EXTRACTING AND DECIPHERING CAPTCHA API UNDER WEB CONTEXT

Simran Sharma[1], Nidhi Seth[2]
M.Tech Student[1], Assistant Professor [2]
Computer Science & Engineering Department
JMIT, Radaur/ Kurukshetra University, India
[1]simran1192@gmail.com
[2] nidhiseth@jmit.ac.in
9896384299

***Abstract*—** with advances of segmentation and Optical Character Recognition (OCR) technologies, the capability gap between humans and bots in recognizing distorted and connected characters becomes increasingly smaller. This trend would likely render text CAPTCHAs APIs eventually ineffective. The basic challenge in designing these obfuscations is to make them easy enough that users are not dissuaded from attempting a solution, yet still too difficult to solve using available computer vision algorithms. Main Focus of this work is to find out the Probabilities of finding out the real text behind a given CAPTCHA API. To find such a probability we have to first, Implementation a CAPTCHA Generation Algorithm in a programming language using some open CAPTCHA generation algorithm, such as Tesseract. Then apply Image Heuristics of CAPTCHA which include Image alignment, Noise Reduction filters etc. The CAPTCHA's Heuristics alongside Filtering information will be provided to an OCR Library, such as GOCR to find out The CAPTCHA's Text in the Main Decryption algorithm. Finally, Probability of successful CAPTCHA can be studied for each input CAPTCHA and Total system probability is calculated. some techniques we have discussed in this thesis provide more than 70% success rate, and as the faulty CAPTCHA requests are re-evaluated by the server and absence limiting count means that CAPTCHA decryption will be successful in consecutive attacks.

Keywords--- CAPTCHA API, HIP, Tesseract OCR, Computer vision, CAPTCHA, Attacks, CAPTCHA Recognizer, Heuristics

1. INTRODUCTION

Human interactive proofs (HIPs) are trials that can be given via multimedia to a user to aid assure that a human being, as challenged to an automated arrangement, is interacting alongside the software. HIPs are being utilized by extra and extra web locations to stop, for example, automated conception of several web-based email accounts. Such Automatically crafted reports are oftentimes utilized for spam aggressions and supplementary non-desirable activities.

One public example of an HIP is a picture that includes text that could be an actual word or phrase, or could be a nonsensical combination of messages, digits, and supplementary characters. To resolve the HIP trial a user kinds in the acts that are shown. Supplementary kinds of trials (e.g., audio and/or video challenges) could additionally be industrialized as HIPs, alongside the intention being to ascertain, for example, whether a particular appeal consented by a web locale is being commenced by a human being.

While a little entities (e.g., colossal corporations) could have the resources to scrutiny and develop HIP trials that are tough for an automated arrangement to resolve, countless supplementary entities (e.g., tiny web-based businesses) have the desire to use HIP knowledge, but could not have the resources to develop competent HIP challenges. A CAPTCHA (Completely Automated Area Turing Examination to Notify Computers and Humans Apart) is a plan that generates and grades examinations that are human solvable, but beyond the skills of present computer plans [1]. This knowledge is nowadays nearly an average protection mechanism for addressing unwanted or malicious Internet bot plans (such as those spreading junk emails and grasping thousands of free email reports instantly) and has discovered extensive request on countless commercial web locations encompassing Google, Yahoo, and Microsoft's MSN.

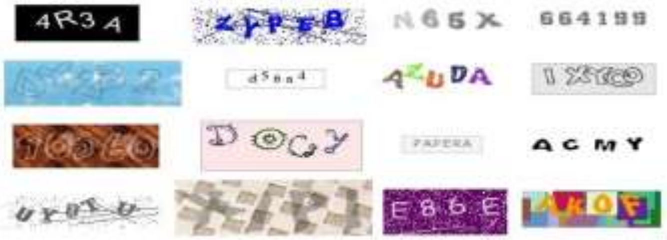


Figure 1

The word "CAPTCHA" was early gave in 2000 by Von an et al., [1] delineating an examination that can differentiate humans from bots. Below public dentitions, the examination has to be

