Attack-Resistant aiCAPTCHA using a Negative Selection Artificial Immune System

Brian M. Powell[1], Ekampreet Kalsy[2], Gaurav Goswami[2], Mayank Vatsa[1,2], Richa Singh[1,2] and Afzel Noore[1]
[1]West Virginia University, [2]IIIT-Delhi
{brian.powell, afzel.noore}@mail.wvu.edu, {ekampreet12035, gauravgs, mayank, rsingh}@iiitd.ac.in

***Abstract*—The growth of online services has resulted in a great need for tools to secure systems from would-be attackers without compromising the user experience. CAPTCHAs (Completely Automated Public Turing Tests to Tell Computers and Humans Apart) are one tool for this purpose, but their popular text-based form has been rendered insecure by improvements in character recognition technology. In this paper, we propose a novel image-based CAPTCHA which employs object recognition as its test. Inspired by the negative selection approach in biological immune systems, an innovative two-phase filtering algorithm is proposed which ensures that the CAPTCHA is resilient to automated attack while remaining easy for human users to solve. In extensive testing involving over 3,000 participants, the proposed aiCAPTCHA achieved a 92.0% human success rate.**

***Index Terms*—CAPTCHA; mobile security; web security; object classification.**

I. Introduction

The Internet has become a critical part of modern human society. Many everyday activities, ranging from writing e-mails to conducting banking, rely on easily accessible and secure online services. If access to these services is disrupted, such as through a denial of service (DoS) attack, both consumers and service providers may incur significant loss of time, money, and resources. Many online services have adopted CAPTCHAs (Completely Automated Public Turing Tests to Tell Computers and Humans Apart) as part of a strategy for preventing misuse of online resources by automated attackers [1]. CAPTCHAs are tests designed to determine if the would-be user is human or a computer algorithm. They are an interactive security layer intended to be easy for humans to solve but challenging for computers.

The most common form of CAPTCHA is the text-based CAPTCHA, which requires users to decipher and input text from a visually distorted image [2]. One popular example is Google's reCAPTCHA [3]. Research has also been conducted into the design of image CAPTCHAs based on tasks such as identifying on image boundaries [4], recognizing faces [5] or biometric features [6], and conducting limited class object recognition such as distinguishing between dogs and cats [7]. However, as research on creating new CAPTCHAs has progressed, so has work on breaking CAPTCHAs [8], [9], [10] that has demonstrated vulnerabilities and weaknesses of the existing CAPTCHAs. A new solution that is not vulnerable to existing attack strategies (OCR, segmentation, object classification, pattern recognition) is needed to avoid further attacks.

Fig. 1. Example of an aiCAPTCHA test based on identifying black chairs.

This paper proposes aiCAPTCHA, a novel image-based CAPTCHA designed to avoid the weaknesses of existing approaches. It requires users to recognize specific object instances in a complex composite image. As shown in Fig. 1, aiCAPTCHA presents users with an image containing multiple photographs. Users are asked to select specific items, such as *black chairs*, by clicking (with computers) or tapping (with tablets and smartphones) on all instances of the specified item type. While this sort of object recognition task has been extensively studied in computer vision [11], [12], [13], existing solutions remain inferior to the human visual system and have difficulty in correctly recognizing the required objects when used with expansive object classes like those used by aiCAPTCHA. To further ensure that computers will be unable to solve aiCAPTCHA tests, the proposed approach incorporates a *negative selection-based artificial immune system* which identifies and removes CAPTCHAs which are susceptible to automated attack. While the resulting CAPTCHAs are difficult for automated attackers, testing with over 3,000 volunteers achieved a 92.0% human success rate in solving aiCAPTCHA tests.

