POSTER: DeepCRACk: Using *Deep* Learning to Automatically *CR*ack *A*udio *C*APTCHAs

William Aiken
Sungkyunkwan University
Suwon, Republic of Korea
billzo@skku.edu

Hyoungshick Kim
Sungkyunkwan University
Suwon, Republic of Korea
hyoung@skku.edu

ABSTRACT

A Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) is a defensive mechanism designed to differentiate humans and computers to prevent unauthorized use of online services by automated attacks. They often consist of a visual or audio test that humans can perform easily but that bots cannot solve. However, with current machine learning techniques and open-source neural network architectures, it is now possible to create a self-contained system that is able to solve specific CAPTCHA types and outperform some human users. In this paper, we present a neural network that leverages Mozilla's open source implementation of Baidu's Deep Speech architecture; our model is currently able to solve the audio version of an open-source CATPCHA system (named SimpleCaptcha) with 98.8% accuracy. Our network was trained on 100,000 audio samples generated from SimpleCaptcha and can solve new SimpleCaptcha audio tests in 1.25 seconds on average (with a standard deviation of 0.065 seconds). Our implementation seems additionally promising because it does not require a powerful server to function and is robust to adversarial examples that target Deep Speech's pre-trained models.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Security and privacy** → *Graphical / visual passwords*; Usability in security and privacy;

KEYWORDS

CAPTCHA; neural networks; adversarial machine learning

ACM Reference Format:
William Aiken and Hyoungshick Kim. 2018. POSTER: DeepCRACk: Using *Deep* Learning to Automatically *CR*ack *A*udio *C*APTCHAs. In *ASIA CCS '18: 2018 ACM Asia Conference on Computer and Communications Security, June 4–8, 2018, Incheon, Republic of Korea.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3196494.3201581

1 INTRODUCTION

A Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) is an online challenge to distinguish humans from computers. That is, a CAPTCHA is based on any problem that a human can easily recognize and solve while a machine cannot (with acceptable probability subject to a real-time constraint). This security mechanism is popularly used for many websites in order to control usage of their services as well as prevent the automation of a variety of attacks. For example, a CAPTCHA can be used to limit the rate at which spammers can create new accounts in an automatic manner.

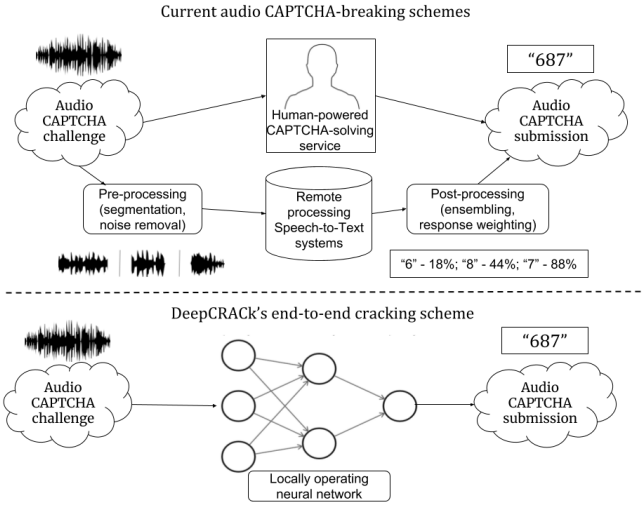

Figure 1: Comparison of DeepCRACk with current audio CAPTCHA-breaking schemes. DeepCRACk performs completely self-contained identification where other CAPTCHA solvers rely on external services.

Image-based CAPTCHAs (e.g., reCAPTCHA [3]) are the most popularly used CAPTCHA type in practice even though they often burden users with difficult challenges. To make matters worse, visual CAPTCHAs cannot properly be used for users with a visual impairment such as blindness or a cognitive disability like dyslexia; thus, visual CAPTCHAs fail to meet the United States' Section 508 requirement [9] to "require Federal agencies to make their electronic and information technology (EIT) accessible to people with disabilities". In such situations, an audio-based CAPTCHA seems a good alternative or complement to visual CAPTCHAs. Therefore, it is important to develop robust and efficient audio-based CAPTCHAs and evaluate their security and performance.