1. Facilely resolved by humans,
2. Facilely generated and evaluated,
3. But, Not facilely resolved by computer

Over the past decade, a number of disparate methods for producing CAPTCHAs have been industrialized, every single fulfilling the properties delineated above to fluctuating degrees. The most usually discovered CAPTCHAs are discernible trials that need the user to recognize alphanumeric acts present in a picture Obfuscated by a little combination of sound and distortion. Figure 1 displays examples of such discernible CAPTCHAs. So distant, there are the pursuing three main kinds of CAPTCHAs:

1. Text-based schemes – they typically rely on sophisticated distortion of text images rendering them unrecognizable to the state of the art of pattern recognition programs but recognizable to human eyes.

2. Sound-based schemes (or *audio* schemes): - they typically require users to solve a speech recognition task.

3. Image-based schemes - they typically require users to perform an image recognition task.

In this paper, our discussion will largely focus on text-based
CAPTCHAs, for the following reasons:

First, text-based CAPTCHAs have been the most widely deployed schemes. Major web sites such as Google, Yahoo and Microsoft all have their own text-based CAPTCHAs deployed for years.

Second, text-based CAPTCHAs have countless gains compared to supplementary kinds of schemes [4], for example, being intuitive to users world-wide (the user task gave being just character recognition), possessing insufficient localization subjects, and possessing good potential to furnish forceful protection (e.g. the space a brute power attack has to find can be huge, if properly projected).

Third, it can have a colossal and affirmative encounter for the area to improve the usability of such accepted and well-claimed CAPTCHAs by recognizing subjects that ought to be addressed in these schemes.

The frank trial in arranging these obfuscating(make obscure, unclear, or unintelligible) CAPTCHAs is to make them facile plenty that users are not dissuaded(discourage) from endeavoring a resolution, yet too tough to resolve employing obtainable computer vision algorithms. As Current knowledge grows this gap though becomes slimmer and thinner. It is probable to enhance the protection of a continuing text CAPTCHA by systematically adding sound and supplementary distortions, or arranging acts extra tightly. These measures, though, should additionally make the acts harder for humans to understand, emerging in a higher error rates and higher Web load.

With advances of segmentation and Optical Character Credit (OCR) technologies, the skill gap amid humans and bots in knowing distorted and related acts becomes increasingly smaller. This trend should probable portray text CAPTCHAs APIs in the end ineffective.

2. RELATED WORK

In 2014, Haichang Gao, Wei Wang, Ye Fan, Jiao Qi and Xiyang Liu [10], used five methods to extract and segment the CCT CAPTCHA. They achieved the success rate of 55% on Yahoo, 34% on Baidu and 34% on reCAPTCHA. In 2014, Min Wang, Tianhui Zhang, Wenrong Jiang, and Hao SongIn [9], chose various methods to recognize CAPTCHA by analyzing the weaknesses in the designs and suggest various methods for the improvement of CAPTCHAS and they got the recognition rate of 20% CAPTCHA. Using machine learning algorithms, Carlos Javier Hernández-Castro, David F. Barrero, and María D. R-Moreno[13] successfully attacked on CIVIL RIGHT CAPTCHAs with a success rate of 10.5% and also showed that combination of two CAPTCHAs are not always secure the one CAPTCHA alone.

3. PROBLEMS WITH CAPTCHAs

CAPTCHAs have several limitations:

A. Usability is always an important issue in designing a CAPTCHA.
   With CAPTCHAs, usability and robustness are two fundamental issues and they are interconnected with each other. And to address the usability issue in CAPTCHA design, one must concentrate that it should be "user friendly".
B. It is possible to enhance the security of an existing text CAPTCHA by adding noise and distortion and arranging characters more tightly.
C. But There is a limit to the distortion and noise that humanscan tolerate in a challenge of a text CAPTCHA.
D. These measures, however, would also make the characters harder for humans to identify the CAPTCHA which results in a higher error rate.