DOI 10.1109/SPW.2017.22

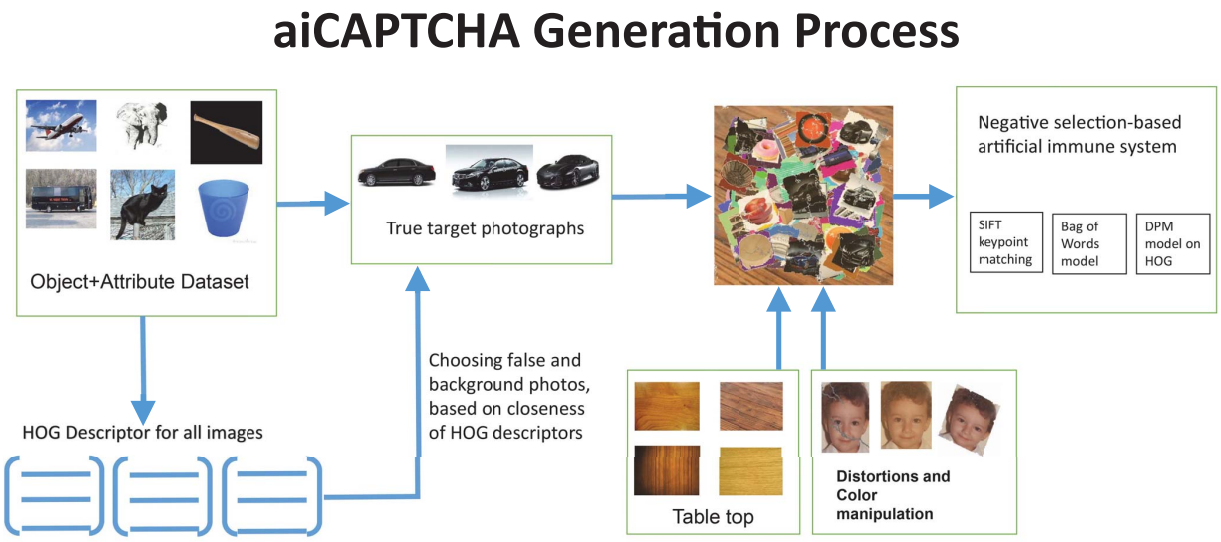

Fig. 2. An overview of the proposed aiCAPTCHA generation process.

II. Proposed aiCAPTCHA

The proposed aiCAPTCHA is based on the premise that humans can quickly recognize objects of interest in a cluttered background while automated algorithms struggle with this task because it requires the ability to segment and recognize objects by their class and attribute. Users are shown a composite image containing a stack of photographs of individual items. The photographed items represent 100 classes and 172 attributes (e.g., green tractors). The photographs are visually overlaid and distorted by adding ragged edges, simulated tears, and/or a dust effect. To solve the CAPTCHA, users must select all instances of a specified object type (e.g., round tables, red tomatoes) present in the aiCAPTCHA image. Users may make up to one mistake, either by failing to select an object that is present or selecting an object that was not specified, and still have their attempt counted as correct. Correct attempts are presumed to come from legitimate users and grant access to the resource aiCAPTCHA protects.

A. aiCAPTCHA Generation Process

Biological immune systems can distinguish between foreign cells and the body's own cells, popularly known as self-nonself discrimination [14], [15]. As mentioned by Dasgupta and Forrest [16], "*this discrimination is achieved in part by T-cells, which have receptors on their surface that can detect foreign proteins (antigens). T-cell receptors are made by a pseudo-random genetic rearrangement process, making it likely that some receptors will bind to self. Such self-reactive T-cells are censored in the thymus, with the result that only those cells that fail to bind to self proteins are allowed to leave the thymus and become part of the body's immune system.*" This concept is the key inspiration of the proposed aiCAPTCHA generation process. In the proposed approach, the CAPTCHAs that are susceptible to *attacks* are discarded and the ones which fail the *attack test* are used for distinguishing humans and computers apart.

The aiCAPTCHA generation process can be represented as

$$C = F(width, height, O, I, H, d, A) \quad (1)$$

where, function F represents the series of operations required to generate a new CAPTCHA of dimensions $width$-by-$height$ pixels. The required object type used for the CAPTCHA test is randomly chosen from O, the set of all tagged object classes and attributes. True target and false target photographs are taken from object-tagged set I. H represents precomputed Histogram of Oriented Gradient (HOG) descriptors for each photograph in I, an alternate representation of image data that can be used to identify similar images [17]. d represents a difficulty level that determines the distortions added to generated CAPTCHAs. A represents the attack algorithms to be used in conducting negative selection and deletion of vulnerable CAPTCHAs. The resulting attack-resistant generated aiCAPTCHA is C.

