J. Hlaváčová (Ed.): ITAT 2017 Proceedings, pp. 93–99
CEUR Workshop Proceedings Vol. 1885, ISSN 1613-0073, © 2017 M. Kopp, M. Nikl, M. Holeňa

Breaking CAPTCHAs with Convolutional Neural Networks

Martin Kopp[1,2], Matěj Nikl[1], and Martin Holeňa[1,3]

[1] Faculty of Information Technology, Czech Technical University in Prague
Thákurova 9, 160 00 Prague
[2] Cisco Systems, Cognitive Research Team in Prague
[3] Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2, 182 07 Prague

Abstract: This paper studies reverse Turing tests to distinguish humans and computers, called CAPTCHA. Contrary to classical Turing tests, in this case the judge is not a human but a computer. The main purpose of such tests is securing user logins against the dictionary or brute force password guessing, avoiding automated usage of various services, preventing bots from spamming on forums and many others.

Typical approaches to solving text-based CAPTCHA automatically are based on a scheme specific pipeline containing hand-designed pre-processing, denoising, segmentation, post processing and optical character recognition. Only the last part, optical character recognition, is usually based on some machine learning algorithm. We present an approach using neural networks and a simple clustering algorithm that consists of only two steps, character localisation and recognition. We tested our approach on 11 different schemes selected to present very diverse security features. We experimentally show that using convolutional neural networks is superior to multi-layered perceptrons.

Keywords: CAPTCHA, convolutional neural networks, network security, optical character recognition

1 Introduction

The acronym CAPTCHA[1] stands for Completely Automated Public Turing test to tell Computers and Humans Apart, and was coined in 2003 by von Ahn et al [20]. The fundamental idea is to use hard AI problems easily solved by most human, but unfeasible for current computer programs. Captcha is widely used to distinguish the human users from computer bots and automated scripts. Nowadays, it is an established security mechanism to prevent automated posting on the internet forums, voting in online polls, downloading files in large amounts and many other abusive usage of web services.

There are many available captcha schemes ranging from classical text-based over image-based to many unusual custom designed solutions, e.g. [3, 4]. Because most of the older schemes have already been proven vulnerable to attacks and thus found unsafe [7, 19] new schemes are being invented. Despite that trend, there are still many places where the classical text-based schemes are used as the main or at least as a fallback solution. For example, Google uses the text-based schemes when you fail in their newer image-based ones.

This paper is focused on automatic character recognition from multiple text-based CAPTCHA schemes using artificial neural networks (ANNs) and clustering. The ultimate goal is to take a captcha challenge as an input while outputting transcription of the text presented in the challenge. Contrary to the most prior art, our approach is general and can solve multiple schemes without modification of any part of the algorithm.

The experimental part compares the performance of the shallow (only one hidden layer) and deep (multiple hidden layers) ANNs and shows the benefits of using a convolutional neural networks (CNNs) multi-layered perceptrons (MLP).

The rest of this paper is organised as follows. The related work is briefly reviewed in the next section. Section 3 surveys the current captcha solutions. Section 4 presents our approach to breaking captcha challenges. The experimental evaluation is summarised in Section 5 followed by the conclusion.

2 Related Work

Most papers about breaking captcha heavily focus on one particular scheme. As an example may serve [11] with preprocessing, text-alignment and everything else fitted for the scheme reCapthca 2011. To our knowledge, the most general approach was presented in [7]. This approach is based on an effective selection of the best segmentation cuts and presenting them to k-nn classifier. It was tested on many up-to-date text-based schemes with better results than specialized solutions.

The most recent approaches use neural networks [19]. The results are still not that impressive as the previous approaches, but the neural-net-based approaches improve very quickly. Our work is based on CNN, being motivated by their success in pattern recognition, e.g. [6, 14].

The Microsoft researcher Chellapilla who intensively studied human interaction proofs stated that, depending on the cost of the attack, automated scripts should not be more successful than 1 in 10 000 attempts, while human success rate should approach 90% [10]. It is generally considered a too ambitious goal, after the publication of [8] showing

[1] The acronym captcha will be written in lowercase for better readability.

the human success rate in completing captcha challenges and [9] showing that random guesses can be successful. Consequently, a captcha is considered compromised when the attacker success rate surpasses 1%.

3 Captcha Schemes Survey

This section surveys the currently available captcha schemes and challenges they present.