In this paper, we demonstrate that developing a secure audio-based CAPTCHA is not easy against CAPTCHA solvers that are based on deep learning techniques. To show the effectiveness of

ASIA CCS '18, June 4–8, 2018, Incheon, Republic of Korea

ACM ISBN 978-1-4503-5576-6/18/06.
https://doi.org/10.1145/3196494.3201581

deep learning-based CAPTCHA solvers, we trained a neural network that leverages Mozilla's implementation[1] of Baidu's Deep Speech architecture [4]; our trained model is currently able to solve the audio version of the open-source CATPCHA system SimpleCaptcha with 98.8% accuracy after only one day of training. Unlike previous deep learning-based CAPTCHA solvers [1, 5, 8], or CAPTCHA-breaking for-hire services, our model operates end-to-end and does not rely on any external systems as shown in Figure 1. Moreover, we evaluate the performance of DeepCRACk against carefully crafted adversarial audio samples that could attempt to fool machine learning-based CAPTCHA solvers. DeepCRACk can successfully recognize all samples while the pre-trained Deep Speech models are fooled by every example.

2 RELATED WORK

The potential to attack CAPTCHAs by leveraging neural networks has already been demonstrated in previous research. A lot of research has gone into breaking visual CAPTCHAs with neural networks, and a noteworthy example of this is Kopp et al.'s work [5]. In their approach, they replace a pipeline that removes hand-designed pre-processing, denoising, segmentation, and post processing with a two-step character localization and recognition convolutional neural network at greater than 50% accuracy for obfuscated character-based image CAPTCHAs. Additionally, Sivakorn et al. [8] developed a pipeline for image recognition and the solving of Google's reCaptcha that included a Google Reverse Image Search, image annotation via various deep-learning libraries, and tag word classifier that performed above 70% accuracy in online mode (using external services) and above 40% in offline mode (self-contained).

While most of this research has focused on breaking visual CAPTCHAs, Bock et al.'s unCaptcha system [1] proposed a low-resource framework for defeating audio CAPTCHAs in Google's reCaptcha. In their system, they download the audio from reCaptcha, pre-process it by extracting each spoken digit by intervals of silence to be sent to remote online speech recognition services like Google Speech API, IBM Bluemix, etc. After receiving the results, their system maps homophones and partial match sounds to digits ("mine" to "9", "*icks" to "6", etc.) and performs ensembling to weight some speech systems higher than others. Ultimately, they were able break audio reCaptcha over an 85% success rate. Google has been updating their reCaptcha system to combat these vulnerabilities, but other CAPTCHA systems still rely on audio digit recognition.

3 PROPOSED APPROACH

External speech-to-text systems may include specific usage limits and are subject to API changes, and human-solving CAPTCHA services are non-free and sometimes unreliable. Unlike other approaches to cracking audio CAPTCHAs, DeepCRACk aims to be end-to-end and able to function without any external assistance. We suspect that the bidirectional recurrent neural network (BRNN) is the most suitable neural network for this task. A BRNN splits a traditional recurrent neural network into two different states: one for analyzing the input in the forward direction, and one for analyzing the input in the backward direction. In this way, the network can use information from both the past and the future to help analyze the current frame. Moreover, the mel-frequency cepstrum coefficients (MFCCs) contain enough information about a time slice of audio to serve as sufficient input to the network; they not only approximate human auditory system responses to audio segments but also represent amplitudes in that spectrum. Fortunately, Mozilla's implementation of Deep Speech already utilizes such a framework.

Because our goal at this time is to target only the CAPTCHAs generated by the audio version of an open-source CATPCHA system (named SimpleCaptcha), we do not need the full size of the Deep Speech neural network. Because Mozilla's Deep Speech is designed for very complex speech-to-text tasks, using a full-size implementation would be unnecessary for our tasks. Thus, we restricted our model to 486 hidden neurons per hidden layer. Using the default settings of SimpleCaptcha, there are 7 voices per number, and 3 background noises (radio tuning, restaurant sounds, and swimming pool environment) are available for obfuscation. Each CAPTCHA consists of 5 digits, so for each position, there are 70 possibilities (7 voices by 10 digits). The total number of combinations of voices and digits is therefore 1,680,700,000 (70^5), each of which could be perturbed by one of the 3 background noises for a total of 5,024,100,000 unique possible CAPTCHAs. While our neural network is unable to solve every possibility, we found that there are some combinations which are quite difficult for humans as well.