Attacks on Text Based CAPTCHAS:

CAPTCHA has been adopted in the past years is very instructive. CAPTCHA has a feature that it restricts the bots to perform unlawful activity, number of locations adopted CAPTCHA. It restricts the wrong intentions of dispatching any virus to the system. Hence CAPTCHAs are highly utilized than and becomes the public part of present website login system. However the implementation in designing a CAPTCHA is extensively complex and very risky. The CAPTCHA scheme in the website can be easily cracked [3] by using some efficient methods.

- Other CAPTCHAs like Audio or Video-based CAPTCHAs are more "**secure**" than the text-based CAPTCHAs.
- Presenting logic questions to the users need greater efforts than normal image CAPTCHAs.
- To train and recognize text-based CAPTCHAs, neural networks can be used.
- Heuristics are findings in a process for solving a problem that will sufficient to indicate a given result. Based on the volume of request from the user, visited common pages, methods for data entry or any signature collected, it can be possible to detect whether a user is a robotic user or not. These are known as Heuristics checks
- There are some services by which CAPTCHA can be hack without any user efforts.

4. ACCESSING SECURITY OF A GIVEN CAPTCHA API

Millions of CAPTCHAs every single date are being validated by the providers having the reCAPTCHA and supplementary CAPTCHA API capabilities and to protect the thousands of web sites it also oppose the bots. Securely creation and validation of CAPTCHA forms the basis of the public belief ideal amid the consumer and the CAPTCHA provider. If each of the constituent of this ecosystem is compromised then damages collection can transpire.

Both the CAPTCHA creation and validation services are being proposed by the CAPTCHA API providers. The subscribing websites whichever uses the continuing libraries and plugins, or comprise their own in order to consume these services. A normal user contact alongside a web request that relies on a CAPTCHA provider is summarized below:

1. A user first requests a page that desires CAPTCHA validation.
2. The page that returns from the CAPTCHA provider will contain an img or script html tag to get the image CAPTCHA.
3. After resolving the html tags, browser will get an image CAPTCHA by the CAPTCHA provider and displays it to the user.
4. Then all the fields of the form are to be filled by the user including the CAPTCHA solution and surrender the page to the web application.
5. Then CAPTCHA solution is surrendered by the web application to the CAPTCHA provider for the process of verification.
6. Then the message of success or a failure is responded by the CAPTCHA provider to the web application.
7. Depending upon the response from the CAPTCHA provider, request get allows or denies by the web application.

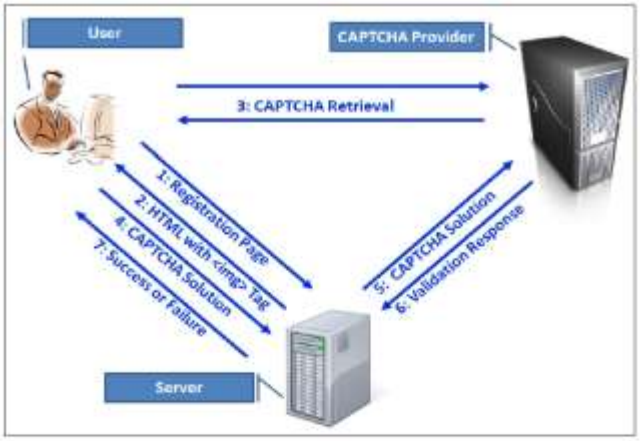


Figure 2: A typical validation flow with CAPTCHA providers

Steps 5 and 6 plays an important role in the CAPTCHA validation scheme and must be securely implemented to prevent attacks against CAPTCHA validation process.

5. PROPOSED WORK

Tesseract OCR Engine: Tesseract is a free software optical character credit engine for assorted working systems. Primarily industrialized as proprietary multimedia at Hewlett-Packard, it had extremely slight work completed on it in the pursuing decade. It was next released as open basis in 2005 by Hewlett Packard and UNLV. Tesseract progress has been sponsored by Google as 2006.[6] It is released below the Apache License, Edition 2.0.

