Machine Learning Approaches to CAPTCHA Recognition Requiring Minimal Image Processing

Abstract

This study focuses on a machine learning approach to the CAPTCHA recognition problem that requires minimal preprocessing of inputs. Both the feed-forward neural nets and the self-organizing maps used in this study took the raw image pixels as input with only simple segmentation and translation performed in advance. The models were trained and tested on four and five letter CAPTCHAs using either letters A-G or A-Z, with both models achieving greater than 90% accuracy on four letter CAPTCHAs of letters A-G. Accuracy decreased with the increase in CAPTCHA length, a result of increased crowding of the letters that reduced the performance of the segmentation algorithms. These results show that the difficulty in CAPTCHA recognition is not in the OCR, but rather in the image segmentation. Future work will focus on the image segmentation problem, in addition to ensemble techniques to increase the accuracy of the current models.

1. Introduction

Interacting with the world that humans can navigate easily has proven to be a challenge for computers. Image processing, something that humans learn before they are even aware that they are learning, has proven especially challenging. A good solution to the image recognition problem would allow us to take the very human strength of pattern recognition and supplement it with the strengths of computers: speed and consistency. Such consistency of analysis is extremely important, whether searching a database of terrorists to compare against a photograph snapped in an airport, verifying that a signature on a receipt truly belongs to the cardholder, or translating a stylus stroke into a character on a tablet computer. In the most extreme cases good automated image recognition can save lives; in the least pressing cases it significantly improves the user experience.

In order to process images, algorithms must be sufficiently robust–an "A" must return the same result regardless of font or rotation; a database query using a picture of a smiling woman must also return images of that woman frowning. Machine learning is a natural choice for this problem, as it allows the model to derive the important features of an image through experience, similar to the way humans learn to recognize images. Artificial neural networks especially have been a popular approach to image recognition, and they have enjoyed a significant amount of success in this area. A simple, three-layer neural network is able to positively identify young corn plants in a field with a 90-100% success rate (Yang et al., 2000), and a similar network can match a sample Chinese character to one of 500 in a database with a success rate of 90%. Artificial neural nets have also been used extensively to attempt to translate from handwritten text to digital characters (Faaborg, 2002) (Aroakar, 2005). This is by no means the only approach that has been applied to the problem. Other machine learning techniques, such as self-organizing maps, while less common, have also been used to address image recognition. Self-organizing maps in particular have the advantage of requiring far fewer learning iterations before converging on a solution than a neural net would require. (Ma, 2007)

In spite of these strides made toward a solution to the challenge of image recognition, it remains such a difficult problem for computers that it is commonly used by websites to prevent malicious computer programs from posting spam or otherwise behaving nefariously. CAPTCHAs (Completely Automated Public Turing Tests to tell Computers and Humans Apart), are the distorted texts users must decipher before gaining admittance to many web services. While one might think that they are of little interest to anyone other than spammers, they in fact present a challenging domain that offers unique advantages over straightforward op-

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

tical character recognition. Because CAPTCHAs are more difficult to read than undistorted text, solutions that excel at solving CAPTCHAs translate well to other contemporary challenges in machine learning, such as handwriting and damaged text recognition. Additionally, CAPTCHA generators allow for near-infinite data sets to be created on demand, reducing difficulties introduced by overfitting and eliminating the necessity to store large quantities of data in memory.

We believe that machine learning is most useful when applied in such a way that it does not require difficult or computationally expensive preprocessing of inputs, and it is this belief that defined our approach to the CAPTCHA recognition problem. In this paper, we present two different systems, one using feed-forward neural nets and the other using self-organizing maps, each taking the raw image pixels as input with only simple segmentation and translation performed in advance.

2. Methods

2.1. JCaptcha

The CAPTCHAs for this study were generated using JCaptcha, an open-source CAPTCHA generation framework written in Java. Highly customizable, JCaptcha allows the user to specify a set of characters and a set of fonts from which to generate CAPTCHAs, in addition to allowing specification of length, color, and degree of distortion. In this study, all CAPTCHAs were generated using random fonts. CAPTCHAs of both four and five letters were used, composed of letters from the set A-G or A-Z.

