Symmetry reCAPTCHA

Christopher Funk Yanxi Liu
School of Electrical Engineering and Computer Science. The Pennsylvania State University
University Park, PA. 16802, USA
{funk, yanxi}@cse.psu.edu

Abstract

This paper is a reaction to the poor performance of symmetry detection algorithms on real-world images, benchmarked since CVPR 2011. Our systematic study reveals significant difference between human labeled (reflection and rotation) symmetries on photos and the output of computer vision algorithms on the same photo set. We exploit this human-machine symmetry perception gap by proposing a novel symmetry-based Turing test. By leveraging a comprehensive user interface, we collected more than 78,000 symmetry labels from 400 Amazon Mechanical Turk raters on 1,200 photos from the Microsoft COCO dataset. Using a set of ground-truth symmetries automatically generated from noisy human labels, the effectiveness of our work is evidenced by a separate test where over 96% success rate is achieved. We demonstrate statistically significant outcomes for using symmetry perception as a powerful, alternative, image-based reCAPTCHA.

1. Introduction

With increasing malicious online attacks against companies, individuals and governments alike, CAPTCHA (**C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part) [1, 41] becomes a necessary tool in many web applications to prevent automatic login, blog post, email, and DoS (denial-of-service) attacks. CAPTCHAs are constructed by taking advantage of the behavioral discrepancies between humans and machines. For example, since humans are in general superior at recognizing distorted text than computers, text-based CAPTCHAs such as [42] have been widely used. **re**CAPTCHAs, where one word is used for testing and the other for dataset expansion, can also be implemented and deployed easily. Meanwhile, successful breakings of existing CAPTCHAs (e.g. [2, 27, 29, 37]) have been extensively reported as well, including a recent self-broken reCAPTCHA by Google [17].

We are witnessing a healthy interleaving of more advanced CAPTCHAs that provide better software security and breakings of CAPTCHAs which advance machine intelligence. In recent years, automatic text recognition is closing the gap to human performance [17], thus more image-based CAPTCHAs have emerged as an alternative [3, 4, 7, 8, 13, 14, 18, 28, 30, 31, 32, 41]. Most current image-based CAPTCHAs focus on recognition of common objects, such as cats, dogs, human faces or relating objects (with the help of words) semantically. The primary challenges for computers come from image distortions and object semantics. Little work has explored and taken advantage of human ability in visual abstraction, for example, the perception of symmetry from photos (e.g. Figure 1).

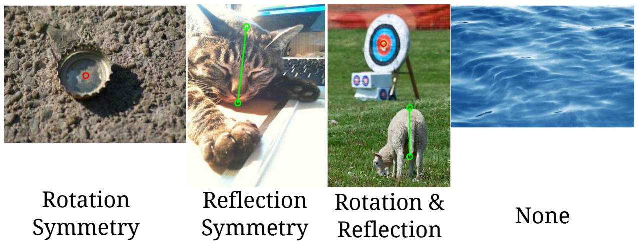


Figure 1. Human labeled symmetries on real photos. Red: rotation center. Green: reflection axis.

Strong evidence of inherent symmetry detection capability in humans has been shown at both behavioral and neural levels [5, 6, 19, 21, 35, 40], while the outcome from the 2011/13 CVPR symmetry detection competitions [23] is less encouraging. This performance gap between humans and machines presents an opportunity for a symmetry-perception-based CAPTCHA. To the best of our knowledge, no systematic elicitations of human understanding of reflection and rotation symmetries on a large set of real-world photos has been reported, nor has the gap of symmetry perception between humans and computers been well-defined or quantified. This leads to our specific contributions in this work. We provide:

- via crowdsourcing, a systematic collection and evaluation of human symmetry labels from 401 online raters on 1,200 photos;
- a well-defined set of symmetry metrics and algorithms to determine human perceived *Ground Truth* symme-

tries from noisy human labels;

- a systematic quantification of the *gap* between the human and machine symmetry perception in terms of precision and recall rates;
- a prototype image-based *symmetry reCAPTCHA*; and
- a validation of the proposed reCAPTCHA on 118 human online-users with a $> 96\%$ success rate.