3.1 Text-Based

The first ever use of captcha was in 1997 by the software company Alta-Vista, which sought a way to prevent automated submissions to their search-engine. It was a simple text-based test which was sufficient for that time, but it was quickly proven ineffective when the computer character recognition success rates improved. The most commonly used techniques to prevent automatic recognition can be divided into two groups called anti-recognition features and anti-segmentation features.

The anti-recognition features such as different sizes and fonts of characters or rotation was a straightforward first step to the more sophisticated captcha schemes. All those features are well accepted by humans, as we learn several shapes of letters since childhood, e.g. handwritten alphabet, small letters, capitals. The effective way of reducing the classifier accuracy is a distortion. Distortion is a technique in which ripples and warp are added to the image. But excessive distortion can make it very difficult even for humans and thus the usage of this feature slowly vanishes being replaced by anti-segmentation features.

The anti-segmentation features are not designed to complicate a single character recognition but instead they try to make the automated segmentation of the captcha image unmanageable. The first two features used for this purpose were added noise and confusing background. But it showed up that both of them are bigger obstacle for humans than for computers and therefore, they where replace by occlusion lines, an example can be seen in Figure 1. The most recent anti-segmentation feature is called negative kerning. It means that the neighbouring characters are moved so close to each other that they can eventually overlap. It showed up that humans are still able to read the overlapping text with only a small error rate, but for computers it is almost impossible to find a right segmentation.

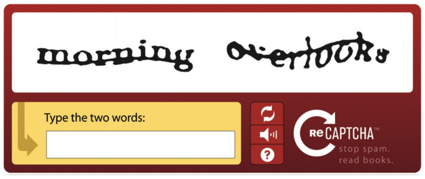


Figure 1: Older Google reCaptcha with the occlusion line.

3.2 Audio-Based

From the beginning, the adoption of captcha schemes was problematic. Users were annoyed with captchas that were hard to solve and had to try multiple times. The people affected the most were those with visual impairments or various reading disorders such as dyslexia. Soon, an alternative emerged in the form of audio captchas. Instead of displaying images, a voice reading letters and digits is played. In order to remain effective and secure, the captcha has to be resistant to automated sound analysis. For this purpose various background noise and sound distortion are added. Generally, this scheme is now a standard alternative option on major websites that use captcha.

3.3 Image-Based

Currently, the most prominent design is image-based captcha. A series of images showing various objects is presented to the user and the task is to select the images with a topic given by a keyword or by an example image. For example the user is shown a series of images of various landscapes and is asked to select those with trees, like in Figure 2. This type of captcha has gained huge popularity especially on touchscreen devices, where tapping the screen is preferable over typing. In the case of Google reCaptcha there are nine images from which the 4 – 6 are the correct answer. In order to successfully complete the challenge a user is allowed to have one wrong answer.

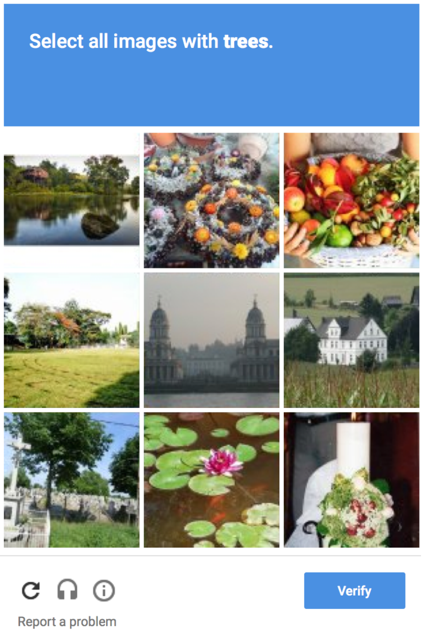
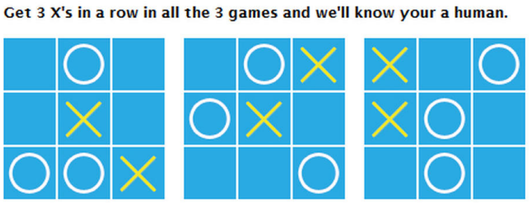


Figure 2: Current Google reCaptcha with image recognition challenge.

Relatively new but fast spreading type of image captcha combines the pattern recognition task presented above with object localisation. Also the number of squares was increased from 9 to 16.