The speed at which JCaptcha was able to produce new CAPTCHAs allowed us to avoid producing and storing large test-sets. Only one CAPTCHA is ever in memory at any given time, and when necessary to store information about the CAPTCHAs, only running totals were used: this allows our implementations to run using essentially constant memory. Further, because of the virtually unlimited test-set, we were able to completely avoid the issue of overfitting.

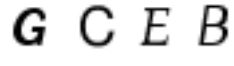

Figure 1. Sample Captcha

2.2. Neural Net

The neural net utilized in this project is a standard fully-connected feed-forward neural net.(Russell & Norvig, 2003) In this paper, two neural net topologies were used: the first, for identifying CAPTCHAs composed of letters A-G, consists of a 400 node input layer, a 1000 node hidden layer, and a seven node output layer (one output for each character in the distribution). (See Figure 2.) The second, for identifying CAPTCHAs composed of letters A-Z, has input and hidden layers of the same size as the aforementioned neural net, but has a 26 node output layer. Sigmoid activation functions were used for the nodes in the hidden layers; the output nodes were linearly activated. As in (Yang et al., 2000), hidden layer sizes were tested extensively to determine which topology yielded the greatest accuracy, but after reaching a certain number of nodes in the hidden layer, accuracy remained largely unaffected.

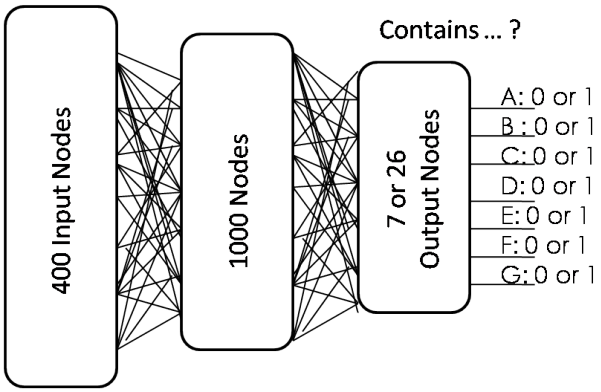

Figure 2. Neural Net Structure

2.3. Self-Organizing Map

Where the neural net requires only one phase for training, the self-organizing map requires two. During the first phase, a predetermined number of "buckets" are initialized to image segments from random CAPTCHAs. Then, segments of CAPTCHAs are compared with each buckets according. The bucket most closely matches that CAPTCHA segment is adjusted so as to be more similar the input segment. The adjustment is done as a bitwise operation on the sparse matrix representing the pixels of the images, and takes the form

$$B = \frac{(B + \alpha I)}{1 + \alpha} \tag{1}$$

where B is the matrix representing the bucket image and I is the matrix representing the input image.This is

repeated until the buckets converge, which may take as few as several hundred to several thousand iterations.

The second phase of training the self-organizing map is an observation phase. Each CAPTCHA the map is given as input is sorted into one of the buckets: the number of CAPTCHAs in a particular bucket containing a particular letter divided by the total number of CAPTCHAs in the bucket is the probability that any CAPTCHA sorted into that bucket will contain that letter.

Figure 3. Sample Buckets
shows

2.4. Image Processing

The only image processing the images underwent before being input into a learning algorithm is segmentation and translation. After each image is segmented, the segments' centers of mass are shifted so that all segments are centered at the same coordinate. This is essential to the convergence of the neural net, and exponentially reduces the size requirement of the self-organizing map.

2.4.1. Segmentation

The CAPTCHAs in the experiments presented in this study were segmented using K-Means clustering. However, prior to the implementation of K-Means, an overlapping segmentation algorithm was used, in which the image was broken into uniformly-sized blocks that overlapped each other by 50% in each direction. While this paper does not present results obtained from training the models on CAPTCHA segments obtained using the overlapping segmentation method, it is of consequence because it is the only segmentation method we have implemented to date that does not assume knowledge of the number of letters in a CAPTCHA. Future work will devote investigation to capitalizing on this advantage.