As shown in Fig. 2, a number of steps are involved in generation of aiCAPTCHA images. They are described in detail below.

1) Background Generation: Generation of a new aiCAPTCHA image begins with the creation of a background of size $width \times height$ pixels, where large sizes (at least 750×750) are used so the CAPTCHA has sufficient detail for use on high resolution displays. The background is designed to resemble a table top upon which a stack of photographs will be placed.

2) Target Object Type Selection: Once the background is generated, one object class-attribute combination (object type) $d_{selected}$ is randomly chosen from D to use for the CAPTCHA

test. This target object type will determine which photographs are embedded in the aiCAPTCHA image and will be provided to users in the instructions to solve the CAPTCHA.

3) True Target Photograph Selection: Using the target object type $d_{selected}$ that has been chosen, between 3 and 5 images are randomly chosen from the subset of corresponding photographs $I_{d_{selected}}$. These images P_{true} will be embedded in the resulting aiCAPTCHA image and will serve as true targets for users to select when solving aiCAPTCHA.

4) False Target Photograph Selection: Next, false target photographs are chosen to function as distractors for a would-be attacker. False targets are chosen by first calculating the Euclidean distance between the HOG descriptor for each photograph in P_{true} relative to each photograph in $I_{notselected} = \{I - I_{d_{selected}}\}$. Photographs with a smaller Euclidean distance in their HOG descriptors are more visually similar and are ideal for use in the CAPTCHA. A would-be attacker may be more likely to confuse one visually similar photograph for another and thus fail their attempt at solving aiCAPTCHA.

For each true target photograph in P_{true}, the 3 to 4 photographs in $I_{notselected}$ with the smallest Euclidean distance between their HOG descriptors are added to P_{false} to serve as false targets in the aiCAPTCHA image.

An additional 10 to 20 randomly selected photographs from $I_{notselected}$ are added to $P_{background}$ to provide a confusing background upon which P_{true} and P_{false} will be layered.

5) Photograph Preparation and Placement: After selecting $P_{background}$, P_{true}, and P_{false}, each photograph in these sets is rotated and scaled as follows:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x cos\theta & -s_y sin\theta & 0 \\ s_x sin\theta & s_y cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2)$$

Here, s_x and s_y are the scaling factors in the x and y directions respectively and θ is the clockwise rotation angle. The scaling parameters are determined based on the individual size of each photograph, and θ is varied randomly within a range carefully chosen to avoid extremely low or extremely high rotations. (x', y') denote the new coordinates for each pixel (x, y) of each photograph. The scaling allows for consistent sizing of each photograph (approximately 100×100 pixels) so it can be easily viewed. The rotation helps achieve the desired output in the form of a stack of photographs.

After scaling and rotation, the photographs may be subjected to the tear distortion depending on the specified difficulty level d. The tear distortion is applied to each photograph individually by randomly selecting one or more pairs of points on the image. For each of these pairs, one point is designated as the starting point (x_0, y_0) and the other is designated as the ending point (x_f, y_f). A number of lines, each denoted by T (initialized with the starting point (x_0, y_0)), are constructed between each starting and ending point as follows:

$$T = T \bigcup (\frac{x_f - x_t}{x_f} r_x(\cdot), \frac{y_f - y_t}{y_f} r_y(\cdot)) \quad (3)$$

Here, (x_t, y_t) denotes the latest point added to the line T, (x_f, y_f) denotes the ending point for this line. $r_x(\cdot)$ and $r_y(\cdot)$ both denote random functions that are either 0 or 1 but with the constraint that for each step $r_x(\cdot) + r_y(\cdot) \geq 1$. Therefore, at each step, Eq. 3 takes a one pixel step towards the ending point but it may choose from three possible directions. Each line T starting from the same pair of starting and ending points is added to the set of tear lines $\mathbf{T}$ and the pixels of the image for each point in this set ($(x, y) \in \mathbf{T}$) are modified as follows:

$$I'(x, y) = t + \delta, \ \delta \in [-v, +v] \quad (4)$$

where, $I'(x, y)$ denotes the new value for the pixel at location (x, y) for the particular image I. t denotes the base tear color (either gray or white) and δ is a small change in the range of $[+v, -v]$. v denotes the maximum permissible deviation from the base tear color.