3.4 Other Types

In parallel with the image-based captcha developed by Google and other big players, many alternative schemes appeared. They are different variations of text-based schemes hidden in video instead of distorted image, some simple logical games or puzzles. As an example of an easy to solve logical game we selected the naughts and crosses, Figure 3. All of those got recently dominated by Google's noCaptcha button. It uses browser cookies, user profiles and history to track users behaviour and distinguish real users from bots.



Get 3 X's in a row in all the 3 games and we'll know your a human.

Figure 3: A naughts and crosses game used as a captcha.

4 Our Approach

Our algorithm has two main stages localisation and recognition. The localisation can be further divided into heat map generation and clustering. Consequently, our algorithm consist of three steps:

1. Create a heat map using a sliding window with an ANN, that classifies whether there is a character in the center or not.

2. Use the k-means algorithm to determine the most probable locations of characters from the heat map.

3. Recognize the characters using another specifically trained ANN.

4.1 Heatmap Generation

We decided to use the sliding window technique to localize characters within a CAPTCHA image. This approach is well known in the context of object localization [16]. A sliding window is a rectangular region of fixed width and height that *slides* across an image. Each of those windows serve as an input for a feed-forward ANN with a single output neuron. Its output values are the probability of its input image having a character in the center. Figure 4 shows an example of such heat map. To enable a character localization even at the very edge of an image one can expand each input image with black pixels.

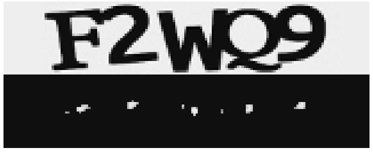
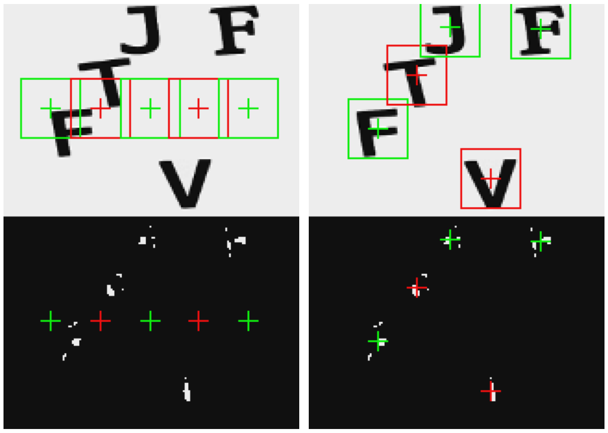


Figure 4: Example of a heat map for a challenge generated by scheme s16.

4.2 Clustering

When a heat map is complete, all points with value greater than 0.5 are added to the list of points to be clustered. As this is still work in progress we simplified the situation by knowing the number of characters within the image in advance and therefore, knowing the correct number of clusters k, we decided to use k-means clustering to determine windows with characters close to their center. But almost an arbitrary clustering algorithm can be used, preferably some, that can determine the correct number of clusters.

The k centroids are initialized uniformly from left to right, vertically in the middle, as this provides a good initial estimation. Figure 5 illustrates the whole idea.



(a) Initial centroids (b) Final centroids

Figure 5: Heatmap clustering on random character locations

4.3 Recognition

Assuming that the character localization part worked well, windows containing characters are now ready to be recognized. This task is known to be easy for computers to solve; in fact, they are even better than humans [10].

Again, a feed-forward ANN is used. This time with an output layer consisting of 36 neurons to estimate the probability distribution over classes: numbers 0–9 and uppercase letters A–Z. Finally, a CAPTCHA transcription is created by writing the recognized characters in the ascending order of their x-axis coordinates.

5 Experimental Evaluation

This section describes the selection of a captcha suite and generation of the labelled database, followed by a detailed description of the artificial neural networks used in our experiments. The last part of this section presents results of the experiments.

5.1 Experimental Set up

Training an ANN usually requires a lot of training examples (in the order of millions in the case of a very deep CNN). It is advised to have at least multiple times the number of all parameters in the network [13]. Manually downloading, cropping and labelling such high number of examples is infeasible. Therefore, we tested three captcha providers with obtainable source code to be able to generate large enough datasets: Secureimage PHP Captcha [5], capchas.net [2] and BotDetect captcha [1]. We selected the last one as it provides the most variable set of schemes.