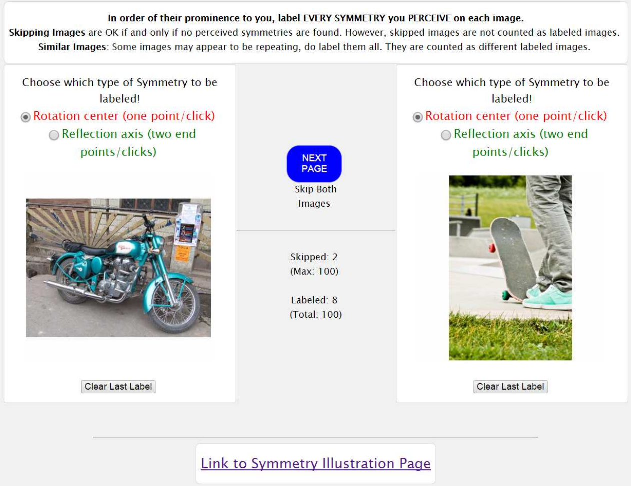
2. Related Work

reCAPTCHA[42] is one of the most well-known CAPTCHAs and is widely used on the Internet today. It combines labeling new ground truth while simultaneously acting as a CAPTCHA. reCAPTCHA uses text recognition as the challenge and provides each rater two words. One word where the transcription is known is used as the CAPTCHA; the other word is unknown. The unknown words are found by two separate OCR algorithms transcribing books. A consensus has to be reached before the word is added into the new challenge word set. The reCAPTCHA system can detect humans with a success rate of 96.1%, but with the steady advancement of OCR [17, 37, 42] in recent years, it has become less successful at rejecting computers.

Researchers have investigated various forms of CAPTCHAs based on natural image understanding. These can be divided into (1) object recognition-based and (2) 3D model-based (e.g. orientation judgment).

Simple image recognition CAPTCHAs [3] present raters with undistorted generic images to be labeled using the provided word lists. ESP-PIX asks raters to recognize what object is common in a set of images [14]. ARTiFACIAL exploits the face detection gap between human and algorithms. It provides synthesized images with a distorted face embedded in a cluttered background and raters are requested to first find the face and then click on four eye corners and two mouth corners on the face [32]. Avatar CAPTCHA uses a similar task where the raters are required to find a synthetic faces among a table of faces, although computers have outperformed humans at the task [12, 20]. Asirra is based on cat/dog classification; it presents images of 12 cats and dogs and asks raters to identify the cats among them [13]. Golle [16] shows that Asirra is vulnerable to machine learning attacks since it is possible to train a classifier that can identify cats and dogs with high accuracy. PI-CATCHA [30] asks the rater to identify a type of object, such as books, balls, or buildings, from 8 listed images. Datta *et al*. [8] explored further the use of systematic image distortion in designing CAPTCHAs and found that combining multiple atomic distortions can significantly reduce machine recognizability [39]. Mitra *et al*. [26] propose "emergence images" rendered with noise and clutter from 3D models as a potential source of CAPTCHA.

The second type of image-based CAPTCHAs relies on an understanding of orientations of 3D models. Gossweiler *et al*. [18] introduced the idea of correcting image orientation for designing the "What's Up" CAPTCHA; it uses images drawn from popular web searches as a potential database and asks the humans to correct the image's orientation. Ross *et al*. [31] presented the Sketcha CAPTCHA that requires raters to determine the upright orientation for a selection of 3D objects rendered as line drawings. Sketcha provides the rater with only four orientation options compared with a continuous rotation from "What's Up".



In order of their prominence to you, label EVERY SYMMETRY you PERCEIVE on each image.
Skipping Images are OK if and only if no perceived symmetries are found. However, skipped images are not counted as labeled images.
Similar Images: Some images may appear to be repeating, do label them all. They are counted as different labeled images.

Choose which type of Symmetry to be labeled!
● Rotation center (one point/click)
○ Reflection axis (two end points/clicks)

Skipped: 2
(Max: 100)

Labeled: 8
(Total: 100)

Choose which type of Symmetry to be labeled!
● Rotation center (one point/click)
○ Reflection axis (two end points/clicks)

Figure 2. The symmetry labeling interface (Section 3.2). Best viewed online.

Many crowdsourcing image annotation tools have been developed for object segmentation and image labeling [9, 10, 11, 33, 34, 38], while little is available for labeling symmetry directly on photos via crowdsourcing, which is a new tool we develop in this work. Furthermore, our work differs from previous image-based CAPTCHAs in utilizing symmetry as a cue for CAPTCHA, which is object-class, scale, shape, orientation and color independent. Our experiments demonstrate that a symmetry-concept-based CAPTCHA can be understood and utilized by human subjects across culture, age, gender, and education levels.