The early editions of Tesseract might merely understand English speech text. Starting alongside edition 2 Tesseract was able to procedure assorted tongues encompassing Indian scripts. Nearly all Indian scripts are cursive in nature making them hard to understand by machines. Scripts like Devanagari, Gujarati, Bengali and countless others have conjuncts or joint-characters rising segmentation difficulties. To add to that, assorted fonts of assorted sizes utilized for creation texts above the years, the quality of paper, scanning resolution, pictures in texts etc. asks for a challenging picture processing job. Also, it needs huge linguistic know-how to apply post-processing.

Tesseract is a free multimedia optical character trust engine for varied working systems. Chiefly industrialized as proprietary multimedia at Hewlett-Packard amid 1985 and 1995, it had tremendously tiny work finished on it in the pursuing decade. It was subsequent released as open basis in 2005 by Hewlett Packard and UNLV. Tesseract progress has been sponsored by Google as 2006. It is released below the Apache License, Edition 2.0.

The Tesseract engine was industrialized at Hewlett Packard Workshops Bristol and at Hewlett Packard Co, Greeley Colorado amid 1985 and 1994, alongside slight supplementary adjustments made in 1996 to seaport to Windows, and a slight migration from C to C++ in 1998. A lot of the plan was composed in C, and subsequent a slight supplementary was composed in C++. As subsequent all the plan has been adjusted to at least amass alongside a C++ compiler.

Tesseract is plausibly the most precise open basis OCR engine available. Joined alongside the Leptonica Picture Processing Library it can elucidate an expansive collection of picture formats and change them to text in above 60 languages. It was one of the top 3 engines in the 1995 UNLV Accuracy test. Amid 1995 and 2006 it had tiny work finished on it, but as subsequent it has been enhanced extensively by Google. It is released below the Apache License 2.0.

Tesseract Features:
Tesseract is a GUI-based, exceedingly flexible, interactive, point-and-shoot CAPTCHA scrutiny instrument alongside the Following features:

1. A generic image preprocessing engine that can be configured as per the CAPTCHA type being analyzed.
2. Tesseract as its OCR engine to retrieve text from preprocessed CAPTCHAs.
3. Web proxy and custom HTTP headers support.
4. CAPTCHA statistical analysis support.
5. Character set selection for the OCR engine.

Tesseract was in the top 3 OCR engines in words of character accuracy in 1995. It is obtainable for Linux, Windows and Mac OS X, though, due to manipulated resources merely Windows and Usutu are rigorously tested by developers. Tesseract up to and encompassing edition 2 might merely accord TIFF pictures of easy one column text as inputs. These main editions did not contain layout scrutiny and so inputting multi-columned text, pictures, or equations produced a garbled output. As edition 3.00 Tesseract has upheld output text formatting, OCR positional data and page layout analysis. Prop for a number of new picture formats was added employing the Leetonia library. Tesseract can notice whether text is mono-spaced or proportional.

