A deep learning model for recognition of complex Text-based CAPTCHAs

Rafaqat Hussain Arain[1], Riaz Ahmed Shaikh[1], Abdullah Maitlo[1], Kamlesh Kumar[2], Syed Safdar Ali Shah[1]

[1]Department of Computer Science, Shah Abdul Latif University, Khairpur, Sindh, Pakistan
[2]Department of Computer Science, Sindh Madressatul Islam University, Karachi, Sindh, Pakistan

Summary
Over last decades or so, CAPTCHAs are used to differentiate between humans and bots. Although many alternatives of CAPTCHAs are introduced in recent years but still Text-based CAPTCHAs are most prevalent among all other alternatives on the internet. The traditional approach to recognize Text-based CAPTCHAs involves preprocessing, segmentation and finally recognition. This approach needs explicit segmentation of individual characters. Any weakness in prior steps leads to incorrect recognition. In this work, instead of using three-step approach, we have implemented a holistic approach to solve Text-based CAPTCHAs using a deep CNN. Significant improvements in the results have been achieved using our deep learning model. As deep leaning network need huge amount of data therefore we have synthetically generated two types of datasets; i.e. easy and complex types of CAPTCHAs. Our model has shown an accuracy of 86.5 % and 83.3% on easy and complex datasets respectively.

Key words:
Bots, CAPTCHAs, CNN, HIPs, Text warping

1. Introduction

The need of development of CAPTCHA (Completely Automated Public Turing test to tell Computers and Human Apart) was raised when an online survey was conducted in 1999 in order to know about best graduate school in computer science. The competition was held between Carnegie Mellon University (CMU) and MIT students. However students of CMU developed a program to vote for them. This automated program voted thousands of times and CMU's score commenced growing quickly. On the other day, students of MIT also developed similar type of program. In the end it became the competition between bots rather than humans. As a result the need for a mechanism to distinguish between bots and humans was raised. The term CAPTCHA was first used by Luis Von Ahn[1]. In this work Ahn et al. suggested many ways to distinguish between human and bots. Since then CAPTCHAs are widely used to protect the web against these automated programs. CAPTCHAs are now used in Online voting systems, free Email registration systems, online blogs, E-ticketing systems etc. Since the development of first Text CAPTCHA, many other alternatives of CAPTCHAs are designed and implemented by numerous websites. However due to its ease of implementation and small size still Text-based CAPTCHAs are most widely used type on the internet [2]. An example of Text-based CAPTCHA is shown in Figure 1. Over the years, various types of CAPTCHAs are developed by designers and decoded by attackers. As the attackers find vulnerabilities in the current design; the designers design new CAPTCHAs to overcome the weaknesses in the previous designs. Therefore it is an ongoing friendly war between designers and attackers. As the main idea behind development of CAPTCHAs was based on an AI (Artificial Intelligence) problem therefore breaking a CAPTHCA actually means solving an unsolved AI problem.

Traditional approaches towards solving a CAPTCHA consists of three steps; i.e. Preprocessing, segmentation and recognition. However any weakness in earlier step leads to incorrect recognition. In this work, we have used a holistic approach to solve complex Text-based CAPTCHAs. In this approach, deep neural networks, i.e. CNNs are used to recognize characters in a CAPTCHA holistically. Typically deep neural networks need huge amount of data. Although in our case no publically available datasets are available on the internet. Therefore we have synthetically generated CAPTCHAs by using a Java library. On our datasets, we have achieved significant improvements in the results as compared to previous traditional approaches. Rest of the paper is divided as follows: Section 2 presents a literature review, section 3 briefly describes CNNs, section 4 presents our proposed model, section 5 presents results, and finally section 6 presents conclusion.

Fig. 1 Static Text-based CAPTCHAs.

Manuscript received February 5, 2018
Manuscript revised February 20, 2018

2. Literature Work

CAPTCHA designing and breaking is an ongoing war since the development of first CAPTCHA[3][4]. Over the years many designing techniques based on hard AI problems have been proposed many researchers. The design weaknesses are however exploited by attackers and successfully decoded those CAPTCHAs. This is a friendly war and resulted in dual benefits. Successful breaking of a CAPTCHA results in one step forward in the field of AI while inability to decode them provides a security mechanism to safeguard the web.