Next, a random walk function is used to create a ragged edge effect around each photograph. Finally, the colors in the photographs are manipulated to make object recognition more challenging.

After all visual effects are added, each photograph in $P_{background}$ is placed layer by layer at random locations on the background generated in Step 1. Next, the photographs in P_{true} and P_{false} are placed in the top-most layer such that the true target P_{true} photographs are always completely visible to the user.

6) Global Distortion: Once the entire aiCAPTCHA image is generated, a dust effect is applied globally for certain values of difficulty level d. The goal of the dust effect is to make aiCAPTCHA harder to solve for automated scripts while adding a type of noise that is commonly seen by human eyes, i.e., the settling of dust. In order to emulate this effect, we divide the aiCAPTCHA image into regions and apply a blending effect to the pixels belonging to a region as follows:

$$I'(x, y) = w_i I(x, y) + w_d D \quad (5)$$

Here, $I'(x, y)$ and $I(x, y)$ denote the modified and original values of the pixel at location (x, y) of the aiCAPTCHA image respectively. D denotes the dust color that is set at (242, 168, 0) RGB. w_i and w_d are the weights for the original pixel and the dust color D used in the weighted sum-based blending approach such that $w_i + w_d = 1$ and w_d varies between 0.1 to 0.3 depending on the image region.

7) Negative Selection and Deletion of Vulnerable CAPTCHAs: To ensure the viability of aiCAPTCHA as a security tool, it is important that all images which are presented to users are resilient to automated attacks. Much as biological immune systems use a negative selection process to identify and eliminate immune cells which do not properly guard against foreign attackers, aiCAPTCHA employs an artificial immune system with its own negative selection process to identify and remove aiCAPTCHA images which may not successfully protect the guarded resource from automated attack [18]. Rather than use generated detectors as in a traditional negative selection algorithm [15],

aiCAPTCHA's process uses input from three distinct object recognition algorithms as simulated *attacks* to determine which aiCAPTCHA images should be removed:

1) SIFT (Scale-Invariant Feature Transform) keypoint-based matching [19], [20], which generates feature-based descriptions of images. The features generated from the aiCAPTCHA images are compared to already tagged image templates representing known objects using a 0.6 cosine similarity threshold. If the threshold is met, the object identified in the aiCAPTCHA is labeled.
2) Bag of Visual Words image classification [21], which constructs a sparse vector of histograms representing image features and then uses a Naive Bayes classifier to attempt to match those to the feature histograms of previously trained images.
3) Discriminatively-trained deformable part-based model (DPM) categorization [22], [23], which builds multiscale deformable models representing portions of images. Support vector machines are used to match the models generated from aiCAPTCHA images to previously trained known models of objects.

Simulated attacks are conducted against the generated aiCAPTCHA using each of the three algorithms. If any algorithm successfully identifies the objects in at least half of the true target photographs, the aiCAPTCHA image is considered defective and is deleted from the aiCAPTCHA database by the negative selection algorithm. This ensures that the resulting aiCAPTCHA images are resilient to adversarial external attacks. As shown by Table I, approximately 14% of generated aiCAPTCHA images were deleted by the negative selection algorithm while conducting this research. Fig. 3 shows examples of aiCAPTCHA images which have passed the negative selection attack process and are ready for use.