Moreover, we believe that we could reduce the size even further without a substantial loss of accuracy because the long short-term memory layers may be unnecessarily large for a CAPTCHA-breaking task. Unlike unCaptcha, we aim to develop our model to be fully independent of other speech recognition systems as well as totally end-to-end. DeepCRACk performs no pre-processing, analyzes the data locally, and returns a complete prediction result. It also has no knowledge of the length of the CAPTCHA, even though SimpleCaptcha always generates CAPTCHAs of length 5 by default. We could not use Mozilla's pre-trained binaries as a starting point because their model only includes lowercase letters and no digits inherently. In order to build upon our model for future use, we simply expanded the default alphabet to include the digits 0 through 9 and retrained from the beginning. Given a change in CAPTCHA type that is alpha-numeric based, our model could serve as the baseline for transfer learning as well.

4 EVALUATION

To show the feasibility of DeepCRACk, we examined its performance with CAPTCHA challenges generated by SimpleCaptcha. For evaluation, we generated 100,000 SimpleCaptcha samples for our train set, 10,000 for our development set, and 10,000 for our test set. We trained our model on the Google Cloud Compute engine using a Tesla K80 GPU. Google provides $300 worth of compute usage for a free trial, and our entire training time was slightly more than 24 hours, which is well within the free trial limit.

On the testing set of 10,000 SimpleCaptcha samples, DeepCRACk performed well, obtaining an accuracy of 98.8%. We consider digits which occurred in the audio and which were detected as true positives (49,946). We consider digits which were reported by the system but which were not actually present as false positives (75), and we consider digits which were present in the audio but which

[1] https://github.com/mozilla/DeepSpeech

were not detected as false negatives (52). Using this, our system performed with 99.85% precision, and with 99.89% recall. The time required to solve a CAPTCHA was 1.25 seconds on average with a standard deviation of 0.065 seconds on a MacBook Pro Early 2011 model (8GB RAM, 2.3GHz CPU) running High Sierra 10.13.3 and running Deep Speech 0.1.1 in CPU mode. Therefore, given a model trained to crack SimpleCaptcha, most commodity hardware will be able to solve SimpleCaptcha's default CAPTCHAs in real time.

Focusing on one type of audio CAPTCHA runs the risk of overfitting to that particular task, and indeed, our model does not generalize to new obfuscation techniques and voices. It cannot solve other audio CAPTCHAs, even those that are numeric. For example, DeepCRACk was unable to solve any numerical audio CAPTCHAs generated by reCAPTCHA. ReCAPTCHA uses a wider variety of obfuscation techniques and utilizes voices different from those that appear in SimpleCaptcha. Nonetheless, creating a model that overfits to a particular CAPTCHA scheme will suffice. We propose that breaking other similar CAPTCHA systems with comparable efficiency will be possible after sufficient data collection and training.

5 ADVERSARIAL SOLUTION EXPLORATION

There already exist many proposals for replacing visual CAPTCHAs, most of which require the user to identify items ranging from biometric features like eyes [7] to everyday objects like chairs [6] in complex scenes. Nonetheless, these types of CAPTCHAs are completely inaccessible to those users with poor or no eyesight. For audio CAPTCHAs, increasing the obfuscation of the sound would deter more bots, but this would also inhibit human users as well. The authors of unCaptcha [1] propose that auditory instructions like "type the following word" could serve as the next generation of audio CAPTCHAs, but there may be solutions available to augment current mechanisms without dramatic architecture changes.