Initial designs of Text-based CAPTCHAs were based on distortion of text along with addition of background noise. However research has proved that reading distorted text and noise removal are trivial tasks for automated attacks [5][6]. In fact addition of random lines and other noise arcs may usually effects the usability of the challenged text. Therefore it is established that segmentation is the most challenging task for current machine learning attacks [5]. Current CAPTCHAs are therefore based on segmentation resistant principle, i.e. the segregation of connected or overlapped characters rather than distortion of individual characters.

Mori and Malik have used shape contexts to decode Gimpy and EZ-Gimpy CAPTCHAs with 33% and 92% accuracy respectively [5]. Chellapilla and Simard decoded various types of CAPTCHAs by using machine learning attacks. They proposed the principles for building better HIPs [3]. Microsoft followed their principles to design their CAPTCHAs. However their proposed CAPTCHA were successfully broken by Yan and Ahmad [2]. Yan and Ahmad used vertical segmentation, pixel count and histogram methods to break Microsoft CAPTCHAs. Yan and Ahmad also presented a shape pattern method to break Google's CAPTCHA v.2010 which was based on connected characters [15]. Burstein et al. offered a tool to decode 13 prevalent CAPTCHAs. They attempted to break 15 CAPTCHAs out of which 13 were decoded successfully. Starostenko et al. proposed a three color bar encoding method to decode reCAPTCHA v.2011 with an accuracy of 54.6% [7].

An explicit segmentation scheme was proposed by Zhang and Wen. They proposed fuzzy matching method to decode CAPTCHAs. Instead of initial preprocessing and segmentation steps they performed integrated recognition based on mask matching [8]. Huang et al. suggested middle axis point separation and projection schemes to segment the connected characters. Gao et al. proposed stroke connection point and dynamic segmentation methods to break connected and disconnected CAPTCHAs with an accuracy from 12 to 88.8% [9]. Chandavale and Sapkal suggested a snake segmentation and modified projection based segmentation method to break connected and disconnected CAPTCHAs. Their method worked well for disconnected and overlapped characters and achieved an accuracy of 98%. However for connected characters this accuracy is reduced to 42% only[10]. Hussain et al. proposed recognition based segmentation method to segment the connected characters and achieved an accuracy of up to 62% using ANNs [11].

Convolutional neural networks (CNNs) were introduced in 1995 by Yann LeCun et al. in order to recognize handwritten characters [12]. CCNs are basically a type of feed forward neural networks. CNNs are composed of many layers including convolutional layers. CNNs are also called space invariant artificial neural networks due to their translation invariance characteristics [13]. CNNs use local dependencies in images which are used to obtain image features in our work. The convolutional layer applies a convolutional filter to convolve the input image which is a great way to produce multidimensional output. Convolutional layers are followed by fully connected layer which is used for mapping the spatial information to the classification information and it combines the former layer and applies non linearity to its output. Apart from convolutional and fully connected layers a max pooling layer is used to control over fitting by reducing the amount of parameters [14].

3. Proposed Model

Our problem of CAPTCHA recognition is a supervised learning problem as we have trained our datasets along with its answer pairs. The deep machine learning here aims to find the function based on the CAPTCHA images as input and answers to these sample images as output in the form of text string of 5 characters. The function y = f(x) is the solution and the great thing about machine learning is that when we provide a set of pairs as inputs and outputs then the machine learning is generic enough to find out the fundamental function. Our training network is somewhat similar to LeNet [12] architecture except the input and output size. We have used three convolutional layers, three pooling layers and two output layers. The proposed model is shown in Figure 2. We have used 36 characters including 0 to 9 decimal digits, and 26 alphabets. The output layer can generate 5 digits and every digit can be represented by 36 neurons (predicting the probability distribution) We have defined a function f(x), which maps every character x= { 0,1,2,…,9,a,b,c,…,z} to a unique number n = {0,1,2,3,…,35}as shown in equation 1.

$$f(x)=\begin{cases} 0,...9\ if x=0,...9 \\ 10,11,12,...35\ if\ \ x=10,11,12,...35 \end{cases} \quad (1)$$