Additional segmentation methods were implemented to compare against K-Means. Even-width segmentation simply divides the total width of the CAPTCHA by the number of letters present in the CAPTCHA, and segments the CAPTCHA according to that result. Whitespace segmentation iterates through the image column by column, looking for columns entirely composed of background pixels, splitting the image when it encounters such a column. For an example of all of these segmentation methods, see Figure 4. Note that while even-width, whitespace, and K-Means segmentation all perform admirably on well-spaced text, their performance is significantly worse if the characters are crowded.

3. Experiments

3.1. Experiment 1: Four Letter CAPTCHAs Using Letters A-G

The neural net and the self-organizing map described above were trained for 40,000 iterations using four letter CAPTCHAs composed of capital letters from A-G. The images were segmented using K-Means.

3.2. Experiment 2: Four Letter CAPTCHAs Using Letters A-Z

The neural net and the self-organizing map were trained for 84,000 iterations using four letter CAPTCHAs composed of capital letters from A-Z. The images were segmented using K-Means. This is a significantly more difficult problem than Experiment 1, and therefore convergence for the neural net took significantly longer.

3.3. Experiment 3: Five Letter CAPTCHAs Using Letters A-G

The neural net and the self-organizing map were trained for 84,000 iterations using five letter CAPTCHAs composed of capital letters from A-Z. The images were segmented using K-Means. This, again, is a significantly more difficult problem than Experiment 1, and therefore convergence for the neural net took significantly longer. However, despite it being a more difficult problem that Experiment 2, training for longer that 84,000 iterations did not result in much improvement.

4. Results

The accuracy for identifying CAPTCHAs of length 4 and letters A-G for both the neural net and the SOM exceeded 90%, with the neural net's accuracy reaching as high as 95%. As one would expect, increasing the difficulty of the problem by expanding the letter