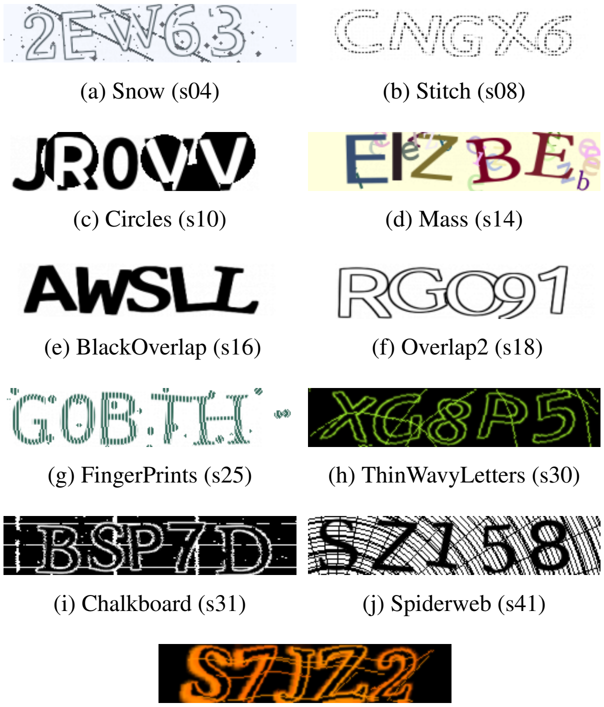
BotDetect CAPTCHA is a paid, up-to-date service used by many government institutions and companies all around the world [1]. They offer a free licence with an access to obfuscated source codes. We selected 11 very diverse schemes out of available 60, see Figure 6 for example of images, and generated 100.000 images cropped to one character for each scheme. The cropping is done to 32x32 pixel windows, which is the size of a sliding window. Cropped images are then used for training of the localization as well as the recognition ANN. The testing set consist of 1000 whole captcha images with 5 characters each.

Schemes display various security features such as random lines and other objects occluding the characters, jagged or translucent character edges and global warp. The scheme s10 - *Circles* stands out with its colour inverting randomly placed circles. This property could make it harder to recognize than others, because the solver needs to account for random parts of characters and their background switching colours.

5.2 Artificial Neural Networks

The perceptron with single hidden layer (SLP), the perceptron with three hidden layers (MLP) and the convolutional neural networks were tested in the localization and recognition. In all ANNs, rectified linear units were used as activation functions.

First experiment tested the influence of the number of hidden neurons of a SLP. The number of hidden neurons used for the localization network was *lns*={15,30,60,90} and the number of neurons for the recognition network was *rns*={30,60,120,180,250}. The results depicted in Figure 7 show the recognition rate for 1000 whole captcha images (all characters have to be correctly recognized) on the scheme s10. The scheme s10 was selected because we consider it the most difficult one.



(a) Snow (s04)

(b) Stitch (s08)

(c) Circles (s10)

(d) Mass (s14)

(e) BlackOverlap (s16)

(f) Overlap2 (s18)

(g) FingerPrints (s25)

(h) ThinWavyLetters (s30)

(i) Chalkboard (s31)

(j) Spiderweb (s41)

(k) MeltingHeat2 (s52)

Figure 6: Schemes generated by the BotDetect captcha

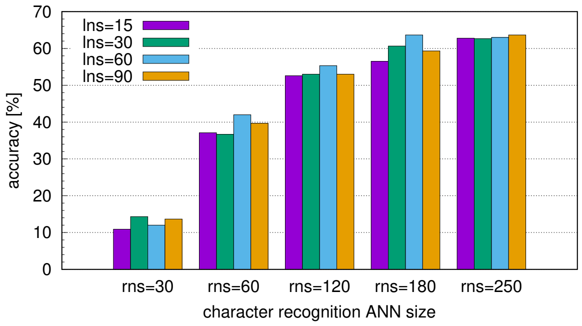


Figure 7: Comparison of SLP recognition rate on the scheme s10, depending on the number of neuron use by the localization network (lns) and the recognition network (rns).

The next experiments was the same but the MLP with three hidden layers was used instead of SLP. Results, depicted in Figure 8, suggest that adding more hidden layers does not improve accuracy of the localization neither of the recognition. Therefore, the rest experiments were done using SLP as it can be trained faster.