Symmetry detection has been a lasting research topic in computer vision and computer graphics [24]. From the benchmarked outcome of 2011/2013 CVPR *symmetry detection from real-world images* competitions [23], the algorithm developed by Loy and Eklundh [25] has been shown to yield consistently the best performance on both reflection and rotation symmetry detection. The algorithm is fast, requires no image-segmentation, and recognizes reflection and rotation symmetries respectively from extracted SIFT keys via an effective voting scheme. During these competitions, [25] has been compared against dozens of algorithms. Thus far, we have not found newer work that surpasses [25] consistently. Thus [25] is chosen to be the representative for computer symmetry detection algorithms in our evaluation of machine perception of real world symmetries.

3. Our Approach

3.1. Image Data Initialization

Our dataset consists of 1,200 images from the Microsoft COCO database[22]. To be included in our initial image data set, each image must satisfy two conditions (through visual inspection by undergraduate/graduate students): (1) the image presents some kind of visual symmetry and (2) the symmetry detection algorithm [25] fails on detecting the most prominent symmetry in the image. Before presenting the images to human raters, each image is scaled to 400 pixels (largest dimension) to permit two images to fit on most computer screens.

3.2. Mechanical Turk Labeling Tool

We have designed and implemented a graphical interface for human raters to enter their choice of perceived real-world symmetries on an image (Figures 1 and 2). After a short introduction on the general concept of symmetry with visual examples, each Amazon Mechanical Turk rater must pass a training session that consists of labeling four images (Figure 1).

The user interface (Figure 2) guides the raters to label a rotation or a reflection symmetry. Once a type of symmetry is selected, they can identify a rotation center by one click or a reflection axis by clicking the two end points of a line segment. The rater has the choice of either identifying at least one perceived symmetry or skipping either image or both images. The task requires the rater to label 100 images total and can skip at most 100 images during the experiment. The ability to skip an image enables the rater to not be forced to label a symmetry unless he or she perceives one. The raters are asked to label the symmetries within the image according to perceived prominence. Each rater is given at least 90 unique images and 10 repeats. The repeats are used to determine the rater's reliability. We have collected 78,310 rater labeled symmetries from 401 raters. Statistical information on data collection is shown in Table 1. We define a Skip-Label Score (**SL-Score**) to reflect the number of times each image is skipped due to a lack of perceived symmetries *and* is labeled due to a perceived symmetry. See Figure 3 for a distribution of images in the SL-score space.

3.3. Symmetry Distance Metrics

Our data is a collection of human inputs in the form of labeled symmetries on an image, either for rotation symmetry centers (one point) or for reflection symmetry axes (two points). An intriguing research question as well as an engineering necessity is *HOW* to group such labels into their intended semantic meanings: *symmetry* **X** *is at this position of this image*. A key theoretical basis for such a grouping task is the *distance metric(s)* that can measure the **similarities** among different human-perceived symmetry labels.

Total # of Initial Images	1,200
Total # of Images after Screening	961
Total # of Human Raters	401
Average # of Labelers / Image ± std	33(±3)
Average # Skippers / Image ± std	3(±3)
Total Rotation Symmetries Labeled	26,374
Total Reflection Symmetries Labeled	51,936
Average Symmetries Labeled / Image ± std	81(±34)
Average Rotation Symmetries Labeled / Image ± std	27(±27)
Average Reflection Symmetries Labeled / Image ± std	54(±22)

Table 1. Statistics of the data used in this paper, pertaining to the images, human raters, and labeled symmetries.

3.3.1 Definitions of Symmetry Distance

We define the *rotation symmetry distance* $D_\circ$ as the Euclidean distance between the two labeled rotation symmetry centers. Given two reflection axes (L_1 and L_2), the *reflection symmetry distance* between them is a measure comparing two line segments with potentially different lengths, positions and orientations.

We propose four different reflection symmetry measures for a pair of reflection symmetry axes: d_a (end points), d_b (mid points), d_c (mid-point to line), and d_d (angle) distances as defined and illustrated in Figure 4.

Previous work has observed that the *closeness* of two line segments is proportional to their lengths, i.e., two 10-foot lines 1-inch apart may be perceived as closer than two 1-inch long line segments 1-inch apart.[23]. We thus weigh our reflection symmetry distance measures d_a, d_b, d_c with the lengths of the corresponding reflection axes by defining:

$$R = \frac{|L_1| + |L_2|}{2} \tag{1}$$

$$D_a = \frac{d_a}{R}, D_b = \frac{d_b}{R}, D_c = \frac{d_c}{R}, D_d = d_d. \tag{2}$$