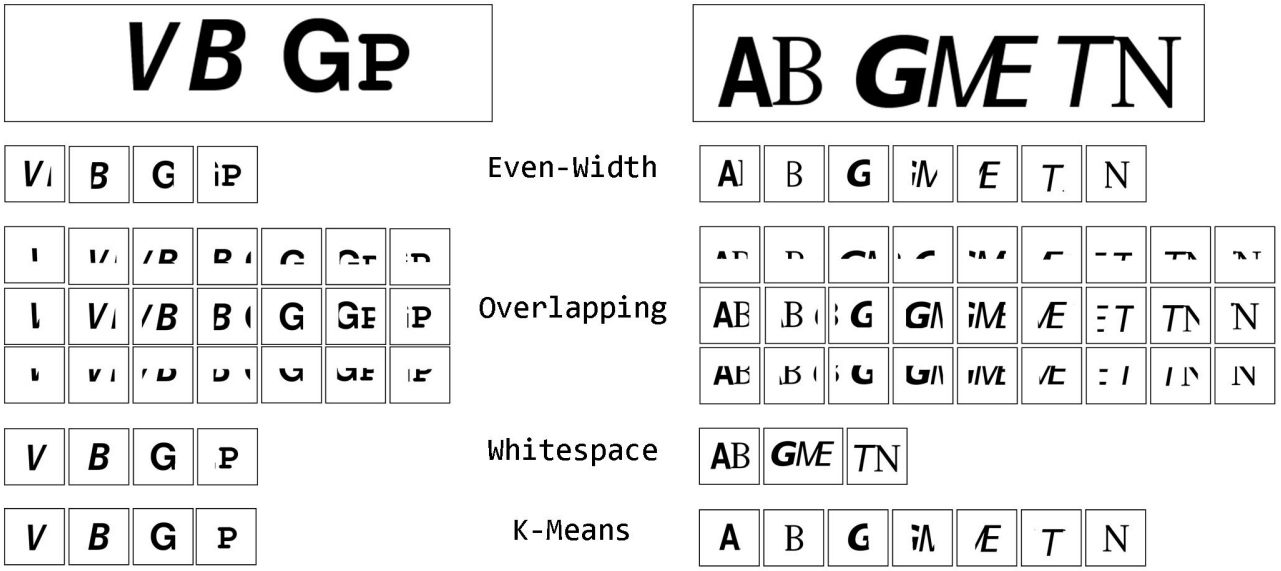

Figure 4. Image Segmentation Comparison

domain from seven letters to twenty-six letters had a negative impact on the accuracy. However, simply increasing the number of letters the models were asked to identify did not decrease the accuracy as much as increasing the number of letters in the CAPTCHA while using the expanded letter domain. Both the neural net and the SOM achieved less than 30% accuracy on the longer CAPTCHA.

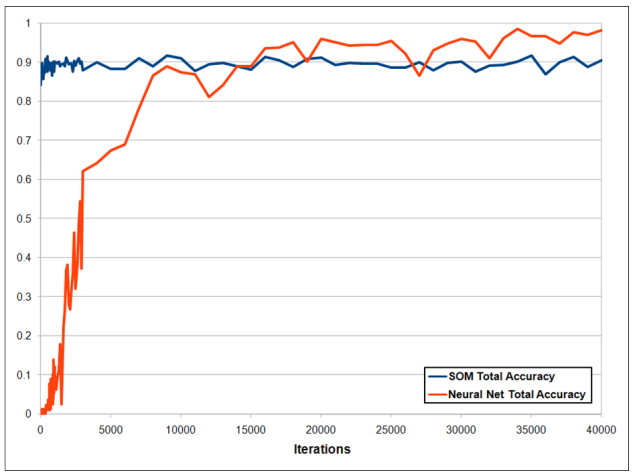

Figure 5. Experiment 1: Neural Net vs SOM in Identifying Four Letter CAPTCHAs Using Letters A-G

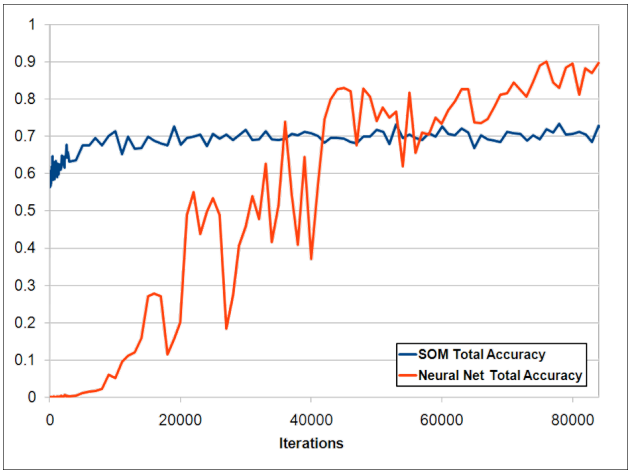

Figure 6. Experiment 2: Neural Net vs SOM in Identifying Four Letter CAPTCHAs Using Letters A-Z

5. Discussion

As highlighted by the above results, the accuracy of the two methods, which is high on four letter CAPTCHAs using letters A-G, decreases as the length of the CAPTCHA increases. This is due primarily to a decline in the quality of results produced by the segmentation algorithms, which perform best on well-spaced text. As the number of letters in the CAPTCHA increases, the space between the letters

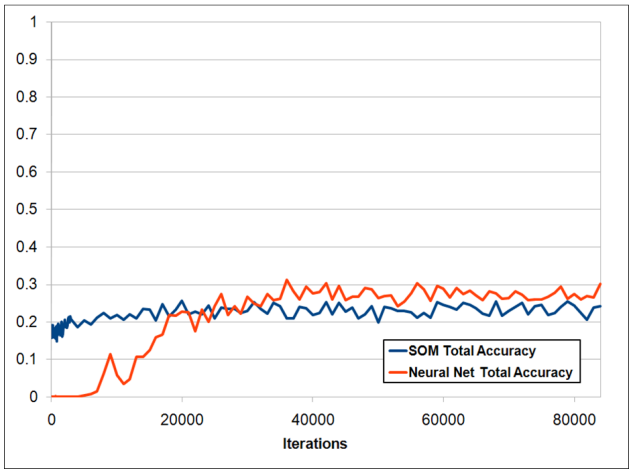

Figure 7. Experiment 3: Neural Net vs SOM in Identifying Five Letter CAPTCHAs Using Letters A-G