Carlini and Wagner have shown that Deep Speech is vulnerable to audio-based adversarial attacks [2]. Although at this point research has not demonstrated universal perturbations, these findings represent a preliminary step to curb CAPTCHA-defeating neural networks. In our test, we used Deep Speech's pre-trained models and Carlini's method hosted on GitHub[2] to generate adversarial audio against DeepCRACk. The source audio samples were SimpleCaptchas, and our target adversarial audio was "one two three four five" (Deep Speech pre-trained releases do not allow for digits). On the 100 SimpleCaptcha samples we used, Carlini's adversarial generation technique was able to correctly fool the same release that generated the audio (either 0.1.0 or 0.1.1) to return the target adversarial audio, and it was able to confuse the other model as well, which returned nonsense like "o i o sax rl". However, our DeepCRACk model correctly returned the ground truth. Our model performed with 100% accuracy while Deep Speech's pre-trained models performed with 0% accuracy on these adversarial examples. Note that at this time we are unable to create adversarial examples targeting DeepCRACk's model because its shape differs from Deep Speech's default shape; we leave this application for future work.

As for the inability to fool DeepCRACk using other pre-trained releases, the difference likely lies in the size of the trained models as well as the data they were trained on. Further research will determine whether it is possible to apply adversarial audio generation from one neural network architecture to another. For now, fooling an audio-cracking neural network requires knowledge of the underlying architecture on which it was trained, which is not reasonable for large scale CAPTCHA deployment.

[2]https://github.com/carlini/audio_adversarial_examples

6 CONCLUSION

In this paper, we proposed an end-to-end neural network capable of defeating SimpleCaptcha's default settings at 98.8% accuracy. A replication of our work could be performed using Google Cloud Compute's free trial well within the free trial time period. We also investigated the robustness of DeepCRACk against adversarial audio samples attempting to confuse our model. Our DeepCRACk implementation successfully recognizes adversarial samples while the popularly used speech recognition system Deep Speech fails against them. Even though the tested adversarial examples were not generated to target our DeepCRACk implementation directly, the experiment results demonstrate the potential effectiveness of DeepCRACk against adversarial examples. As part of future work, we plan to implement more sophisticated adversarial audio samples and evaluate the performance of DeepCRACk against them.

RESPONSIBLE DISCLOSURE

We demonstrate a fully working attack tool that is capable of solving SimpleCaptcha. We contacted the primary creator to report our findings via SourceForge and highly encourage future adopters to provide their own voice samples and obfuscating background sounds. Doing so would allow for audio CAPTCHA implementation while still providing some protection, at least in the short term.

ACKNOWLEDGMENTS

This research was supported in part by the ITRC program (IITP-2017-2015-0-00403) and the NRF program (2017R1D1A1B03030627).

REFERENCES

[1] Kevin Bock, Daven Patel, George Hughey, and Dave Levin. 2017. unCaptcha: a low-resource defeat of reCaptcha's audio challenge. In *Proceedings of the 11th USENIX Workshop on Offensive Technologies*.
[2] Nicholas Carlini and David Wagner. 2018. Audio adversarial examples: Targeted attacks on speech-to-text. *arXiv preprint arXiv:1801.01944* (January 2018).
[3] Google. 2018 (accessed March 1, 2018). *Google reCAPTCHA: Tough on bots, Easy on humans.* https://www.google.com/recaptcha/intro/android.html
[4] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. 2014. Deep Speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567* (December 2014).
[5] Martin Kopp, Matěj Nikl, and Martin Holeň. 2017. Breaking CAPTCHAs with convolutional neural networks. In *Proceedings of the 17th Conference on Information Technologies - Applications and Theory.*
[6] Brian M. Powell, Ekampreet Kalsy, Gaurav Goswami, Mayank Vatsa, Richa Singh, and Afzel Noore. 2017. Attack-Resistant aiCAPTCHA using a negative selection artificial immune system. In *IEEE Security and Privacy Workshops.*
[7] Brian M. Powell, Abhishek Kumar, Jatin Thapar, Gaurav Goswami, Mayank Vatsa, Richa Singh, and Afzel Noore. 2016. A multibiometrics-based CAPTCHA for improved online security. In *IEEE 8th International Conference on Biometrics Theory, Applications and Systems.*
[8] Suphannee Sivakorn, Iasonas Polakis, and Angelos D Keromytis. 2016. I am robot: (deep) learning to break semantic image CAPTCHAs. In *IEEE European Symposium on Security and Privacy (EuroS&P).*
[9] Christine Sket. 2017 (accessed March 8, 2018). *508 Compliance: Who Needs to be Compliant?* https://brailleworks.com/508-compliance-needs-compliant/