TABLE I
NUMBER OF AICAPTCHAS SOLVED BY ATTACKERS IN NEGATIVE SELECTION ARTIFICIAL IMMUNE SYSTEM WHILE GENERATING IMAGES

Generated CAPTCHAs	**860**
CAPTCHAs Solved by SIFT	33
CAPTCHAs Solved by Bag of Words	0
CAPTCHAs Solved by DPM	91
Defective Attackable CAPTCHAs Deleted from Database	**124**
Remaining Resilient CAPTCHAs for Public Use	**736**

III. EXPERIMENTAL RESULTS AND ANALYSIS

Generated aiCAPTCHA images have been tested by over 3,000 human participants. This section provides details of the source images, research participants, and protocol used in evaluating aiCAPTCHA along with results and analysis.

A. Image databases

In order to generate aiCAPTCHA images, a database with attribute-labeled images of various object categories is required. Since existing object databases are either: (a) not labeled with attributes, or (b) restricted to specific groups of objects such as animals, a new database was collected to support this research. Collection began with the creation of a list of object classes and associated attributes for each class. For example, cats may have the attributes "white," "brown," or "black" based on color, whereas books may be "open" or "closed" depending on their position. Overall, 100 classes with 172 attribute-based subclasses were identified.

The identified classes and attributes were used to generate search queries to retrieve images for each combination of object class and attribute. The retrieved images were manually filtered to remove images which did not accurately represent the intended object class and attribute. The resulting database contains a total of 7,765 tagged images.

The aiCAPTCHA database is initially populated with generated aiCAPTCHA images that can resist external attacks. This is determined by performing the negative selection filtering process using multiple attack algorithms. Over time, images in the aiCAPTCHA database which have a record of good human performance and user experience (UX) migrate to the aiCAPTCHA+UX database. The criteria to migrate a CAPTCHA to the aiCAPTHCA+UX database is that users must successfully solve each CAPTCHA 90% of the time over 10 attempts. This novel adaptive filtering mechanism ensures that CAPTCHAs in the aiCAPTCHA+UX database are both resilient to external attacks and provide an excellent user experience as quantitatively determined by user performance.

B. Participants and Testing Protocol

The proposed approach was evaluated by 3,135 volunteers using a set of 736 rendered aiCAPTCHA images. Volunteers attempted to access portions of a website that were protected by aiCAPTCHA. Participants were unsupervised and allowed to use their choice of browser and computing device (desktop computer, laptop, tablet, or smartphone). One aiCAPTCHA image was presented at a time. Users were asked to continue attempting to solve the CAPTCHAs until they were successful, at which point they gained access to the protected website.

C. Analysis

In total, 19,360 attempts to solve aiCAPTCHA were recorded with 7,360 attempts from CAPTCHAs in the aiCAPTCHA database and 12,000 attempts from CAPTCHAs in the aiCAPTCHA+UX database. As shown in Table II, humans achieved a 92.0% success rate (correct 23 of 25 times) when attempting CAPTCHAs in the aiCAPTCHA+UX database.

TABLE II
HUMAN SUCCESS RATES IN SOLVING CAPTCHAS

Database	**Number of CAPTCHAs**	**User Attempts**	**Success Rate**
aiCAPTCHA	736	7,360	80.7%
aiCAPTCHA+UX	425	12,000	92.0%

Fig. 3. Four examples of aiCAPTCHA images, with the true target photographs needed to solve the CAPTCHA outlined.

TABLE III
EVALUATING HUMAN SUCCESS RATES ON AICAPTCHA DATABASE IMAGES BY DIFFICULTY LEVEL

Difficulty Level	User Attempts	Success Rate
Level 1	1,770	82.8%
Level 2	1,770	79.3%
Level 3	1,920	81.3%
Level 4	1,900	79.0%

The tested aiCAPTCHAs come from four distinct difficulty levels, each with its own set of distortions applied during the generation process. These levels are determined by the application where the aiCAPTCHAs are deployed and the security needed. The difficulty levels are:

- Level 1: No dust or tear distortions
- Level 2: Dust distortions only
- Level 3: Tear distortions only
- Level 4: Both dust and tear distortions

Overall, the impact of the difficulty levels and distortions on human success rates was small. As shown in Table III, humans performed about 4% better on aiCAPTCHA database images with no distortions than those with both the tear and dust distortions. The effect of distortions on the automated attackers used in the negative selection algorithm was more pronounced. Adding the tear distortion to an aiCAPTCHA image cut the success rate of automated attackers by one-third.