Both CNNs architectures resemble the LeNet-5 presented in [17] for handwritten digits recognition. The localization CNN consists of two convolutional layers with six and sixteen 5x5 kernels, each of them followed by the

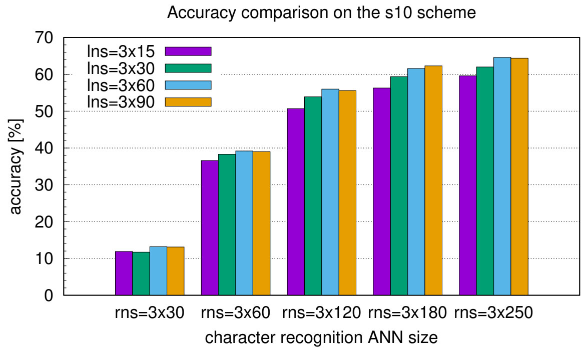


Figure 8: Comparison of MLP recognition rate on the scheme s10, depending on the number of neuron use by the localization network (lns) and the recognition network (rns).

Table 1: Results of the statistical test of Friedman [12] and the correction for simultaneous hypotheses testing by Holm [15] and Shaffer [18]. The rejection thresholds are computed for the family-wise significance level $p = 0.05$ for a single scheme.

Algorithms	p	Holm	Shaffer
SLP+SLP vs. CNN+CNN	7.257e-7	0.0083	0.0083
SLP+SLP vs. SLP+CNN	1.456e-4	0.01	0.0166
CNN+SLP vs. CNN+CNN	5.242e-4	0.0125	0.0166
CNN+SLP vs. SLP+CNN	0.020	0.0166	0.0166
SLP+SLP vs. CNN+SLP	0.137	0.025	0.025
SLP+CNN vs. CNN+CNN	0.247	0.05	0.05

2x2 max pooling layers,and finally, the last layer of the network is a fully connected output layer.

The recognition CNN contains an additional fully-connected layer with 120 neurons right before the output layer as illustrated in Figure 9.

5.3 Results

After choosing the right architectures, we followed by testing the accuracy of captcha transcription on each scheme separately where both training and testing sets were generated by the same scheme. All images in the test set contained 5 characters and only the successful transcription of all of them was accepted as a correct answer. The results, depicted in Figure 10, show appealing performance of all tested configurations. In the most cases it doesn't matter if the localization network was a SLP or a CNN, but the CNN clearly outperforms the SLP in the role of a recognition network. This observation is also confirmed by the statistical test of Friedman [12] with corrections for simultaneous hypothesis testing by Holm[15] and Shaffer [18], see Table 1.

A subsequent experiment tested the accuracy of captcha transcription when training and testing sets consist of im-

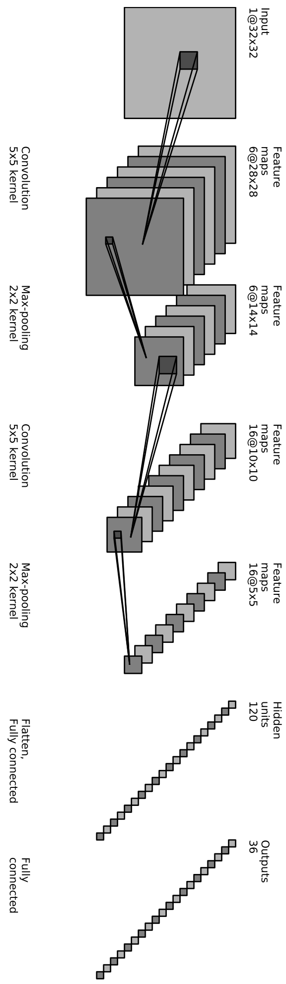


Figure 9: The architecture of a character recognition CNN.

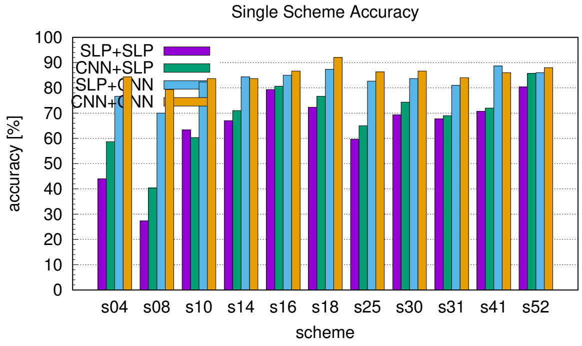


Figure 10: The accuracy of captcha image transcription separately for each scheme.