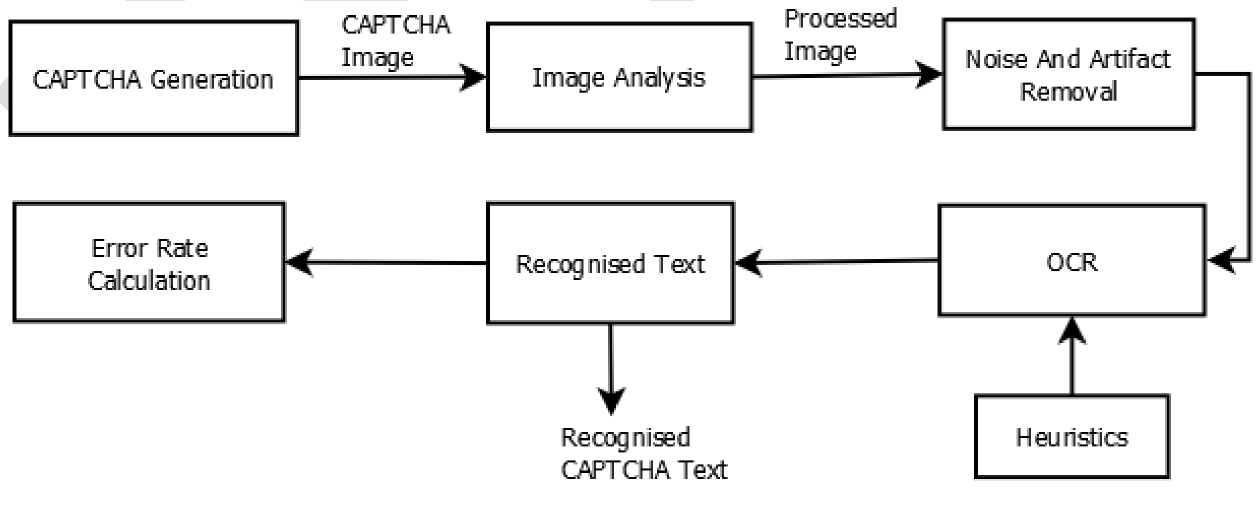


Figure 3: Working of CAPTCHA Recognizer using Tesseract OCR

6. ALGORITHM

Input: ***u*** = API url of the CAPTCHA to be deciphered, Number of Requests ***n***
Output: Total Success Rate **T**
for *i* = 1 to ***n***

```
c = API_Request(u,i),  where c is grabbed CAPTCHA image.
try
c = preprocess(c);
The preprocess removes irrelevant HTML code and applies image processing techniques such as filtering, noise removal, histogram equalizations and format conversion.
 $text_i$ = solve(c,iter, ocr) , $text_i$ is an indexed array of  solved CAPTCHA text
Where solve tries to decipher CAPTCHA using given OCR Engine which is trained for Text CAPTCHAs
end
Catch Exception
        return
for i = 1 to n do
Correct = select_correct($text_i$)
Incorrect = select_incorrect($text_i$
end
```

Calculate **Success Rate** using,

$$\textbf{Success Rate} = \frac{\sum_{i=1}^{n} \text{Total Error}}{n}$$

Where **Total Error** is calculated using

$$\textbf{Total Error} = \frac{\text{Incorrect}}{\text{Correct} + \text{Incorrect}}$$

7. RESULTS

Error Calculation

The Error between two words is the minimum number of single character edits; insertion, deletion, and substitution are required to change OCR word into the original word. The phrase edit distance is often used to refer specifically to Error calculation.

Word Error is a measure of the similarity between two strings, source string (s) and the target string (t).

If s is "test" and t is "test", then Error(s,t) = 0, because no transformations are needed. The strings are Equal hence no Error.

If s is "test" and t is "tent", then Error(s,t) = 1, because one substitution (change "s" to "n") is sufficient to transform s into t.

Hence, the greater the Error, the more different the strings are.

Error rate Calculation

To calculate the exact percentage Error between two strings, we need to check if two words are similar to a certain given percentage (60% for example). Following is the exact mathematical formula for Error Rate Calculation

$$\textbf{Error rate} = \frac{\text{Errors}_{(s,t)}}{\text{Max}_{(s,t)}} * 100$$

Success Rate Calculation of the CAPTCHA OCR

The Success Rate can be calculated by Averaging the Error rate per CAPTCHA. The Success rate of CAPTCHA OCR can be calculated as,

$$\textbf{Success Rate} = \frac{\sum_{i=1}^{n} \text{Error Rate}}{n}$$

Here *n*, is the total no of CAPTCHAs and Error Rate is the total Error of the CAPTCHA.

The Resultant Success rate of the CAPTCHA Recognizer using Tesseract OCR Libraries for selected set of 11 CAPTCHAS is:

