore did not add sufficient challenge to the recognition task. Through a better process of parameter selection, we can avoid most of these situations. On the other hand, the recognizers have difficulty with fairly clean images with well chosen parameters for transformations (Figure 7).

![Image of handwritten text](image)

**Fig. 34.** Examples of handwritten images that were recognized by one of our testing recognizers. The truth words are: Pleasantville, Amherst, Silver Springs.

The tests on human subjects suggest that human performance depends on context, and prior knowledge of the word provides the greatest advantage to human readers. Therefore, memory and word familiarity (Gestalt principles) have proven to be useful cues for humans. In general, if the original handwritten sample is clean, after deformation it does not create problems for humans, but does for machines. However, if the original sample contains noise or is poorly written, then even the original image causes problems to both human and computer, even before deformation. We have noticed that most of the human errors come from nonsensical original images rather than difficulties with the deformations applied to those images. For occlusion by circles, we can explain the lower accuracy results using the fact that some of the omissions perhaps covered a large part of a letter or the entire letter. For larger space between the words that overlap in the horizontal direction, we can explain it by saying that it might have caused confusing words, as shown in a previous example (Figure 17). The human results are presented in Table 7. The gap in the ability in recognizing handwritten text between humans and computers is illustrated in Figure 35.

<table>
<thead>
<tr>
<th>Transformation</th>
<th># of Images</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Transformations</td>
<td>1069</td>
<td>76.98%</td>
</tr>
<tr>
<td>Empty Letters</td>
<td>89</td>
<td>82.02%</td>
</tr>
<tr>
<td>Small Fragmentation</td>
<td>88</td>
<td>73.86%</td>
</tr>
<tr>
<td>High Fragmentation</td>
<td>90</td>
<td>74.44%</td>
</tr>
<tr>
<td>Displacement</td>
<td>89</td>
<td>78.65%</td>
</tr>
<tr>
<td>Mosaic</td>
<td>90</td>
<td>74.44%</td>
</tr>
<tr>
<td>Jaws/Arcs</td>
<td>89</td>
<td>71.19%</td>
</tr>
<tr>
<td>Occlusion by circles</td>
<td>90</td>
<td>67.78%</td>
</tr>
<tr>
<td>Occlusion by waves</td>
<td>87</td>
<td>80.46%</td>
</tr>
<tr>
<td>Black Waves</td>
<td>90</td>
<td>80.06%</td>
</tr>
<tr>
<td>Vertical Overlap</td>
<td>88</td>
<td>87.50%</td>
</tr>
<tr>
<td>Horizontal Overlap (Small)</td>
<td>90</td>
<td>76.67%</td>
</tr>
<tr>
<td>Horizontal Overlap (Large)</td>
<td>89</td>
<td>65.17%</td>
</tr>
</tbody>
</table>

**Table 7.** The accuracy of human readers for all image transformations.

![Graph](image)

**Fig. 35.** Ability gap in recognizing handwritten text between humans and computers per type of transformation (empty letters, fragmentation small/high, displacement, mosaic, jaws, occlusion by circles, occlusion by waves, waves, vertical overlap, horizontal overlap small and large).

5.4 Various Transformations on Synthetic Words

**Method.** We have also automatically generated 300 synthetic handwriting samples corresponding to US city names, US states, and world wide countries, using the method described in [22], and applied various transformations to make them unreadable by automatic computer programs. We have applied noise, extra strokes, lines, grids, arcs, circles, background, and occlusions, deleting ligatures, using empty letters, broken letters, and fragmentation, using the methods previously described. Several examples of synthetic word images, deformed or not, are presented in Figure 36. We have tested our recognizers on the set of 300 transformed synthetic images. We have also administered tests on human subjects and compared the human abilities in recognition on a set of handwritten US city name images available from postal applications to the set that contains 79 synthetic US city name, state, or country name images automatically generated by our programs.