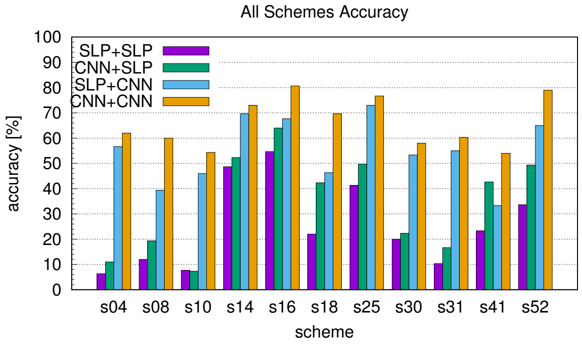


Figure 11: The accuracy of captcha image transcription when example images generated by all schemes were available in the training and test sets.

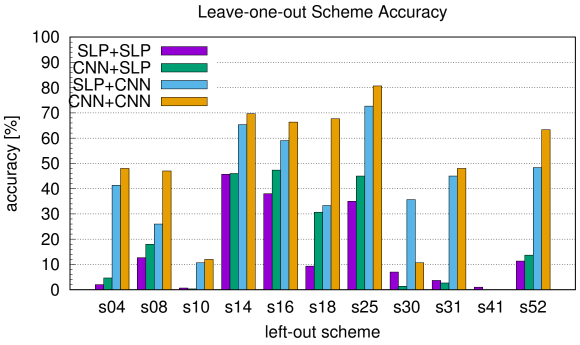


Figure 12: The accuracy of captcha image transcription in leave-one-scheme-out scenario.

ages generated by all schemes. Both training and testing set contained examples generated by all schemes. The results are depicted in Figure 11. In this experiment the CNN outperformed the SLP not only in the recognition but even in the localization accuracy. The most visible difference is on schemes s08, s18, s41. Overall performance is again compared by the statistical test with results summarized in Table 2. All accuracies are lower than in the previous experiment, as the data set complexity grown (data were generated by multiple schemes), but the number of training examples remained the same.

Table 2: Results of the statistical test of Friedman [12] and the correction for simultaneous hypotheses testing by Holm [15] and Shaffer [18]. The rejection thresholds are computed for the family-wise significance level $p = 0.05$ for all schemes.

Algorithms	p	Holm	Shaffer
SLP+SLP vs. CNN+CNN	1.259e-7	0.0083	0.0083
CNN+SLP vs. CNN+CNN	2.799e-4	0.01	0.0166
SLP+SLP vs. SLP+CNN	9.569e-4	0.0125	0.0166
SLP+CNN vs. CNN+CNN	0.047	0.0166	0.0166
SLP+SLP vs. CNN+SLP	0.098	0.025	0.025
CNN+SLP vs. SLP+CNN	0.098	0.05	0.05

The last experiment tested the accuracy of captcha transcription in leave-one-scheme-out scenario. The training set contained images generated by only 10 schemes and the images used for testing were all generated by the last yet unseen scheme. Trying to recognize characters from images generated by an unknown scheme is a challenging task, furthermore the schemes were selected to differ form each other as much as possible. The results are depicted in Figure 12. All configurations using a perceptron as the recognition classifier fail in all except the most simple schemes, e.g. s12 and s16. The combination of two CNNs is the best in all cases, with only exception being the scheme s30, where the combination of the localization perceptron and the recognition CNN is the best. Overall, the accuracy may seem relatively low, especially for schemes s10, s30, s31 and s41, but lets recall that recognition rate of 1% is already considered enough to compromise the scheme. The failure of CNNS on scheme s41 is understandable as the spiderweb background confuses the convolutional kernels learned on other schemes.

This is the most important experiment showing the ability to solve yet unseen captcha .The ranking of all algorithms is summarized in Table 3 and the statical tests in Table 4.

Table 3: Average Rankings of the algorithms

Algorithm	Ranking
CNN+CNN	1.27
SLP+CNN	2.00
CNN+SLP	3.27
SLP+SLP	3.45

Table 4: Results of the statistical test of Friedman [12] and the correction for simultaneous hypotheses testing by Holm [15] and Shaffer [18]. The rejection thresholds are computed for the family-wise significance level $p = 0.05$ for the leave-one-scheme-out scenario.