The aiCAPTCHA generation process is designed to be resilient against attacks by conventional image classifiers through its artificial immune system. Unconventional and brute force attacks remain possible although our testing with best-of-breed approaches finds them unlikely to succeed. Fig. 4 illustrates the results of attempting one such unconventional attack, image recognition with Very Deep Convolutional Networks [24], on two aiCAPTCHA images. As the figure shows, the algorithm was unsuccessful in correctly identifying the objects of interest in the CAPTCHAs.

Brute force attempts, where an attacker selects random locations in the CAPTCHA, are similarly likely to fail. Each aiCAPTCHA contains 3-5 true target photographs. Since one target is allowed to be missed in a successful attempt for the sole purpose of an improved user experience, an attacker would need to correctly identify between 2 and 4 targets to solve the CAPTCHA. Each target is approximately 100×100 pixels in size, with the overall aiCAPTCHA image size being at least 750×750 pixels. Thus, the average chance of a single brute force attempt at correctly solving aiCAPTCHA is extremely small:

$$\left(\frac{1}{3}\right)\sum_{i=2}^{4}\left(\prod_{j=1}^{i}\frac{(100)(100)j}{(750)(750)}\right) = 0.0223\% \qquad (6)$$

Since a new image is chosen at random from a large set each time aiCAPTCHA is presented, a would-be attacker would likely have to wait hundreds of times before they will see the same aiCAPTCHA image again if their attempt is unsuccessful. When combined with the 2-in-10,000 chance of correctly solving the CAPTCHA on a given attempt, a would-be attacker would likely spend a significant amount of time using a brute force approach to successfully attack aiCAPTCHA.

IV. CONCLUSION

This paper presents the novel combination of an attribute-based object recognition CAPTCHA with a negative selection-based artificial immune system and two-phase filtering model that provides a security mechanism which is effective at preventing automated attacks without compromising the user experience. aiCAPTCHA has a high 92.0% human success rate for attempts on CAPTCHAs in the aiCAPTCHA+UX database, which is well above the 70%-80% rate of existing CAPTCHAs such as reCAPTCHA v1 and IMAGINATION [25], [26], and is designed to facilitate use on both traditional computers and touchscreen devices. When combined with its near zero likelihood of successful attacks, it offers significant advantages over CAPTCHAs commonly employed today.

APPENDIX

A working demonstration of aiCAPTCHA is available at http://aicaptcha.captcharesearch.com.

REFERENCES

[1] S. Shirali-Shahreza and M. H. Shirali-Shahreza, "Bibliography of works done on CAPTCHA," in *Proc. 3rd Int. Conf. on Intelligent System and Knowledge Engineering*, vol. 1, Xiamen, China, Nov. 2008, pp. 205–210.

Fig. 4. Deep learning strategies were unsuccessful at correctly identifying the objects embedded in aiCAPTCHA images to solve the CAPTCHAs. The labels atop each image above list the objects identified by Very Deep Convolutional Networks as being present in the image. None were accurate.

[2] S. K. Saha, A. K. Nag, and D. Dasgupta, "Human-Cognition-Based CAPTCHAs," *IT Professional*, vol. 17, no. 5, pp. 42–48, Sep. 2015.

[3] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "reCAPTCHA: Human-Based Character Recognition via Web Security Measures," *Science*, vol. 321, no. 5895, pp. 1465–1468, Sep. 2008.

[4] R. Datta, J. Li, and J. Z. Wang, "IMAGINATION: a robust image-based CAPTCHA generation system," in *Proc. 13th Annual ACM International Conference on Multimedia*, Singapore, Nov. 2005, pp. 331–334.

[5] G. Goswami, B. M. Powell, M. Vatsa, R. Singh, and A. Noore, "FR-CAPTCHA: CAPTCHA Based on Recognizing Human Faces," *PLoS ONE*, vol. 9, no. 4, p. e91708, Apr. 2014.