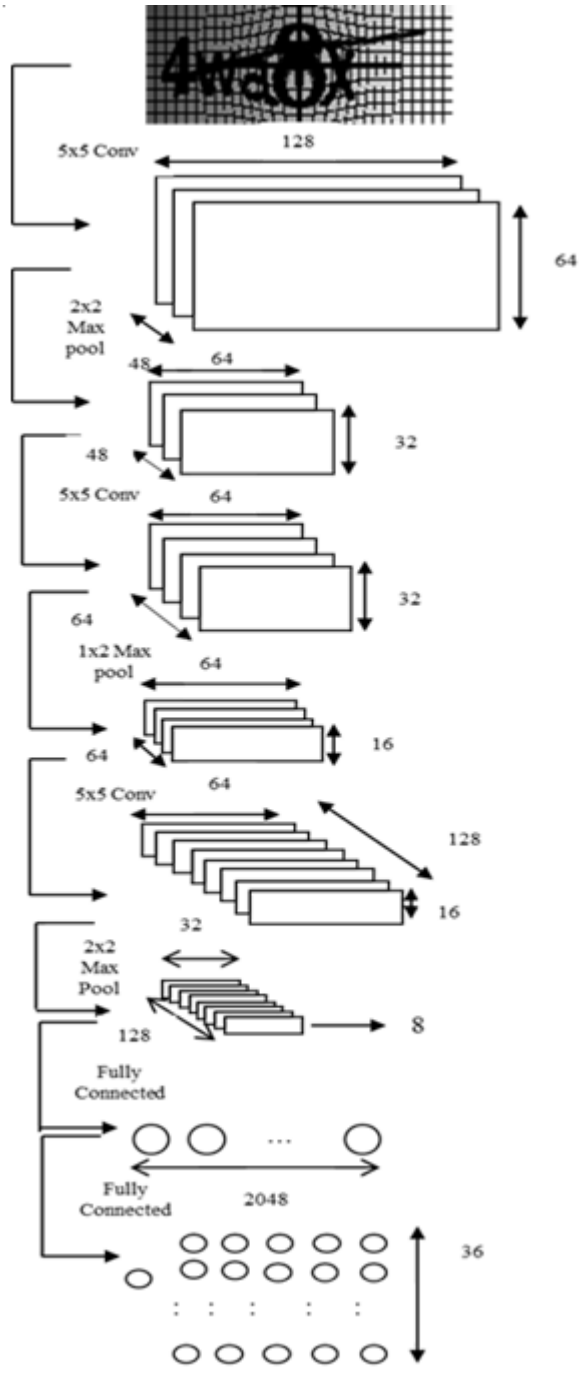

Fig. 2 Proposed CNN model

Since our problem is also a deep machine learning problem so we needed a huge amount of data to train our network but in our case no standard dataset is publically available. Therefore we decided to use Java library [15] for automatic creation of CAPTCHAs. This library lets you create multiple types of CAPTCHAs along with their labels (which are answers to the CAPTHCAs). We created 50000 sample images (64 x 128) having 5 characters in each sample image. The generated images were distorted using arcs, grids and other noises available in the above said library. A sample of our dataset image is shown in Figure 3.

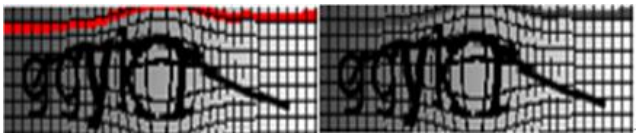

Fig. 3 Synthetically generated complex CAPTCHA images

Using the same library we further created 50,000 images of comparatively simple CAPTCHAs as shown in Figure 4.

Fig. 4 Synthetically generated simple CAPTCHA images

We have used cross entropy as a loss function in our model. This is a standard function used to measure the loss in various classification problems. Equation 2 is used to predict the loss.

$$L_i = -\sum j t_{i,j} \log\left(p_{i,j}\right) \tag{2}$$

Where P indicates the predication and t is the target matrix. We have 3 convolutional layers each having kernel size of 5x5. 3 max pooling layers and two fully connected layers in our model. Several activation functions are used to add the inputs from all the neurons such as sigmoid, logistics and ReLU. We have used ReLU in our model due to its speed and non-exploding nature [16]. Learning rate is set at 0.1. We have used Ceil Max pooling in order to save space as much as possible and control over fitting. In the dropout layer [17] a probability of 0.4 is used to consider a unit from the previous dense layer.