decreases. This crowding makes automatic segmentation less successful, as seen in the Figure 4. The four-letter CAPTCHA segments perfectly using both K-Means and whitespace segmentation, and nearly perfectly using even-width segmentation. The comparative quality of the segmentation performed by the algorithms drops of steeply in the seven-letter case, however. This, in combination with a preliminary neural net that could recognize a single character with 99% accuracy, shows that the true difficulty in CAPTCHA recognition lies not in optical character recognition, but rather in segmentation.

6. Future Work

Future work will therefore focus primarily on the segmentation problem. Ensemble learning methods are a promising technique to address segmentation: such an approach reduces or eliminates the need to create complicated segmentation models and allows us to take advantage of the different approaches already available. Such an ensemble-learning segmentation system would include neural nets and SOMs trained to recognize whether an image chunk is a letter with some percentage of confidence. Then, one could begin at the left and feed the models increasingly larger slices of the image, waiting for a maximum confidence. Other work in segmentation should focus on removing the primary assumption of our models, which, with the exception of overlapping segmentation, require that the number of letters in the CAPTCHA be known.

Other future work will focus on increasing the true positive rate of the current models. Ensemble learning is also promising here: the current systems are incorrect about different kinds of input. Preliminary results suggest that accuracy of the models described above could be increased to 70% on 5 letter CAPTCHAs using letters A-Z simply by adding the probability output by the neural net and the SOM.

Finally, we would like to apply these techniques to problems with immediate uses beyond building unstoppable spam bots, such as handwriting recognition and digitizing damaged text.

7. Conclusions

The results of our experimentation demonstrate that CAPTCHAs can be solved to a high degree of accuracy with minimal preprocessing beyond segmentation and translation. However, in demonstrating this it became clear that optical recognition is a less difficult portion of the problem than previously thought: single characters can be identified with 99% accuracy, and well-spaced CAPTCHAs can be identified with as great as 95% accuracy. Instead, the difficulty in recognizing CAPTCHAs lies almost entirely in the problem of separating one character from another. Future work focusing on ensemble learning techniques to segment crowded CAPTCHAs will significantly enhance the ability of our models to accurately identify CAPTCHAs and other text.

References

Aroakar, S. (2005). Visual character recognition using artificial neural networks.

Faaborg, A. J. (2002). *Using neural networks to create an adaptive charactre recognition system.* Doctoral dissertation, Cornell University, Ithaca, NY.

Gonzaga, A., Marin, A., Silva, E. A., Bertoni, F. C., Costa, K. A., & Albeuguerque, L. A. (2002). Neutral facial image recognition using parallel hopfield neural networks.

Ma, H. (2007). Off-line Chinese-based signature verification using a threshold Self-Organizing Map. *Journal of the Chinese Institute of Industrial Engineers*, *24*, 225–235.

Rizon, M., Hashim, M. F., Saad, P., Yaacob, S., Mamat, M. R., Shakaff, A. Y. M., Sadd, A. R., Desa, H., & Karthigayan, M. (2006). Face recognition using eigenfaces and neural networks. *American Journal of Applied Sciences*, *2*, 1872–1875.

Russell, S., & Norvig, P. (2003). *Artificial intelligence: A modern approach.*

Vuori, V. (2002). Clustering writing styles with a self-organizing map. *In Proc. of the 8th IWFHR* (pp. 345–350).

Yang, C.-C., Prasher, S., Landry, J.-A., Ramaswamy, H., & Ditommaso, A. (2000). Application of artificial neural networks in image recognition and classification of crop and weeds. *Canadian Agricultural Engineering*, *42*, 147–152.

Yang, Z., & Laaksonen, J. (2005). Interactive retrieval in facial image database using self-organizing maps. *IAPR Conference on Machine Vision Applications*, 112–115.