3.3.2 Distance Distribution of Nearest Symmetries

Given a pair of human labels, we need to find out: *are they labeling the same symmetry*? In addition to a proper set of symmetry distance measures (Section 3.3.1), we need a *membership* algorithm to determine whether two labels belong to the same group. We build statistical distributions of *nearest neighbors* of all labeled symmetries for each symmetry distance (Figure 5). To determine the membership threshold τ automatically, we algorithmically discover the

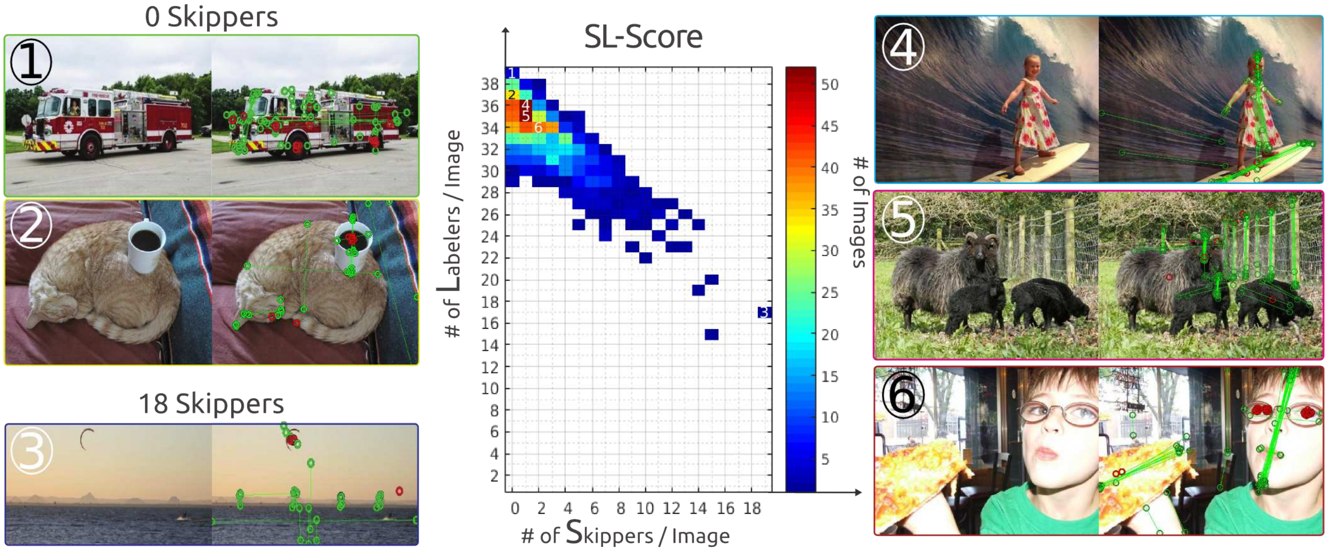


Figure 3. The average number of labelers per image is 33 and the average number of skippers per image is 3. Images labeled 1, 2, and 3 (left column) represent those images without any skips by any rater and images labeled 4, 5, and 6 (right column) represent images around the peak of the distributions of the SL-images). The different colors of the image numbers are for ease of viewing. Each image pair shows the input image and the image with all labeled symmetries. This figure is best viewed in color.

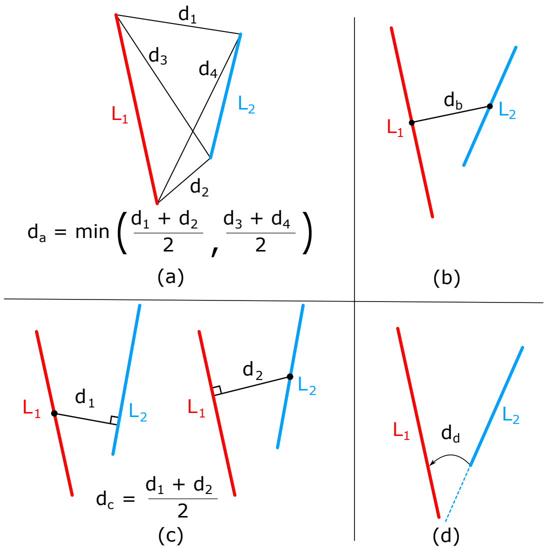


Figure 4. The four different reflection symmetry distance measures between two reflection symmetry axes L_1 and L_2; (a) d_a measures the minimum mean distance between the corresponding end points of L_1 and L_2; (b) d_b measures the distance between the midpoints; (c) d_c measures the average shortest distance between their midpoints and the other line; and (d) d_d measures the angle difference between the two line segments.