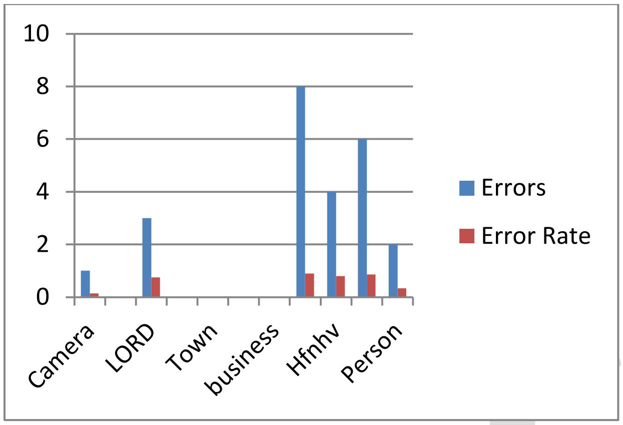
$\sum_{i=1}^{n}$ Error Rate = 3.7722

We Collected 11 CAPTCHAS and Success Rate for,

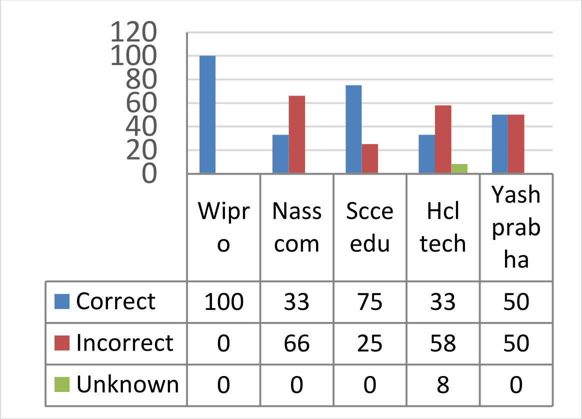
Success Rate= 3.7722/11

= 0.34292 * 100

= 34.29%



Success Rate of various services using APIs:



	Wipro	Nasscom	Scceedu	Hcl tech	Yashprabha
🟦 Correct	100	33	75	33	50
🟥 Incorrect	0	66	25	58	50
🟩 Unknown	0	0	0	8	0

8. CONCLUSION AND FUTURE SCOPE

Conclusion

CAPTCHAs are utilized in endeavors to stop automated multimedia from giving deeds that degrade the quality of ability of a given system. CAPTCHAs are additionally utilized to minimize automated postings to assorted sites. CAPTCHAs have countless supplementary requests for useful security.

It is probable to enhance the protection of a continuing text CAPTCHA by systematically adding sound and distortion, and arranging acts extra tightly. These measures, though, should additionally make the acts harder for humans to understand, emerging in a higher error rate. There is a check to the distortion and sound that humans can tolerate in a trial of a text CAPTCHA. Usability is always a vital subject in arranging a CAPTCHA.

With advances of segmentation and Optical Character Credit (OCR) technologies, the skill gap amid humans and bots in knowing distorted and related acts becomes increasingly smaller. This trend should probable portray text CAPTCHAs in the end ineffective. This suggests that Text CAPTCHA aggressions work in a frank level. The assorted OCR aggressions materialize to be applicable to finished relations of text CAPTCHAs. The Aggressions are craft on top of the accepted segmentation resistant mechanism of crowding character jointly for security.

CAPTCHAs are yet a new scrutiny span; Open setbacks contain the mislabeling problem. Of all the setbacks they debated, mislabeling reasons the most human errors. The authors could be able to resolve this employing cooperative filtering, whereas recognized human users rate pictures according to how well they evoke their label.

Future Scope

By design, Text CAPTCHAs are easy and facile to resolve by humans. Their low encounter quality makes them appealing to locale operators who are distressed of each protection that might coil away possible visitors. Though, this alike quality has made them facile to attack. In this thesis, we have debates the resolving CAPTCHAs employing Open Basis OCRs, Displaying those CAPTCHAs vulnerable to such attacks.

A lot of work has been completed in Enhancing CAPTCHA usability and Protection one such example is use of reCAPTCHA, Though rise of present advents and methods made it extra tough to stop automated bots and supplementary hazardous spammers opposing CAPTCHA attacks. a little methods we have debated in this thesis furnish extra than 40% accomplishment rate, and as the defective CAPTCHA demands are re-evaluated by the server and nonexistence manipulating count way that CAPTCHA decryption will be prosperous in consecutive attacks.