Algorithms	p	Holm	Shaffer
SLP+SLP vs. CNN+CNN	7.386e-5	0.0083	0.0083
CNN+SLP vs. CNN+CNN	2.799e-4	0.01	0.0166
SLP+SLP vs. SLP+CNN	0.008	0.0125	0.0166
CNN+SLP vs. SLP+CNN	0.020	0.0166	0.0166
SLP+CNN vs. CNN+CNN	0.186	0.025	0.025
SLP+SLP vs. CNN+SLP	0.741	0.05	0.05

The above experiments show that most of current schemes can be compromised using two convolutional networks or a localization perceptron and a recognition CNN.

6 Conclusion

In this paper, we presented a novel captcha recognition approach, which can fully replace the state-of-the art scheme specific pipelines. Our approach not only consists of less steps, but it is also more general as it can be applied to a wide variety of captcha schemes without modification. We were able to compromise 10 out of 11 using two CNNs or a localization perceptron and a recognition CNN without previously seeing any example image generated by that particular scheme. Furthermore, we were able to break all 11 captcha schemes using a CNN for the localization as well as for the recognition, with the accuracy higher than 50% when we included example images of each character generated by the particular scheme into the training set. Lets recall that 1% recognition rate is enough for a scheme to be considered compromised.

We experimentally compared the ability of SLP, MLP and CNN to transcribe characters from captcha images. According to our experiments, CNNs performs much better in both localization and recognition.

Acknowledgement

The research reported in this paper has been supported by the Czech Science Foundation (GAČR) grant 17-01251 and student grant SGS17/210/OHK3/3T/18.

References

[1] Botdetect captcha generator [online], 2017. www.captcha.com [Cited 2017-06-01].

[2] Free captcha-service [online], 2017. www.captchas.net [Cited 2017-06-01].

[3] Metal captcha, 2017. www.heavygifts.com/metalcaptcha [Cited 2017-06-01].

[4] Resisty captcha, 2017. www.wordpress.org/plugins/resisty [Cited 2017-06-01].

[5] Secureimage php captcha [online], 2017. www.phpcaptcha.org [Cited 2017-06-01].

[6] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention. In *International Conference on Learning Representations*, 2015.

[7] Elie Bursztein, Jonathan Aigrain, Angelika Moscicki, and John C Mitchell. The end is nigh: Generic solving of text-based captchas. In *8th USENIX Workshop on Offensive Technologies (WOOT 14)*, 2014.

[8] Elie Bursztein, Steven Bethard, Celine Fabry, John C Mitchell, and Dan Jurafsky. How good are humans at solving captchas? a large scale evaluation. In *2010 IEEE Symposium on Security and Privacy*, pages 399–413. IEEE, 2010.

[9] Elie Bursztein, Matthieu Martin, and John Mitchell. Text-based captcha strengths and weaknesses. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 125–138. ACM, 2011.

[10] Kumar Chellapilla, Kevin Larson, Patrice Simard, and Mary Czerwinski. Designing human friendly human interaction proofs (hips). In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 711–720. ACM, 2005.

[11] Claudia Cruz-Perez, Oleg Starostenko, Fernando Uceda-Ponga, Vicente Alarcon-Aquino, and Leobardo Reyes-Cabrera. Breaking recaptchas with unpredictable collapse: heuristic character segmentation and recognition. In *Pattern Recognition*, pages 155–165. Springer, 2012.

[12] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701, 1937.

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[14] Ian Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. In *International Conference on Learning Representations*, 2014.

[15] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[16] CH. Lampert, MB. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR 2008*, pages 1–8, Los Alamitos, CA, USA, 2008. Max-Planck-Gesellschaft, IEEE Computer Society.

[17] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[18] Juliet Popper Shaffer. Multiple hypothesis testing. *Annual review of psychology*, 46(1):561–584, 1995.

[19] F. Stark, C. Hazırbaş, R. Triebel, and D. Cremers. Captcha recognition with active deep learning. In *GCPR Workshop on New Challenges in Neural Computation*, 2015.

[20] Luis Von Ahn, Manuel Blum, Nicholas J Hopper, and John Langford. Captcha: Using hard ai problems for security. In *Advances in Cryptology—EUROCRYPT 2003*, pages 294–311. Springer, 2003.