**Results and Discussion.** Similar high human accuracies in recognition for both sets have been observed which guarantees that synthetic handwritten images do not pose problems to the user when used online. The accuracies achieved by the state-of-the-art handwriting recognizers for the synthetic word images was as low as for the real handwritten samples, and for a set of 300 automatically generated synthetic images shown in Table 8.
ments to be a CAPTCHA: i) there is little risk of image repetition since the image generation is completely automated, the words, images and distortions are chosen at random; ii) the transformed images cannot be easily normalized or rendered noise-free by present computer programs (i.e., handwriting recognizers, OCRs), although the original handwritten images are open public knowledge; iii) the deformed images do not pose problems to humans, whereas the handwritten CAPTCHA images remain unbroken by state-of-the-art recognizers throughout our tests. Experimental results on three handwritten word recognizers have shown the gap in the ability between humans and computers in handwriting recognition. We also conducted user studies and human surveys on handwritten CAPTCHAs, since human users are an important part of building a practical security system, and the analysis of the results correlates strongly with our hypothesis.

We have also administered experiments to determine how robust is our algorithm for image transformation and degradation, or how easily an image deformation can be reversed and the original image retrieved. Although the testing handwriting recognizers use general image processing techniques in the preprocessing phase, we have considered developing more sophisticated methods to attack Handwritten CAPTCHA, using preprocessing techniques for de-noising, eliminating small degradations and gaps, line removal, etc. Experimental studies on the effect of Gestalt laws-based transformation on words have been conducted on both humans and computers. The experiments show significant benefits in using handwritten CAPTCHA, as opposed to less-efficient machine-printed CAPTCHAs, and impractical CAPTCHAs based on facial features or images of objects.

Experimental results support the importance of cognitive factors involved in human visual recognition so we explain the results by identifying the role of Gestalt principles, geon theory and prior knowledge of the context. Humans have an innate ability to recognize writing in any form, whether it is machine-printed text or handwriting, also to distinguish between text and graphics. However, this cannot be said about machine recognition. There are many reasons why machine recognition of handwriting is more difficult than machine-printed text, for instance segmentation problems, character confusion, unconstrained writing styles and inconsistent space between letters and words, etc. Given the success of our empirical study conducted on humans and machines recognition and comparing with other CAPTCHA approaches, in particular machine-printed CAPTCHA, we can conclude that the handwritten HIP system that we propose is more efficient than the currently used HIPs for cyber security applications.

Table 8. The accuracy of handwriting recognizers for synthetic words.

<table>
<thead>
<tr>
<th>Test Images</th>
<th>WMR</th>
<th>Accuscript</th>
<th>CMR</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>1.00%</td>
<td>0.7%</td>
<td>0.3%</td>
</tr>
</tbody>
</table>

6 Conclusions

We have presented a handwritten CAPTCHA-based HIP system as a security protocol for Web services and evaluated the performance of our challenge generation algorithm. The norms of CAPTCHA generation dictate that the method of generating these images must be public knowledge giving those who want to break the CAPTCHAs a fair shot. Evaluating our handwritten image challenges reveals that they satisfy all the require-

References


Amalia Rusu Dr. Amalia Rusu is an Assistant Professor in Department of Software Engineering, School of Engineering, at Fairfield University. She received her Ph.D. in Computer Science and Engineering from University at Buffalo (SUNY) in 2007. Her main research focus is on Handwriting Recognition and CAPTCHAAs.

Achint Thomas Achint Thomas completed his MS in Computer Science and Engineering from the University at Buffalo (SUNY Buffalo) in 2007 and is currently a PhD candidate there. His areas of interest include HIP’s, CAPTCHA’s, pattern recognition and biometrics.

Venu Govindaraju Dr. Venu Govindaraju is a Professor of Computer Science and Engineering at the University at Buffalo (SUNY Buffalo). He received his B-Tech (Honors) from the Indian Institute of Technology (IIT), Kharagpur, India in 1986, and his Ph.D. from SUNY Buffalo in 1992. His is also the Founding Director of the Center for Unified Biometrics and Sensors (CUBS) and the Associate Director of
the Center of Excellence for Document Analysis and Research (CEDAR), both at SUNY Buffalo.