There are many machine learning frameworks available including Caffe, Torch, Theano and TensorFlow etc. We have used Torch 7 in our work. It is developed in U.S.A at New York University and NEC Laboratories. It is an open source, numerical computing framework which supports many machine learning, Computer vision, and image processing algorithms. It is based on fast scripting language LuaJIT with underlying C/CUDA implementation. It supports N-dimensional arrays, wonderful interface to C via LuaJIT, supports deep neural networks, and numeric optimization routines etc. Torch is

used by Google, Facebook, Twitter and many other companies. Due to its high speed, versatility, ease of use and flexibility we preferred it over other frameworks.

4. Results and Discussions

After generating CAPTCHA images by using Java library we started training. We put 10,000 images each for testing and validation. After training for 10,000 iterations on our GPU Nvidia 780 Titan, we achieved an accuracy of 67.25 % and 39.5% for maximum one error and whole sequence respectively on easy CAPTCHA images as shown in Figure 6-5. Here we are obviously overfitting. Adding drop out of 50 % on convolutional layers and 1st fully connected layer we achieved much better results with our model for 50,000 iterations. We have achieved an accuracy of 86.5% on whole sequence and 95.4% with maximum of one error on easy CAPTCHA images as shownn Figure 5. Similar experiments were carried out on complex CAPTCHA images (see Figure 6). Table 1 shows overall results of our model for both complex and easy CAPTCHAs.

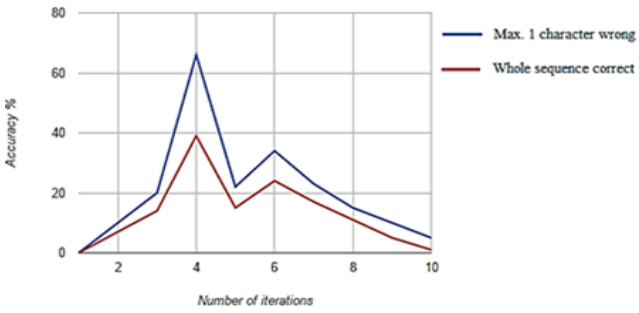

Fig. 5 Accuracy v/s Number of iterations

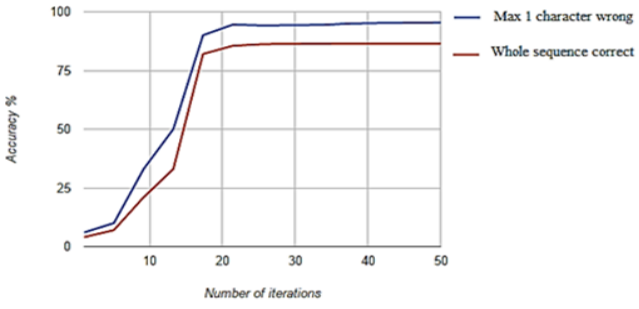

Fig. 6 Improvement in accuracy with increased number of iterations

5. Conclusion and Future Work

In this work a deep learning model is presented for the recognition of synthetically generated CAPTCHA images. These images were generated by using Java library. Instead of using traditional approaches for the recognition of CAPTCHAs where explicit segmentation is required before the recognition stage, here we have performed holistic recognition of CAPTCHAs. We have used 3 convolutional layers, 3 max pooling layers and two fully connected layers in our model to recognize the said Text-based CAPTCHAs. We have achieved an accuracy of 86.5 % and 83.3 % on complex and easy CAPTCHAs respectively. As a future work, variable length CAPTCHAs can be recognized by using deep learning networks. has been carried out to check the security of non lexicon based merged character CAPTCHAs. We have proposed an algorithm

Table 1: Results of Proposed Model

Type	Whole sequence Accuracy %	Max 1 character wrong %	No. of iterations
Easy CAPTCHA	39.5 86.5	67.25 95.4	10000 50000
Complex CAPTCHA	33.5 83.3	58.5 91	10000 50000

References

[1] L. V. Ahn, M. Blum, J. Langford. Telling humans and computers apart automatically[J]. Communications of the ACM, 2004, 47(2): 56-60

[2] J. Yan, A. S. El Ahmad. Usability of captchas or usability issues in captcha design[C]. 4th symposium on Usable privacy and security Proceedings, New York, 2008, 44–52