[6] B. M. Powell, A. Kumar, J. Thapar, G. Goswami, M. Vatsa, R. Singh, and A. Noore, "A Multibiometrics-based CAPTCHA for Improved Online Security," in *Proc. IEEE 8th Int. Conf. on Biometrics: Theory, Applications and Systems*. Niagara Falls, New York: IEEE, Sep. 2016.

[7] J. Elson, J. Douceur, J. Howell, and J. Saul, "Asirra: a CAPTCHA that Exploits Interest-Aligned Manual Image Categorization," in *Proc. 14th ACM Conf. on Comput. and Commun. Security*, Alexandria, Virginia, Oct. 2007, pp. 366–374.

[8] P. Baecher, N. Buscher, M. Fischlin, and B. Milde, "Breaking re-CAPTCHA: A Holistic Approach via Shape Recognition," in *Future Challenges in Security and Privacy for Academia and Industry*, J. Camenisch, S. Fischer-Hbner, Y. Murayama, A. Portmann, and C. Rieder, Eds. Berlin, Germany: Springer, 2011, vol. 354, pp. 56–67.

[9] G. Mori and J. Malik, "Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA," in *Proc. 2003 IEEE Comput. Society Conf. on Comput. Vision and Pattern Recognition*, vol. 1, Madison, Wisconsin, Jun. 2003, pp. 134–141.

[10] J. Yan and A. S. El Ahmad, "Breaking Visual CAPTCHAs with Naive Pattern Recognition Algorithms," in *Proc. 23rd Annu. Comput. Security Applicat. Conf.*, Miami Beach, Florida, Oct. 2007, pp. 279–291.

[11] V. Nair and G. E. Hinton, "3d Object Recognition with Deep Belief Nets," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 1339–1347.

[12] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust Object Recognition with Cortex-Like Mechanisms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 411–426, Mar. 2007.

[13] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *Proc. 10th IEEE Int. Conf. on Comput. Vision*, vol. 2, Beijing, China, Oct. 2005, pp. 1800–1807.

[14] F. Esponda, S. Forrest, and P. Helman, "A formal framework for positive and negative detection schemes," *IEEE Trans. Sys. Man Cyber. Part B*, vol. 34, no. 1, pp. 357–373, Feb. 2004.

[15] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, "Self-Nonself Discrimination in a Computer," in *Proc. 1994 IEEE Symp. on Security and Privacy*, Oakland, California, May 1994, pp. 202–212.

[16] D. Dasgupta and S. Forrest, "Novelty detection in time series data using ideas from immunology," in *Proc. ISCA 5th Int. Conf. on Intelligent Systems*, Reno, Nevada, Jun. 1996, pp. 82–87.

[17] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proc. 2005 IEEE Comput. Society Conf. on Comput. Vision and Pattern Recognition*, San Diego, California, Jun. 2005, pp. 886–893.

[18] J. Brownlee, *Clever Algorithms: Nature-Inspired Programming Recipes*. Raleigh, North Carolina: Lulu.com, 2012.

[19] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. Journal of Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[20] ——, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. on Comput. Vision*, vol. 2, Kerkyra, Greece, Sep. 1999, pp. 1150–1157.

[21] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. 8th European Conf. on Comput. Vision, Workshop on Statistical Learning in Comput. Vision*, vol. 1, Prague, Czech Republic, May 2004, pp. 1–2.

[22] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[23] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester, "Discriminatively trained deformable part models, release 5," Sep. 2012. [Online]. Available: http://people.cs.uchicago.edu/ rbg/latent-release5/

[24] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs]*, Sep. 2014.

[25] E. Bursztein, S. Bethard, C. Fabry, J. C. Mitchell, and D. Jurafsky, "How Good Are Humans at Solving CAPTCHAs? A Large Scale Evaluation," in *Proc. 2010 IEEE Symp. on Security and Privacy*, Los Alamitos, California, 2010, pp. 399–413.

[26] R. Datta, J. Li, and J. Z. Wang, "Exploiting the Human-Machine Gap in Image Recognition for Designing CAPTCHAs," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 3, pp. 504–518, 2009.