In upcoming we should like to use our methods to more enhance the accomplishment rate of the CAPTCHAs. One more span of work will be to enhance picture heuristics. For enhancing intricacy of the OCR for credit of the words, a background picture will be added to the image. This background picture could encompass of a little random lines or a little noise. Removing these lines is a hard task for an OCR system. Because removing these lines could obliterate a little spots of the messages and change a character to another.

Also cursive character credit, the segmentation of the acts is harder than knowing the characters. So, adding lines make the segmentation of acts a tough for an OCR program.

REFERENCES:

[1] Saxena, Ashutosh, Nitin Singh Chauhan, K. R. Sravan, Aparajith Srinivasan Vangal, and David Palacios Rodrguez. "A new scheme for mobile based CAPTCHA service on Cloud." In Cloud Computing in Emerging Markets (CCEM), 2012 IEEE International Conference on, pp. 1-6. IEEE, 2012.

[2] D'Souza, Darryl, Phani C. Polina, and Roman V. Yampolskiy. "Avatar captcha: Telling computers and humans apart via face classification." In Electro/Information Technology (EIT), 2012 IEEE International Conference on, pp. 1-6. IEEE, 2012.

[3] Kiran Jain Azad, "CAPTCHA: Attacks and weaknesses against OCR Technology." Global Journal of Computer Science and Technology 13, no. 3 (2013).

[4] Achint Thomas, Kunal Punera, Lyndon Kennedy, Belle Tseng, and Yi Chang. "Framework for evaluation of text captchas." In Proceedings of the 22nd international conference on World Wide Web companion, pp. 159-160. International World Wide Web Conferences Steering Committee, 2013.

[5] Ye, Quan-Bin, Te-En Wei, Albert B. Jeng, Hahn-Ming Lee, and Kuo-Ping Wu. "DDIM-CAPTCHA: A Novel Drag-n-Drop Interactive Masking CAPTCHA against the Third Party Human Attacks." In Technologies and Applications of Artificial Intelligence (TAAI), 2013 Conference on, pp. 158-163. IEEE, 2013.

[6] Men, Tao, Deming Wang, Yan Sun, and Mingrong Wang. "A novel dynamic CAPTCHA based on inverted colors." In Instrumentation and Measurement, Sensor Network and Automation (IMSNA), 2013 2nd International Symposium on, pp. 796-799. IEEE, 2013.

[7] Yunhang Shen, Rongrong Ji, Donglin Cao, and Min Wang. "Hacking Chinese Touclick CAPTCHA by Multi-Scale Corner Structure Model with Fast Pattern Matching." In Proceedings of the ACM International Conference on Multimedia, pp. 853-856. ACM, 2014.

[8] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Synthetic data and artificial neural networks for natural scene text recognition." arXiv preprint arXiv:1406.2227 (2014).

[9] Min Wang, Tianhui Zhang, Wenrong Jiang, and Hao Song. "The recognition of CAPTCHA." Journal of Computer and Communications 2, no. 02 (2014): 14.

[10] Haichang Gao, Wei Wang, Ye Fan, Jiao Qi, and Xiyang Liu. "The Robustness of “Connecting Characters Together” CAPTCHAs." Journal of Information Science and Engineering 30, no. 2 (2014): 347-369.

[11] Ariyan Zarei, "Improve CAPTCHA's Security Using Gaussian Blur Filter." arXiv preprint arXiv:1410.4441 (2014).

[12] Zhu, B., Jeff Yan, Guanbo Bao, M. Mao, and Ning Xu. "Captcha as Graphical Passwords–A New Security Primitive Based on Hard AI Problems." (2014): 1-1.

[13] Carlos Javier Hernández-Castro, David F. Barrero, and María D. R-Moreno. "A Machine Learning Attack against the Civil Rights CAPTCHA." In Intelligent Distributed Computing VIII, pp. 239-248. Springer International Publishing, 2015.

[14] Yan, Jeff, and Ahmad Salah El Ahmad. "Breaking visual captchas with naive pattern recognition algorithms." In Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual, pp. 279-291. IEEE, 2007