[3] K. Chellapilla, K. Larson, P. Y. Simard, et al. Building segmentation based human-friendly human interaction proofs[C]. Second International conference on Human Interactive Proofs Proceedings, Pennsylvania, 2005, 1-26

[4] Y. W. Chow, W. Susilo. AniCAP: An animated 3D CAPTCHA scheme based on motion parallax[C]. International Conference on Cryptology and Network Security Proceedings, Sanya, 2011, 255-271

[5] G. Mori, J. Malik. Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA[C]. International conference on Computer Vision and Pattern Recognition Proceedings, Washington, 2003, 134-141

[6] K. Chellapilla, P. Y. Simard. Using machine learning to break visual human interaction proofs[C].17th International Conference on Neural Information Processing Systems Proceedings, Columbia, 2004, 265-272

[7] O. Starostenko, C. Cruz-Perez, F. Uceda-Ponga, et al. Breaking text-based CAPTCHAs with variable word and character orientation [J].Pattern Recognition, 2015, 48(4): 1101-1112

[8] H. Zhang, X. Wen. The Recognition of CAPTCHA Based on Fuzzy Matching[C]. Eighth International Conference on Intelligent Systems and Knowledge Engineering Proceedings, Shenzhen, 2014, 759-768

[9] H. Gao, X. Wang, F. Cao, et al. Robustness of text-based completely automated public Turing test to tell computers and humans apart [J]. IET Information Security,2016, 10(1): 45-52

[10] A. A. Chandavale, A. Sapkal. A New Approach towards Segmentation for Breaking CAPTCHA[C]. International Conference on Security in Computer Networks and

Distributed Systems Proceedings, Trivandrum, 2012, 323-335
[11] R. Hussain, H. Gao, R. A. Shaikh. Segmentation of connected characters in text-based CAPTCHAs for intelligent character recognition[J]. Multimedia Tools and Applications, 2017, 76(24), 25547-25561.
[12] Y. LeCun, L. Bottou, Y. Bengio, et al. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11):2278-324.
[13] Z. Wei. Parallel distributed processing model with local space-invariant interconnections and its optical architecture [J]. Applied Optics, 1990, 29(32): 4790-4797
[14] D. Scherer, A. Müller, S. Behnke. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition [C]. 20th International Conference on Artificial Neural Networks (ICANN) Proceedings, Thessaloniki, 2010, 92–101.
[15] Simple CAPTCHA [OL], http://simplecaptcha.sourceforge.net/ 1 March 2017).
[16] A. Krizhevsky, I. Sutskever, G. E. Hinton. Imagenet classification with deep convolutional neural networks[C]. Advances in neural information processing systems Proceedings, Nevada, 2012, 1097-1105
[17] N. Srivastava, G. E. Hinton, A. Krizhevsky, et al. Dropout: A Simple Way to Prevent Neural Networks from overfitting [J]. Journal of Machine Learning Research, 2014, 15(1): 1929–1958.

Dr. Rafaqat Hussain Arain has completed his Ph.D. in Computer Science, from University of Electronic Science and Technology of China in 2017. His areas of interest include Image Processing, Machine learning, and Artificial Intelligence. He is serving as Assistant Professor in the Department of Computer Science, Shah Abdul Latif University, Pakistan.

Dr. Riaz Ahmed Shaikh has received his Ph.D. in Computer Science, from University of Electronic Science and Technology of China in 2016. Currently he is working as Assistant Professor in the Department of Computer Science, Shah Abdul Latif University, Pakistan. His research areas include Image Processing, CBIR and Computer Vision.

Dr. Abdullah Maitlo, Assistant Professor, Department of Computer Science, Shah Abdul Latif University Pakistan has done Ph.D. from University of Central Lancashire, UK. His research interests include Cyber Security, Online Business Security, Knowledge Management, Knowledge Economy, and Social Networks Security.

Dr. Kamlesh Kumar obtained his Ph.D. from University of Electronic Science and Technology of China in 2016. Currently, he is working as Assistant Professor at Sindh Madressatul Islam University, Karachi, Pakistan. His research interests include image processing, Computer Vision, CBIR System.

Syed Safdar Ali Shah pursuing his Ph.D. in Computer Science, from Shah Abdul Latif University (SALU), Pakistan. He is serving as Computer Programmer in SALU. His area of research is Image Processing.