'knee-of-the-curve' for the distribution curves (Figure 5), where the 'knee' is the point at which a function has the maximum curvature and where the curve starts to grow exponentially. We use the Kneedle algorithm by Satopää *et al*. [36] since it is a general knee-of-the-curve finding algorithm and meets our needs of adaptive thresholding based on human perceptual input. The neighbors with distance beyond τ can be considered as symmetry labels for a different symmetry. In the case of reflection symmetry, the four different thresholds have to work jointly to determine the membership (Section 3.4).

3.4. Ground Truth Extraction

The ground truths (GTs) rotation or reflection symmetry in an image is computed from a consensus of human labels based on each rater's perception of a real-world symmetry in that image. To obtain a set of symmetry GTs objectively and computationally, we adapt DBSCAN [15], a method for **D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise (the winner of the test-of-time award in 2014), with special considerations for perceptual symmetries by humans. Different from K-means, DBSCAN does not need the number of clusters as input. DBSCAN is a local-neighborhood distribution-based method whose inputs are the minimum distance ϵ of a neighbor for being a *member* of the cluster and the minimum number of neighbors *minPts* to be a dense region or cluster. In our case, we have automatically obtained symmetry distance threshold τ for nearest-neighbors under different symmetry distance metrics (Section 3.3.2, Figure 5), and for each labeled symmetry to be considered as a GT symmetry we expect that at least two unique raters have labeled it independently. Here we have to enforce the independence condition for each cluster to containing a unique set of raters since the same rater may label a symmetry twice when given the same image as a repeat to test rater consistency (Section 3.2). Therefore, we have $\epsilon = \tau$ and $minPts = 2$ as the input parameters for

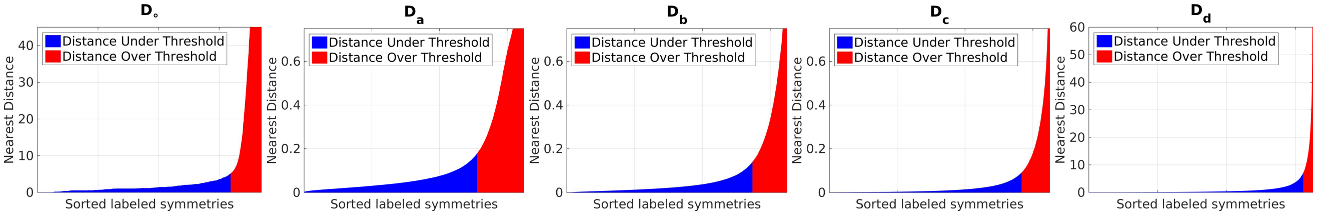


Figure 5. The nearest symmetry distance distributions (D_o for rotation symmetry, D_a, D_b, D_c, D_d for four reflection symmetry distance measures defined in Figure 4) sorted from low to high (left to right). The 'knee point' τ is automatically determined using the Kneedle algorithm by Satopää *et al.* [36]. The 'knee' is the point at which a function has the maximum curvature and where the curve starts to grow exponentially. The automatically found threshold for rotation symmetry labels is 5 (pixels); the four thresholds for reflection symmetry labels are: 0.18 for D_a, 0.14 for D_b, 0.09 for D_c and 7 (degrees) for D_d. Note: D_a, D_b, D_c are normalized by the average lengths of the pair of reflection axes under consideration (Equations 1, 2).

DBSCAN.

Figure 6 demonstrates a set of sample GTs extracted automatically from rater labels. To illustrate the human perceived level of prominence of a labeled symmetry, the radii of the rotation symmetry centers and the thickness of the reflection axes shown in Figure 6 are proportional to the number of raters who have labeled that particular symmetry. The circular contour around each point (rotation center, end points of a reflection axis) indicates the location uncertainty of the labels. Table 2 shows the statistics of the automatically extracted GTs. Using this set of automatically computed reflection and rotation symmetry GTs, we re-evaluate the admissible image set to ensure that the algorithm [25] can not pass as human. This step reduces the initial dataset size from 1200 images to 961 images (Table 1).

Total # of GT	8,146
Total # of Rotation GT	2,704
Total # of Reflection GT	5,442
Total # of Images with Rotation GT	836
Total # of Images with Reflection GT	961
Average # of Reflection GT/Image	6 ± 3
Average # of Rotation GT/Image	3 ± 3

Table 2. Statistics of the automatically extracted Ground Truth Symmetries from the raters' labels.

4. Quantitative Evaluation and Comparison

After the construction of both a user interface for capturing human perceived symmetries and a computational method for extracting GT symmetries, we validate that our method is indeed achieving the intended goals:
(1) On the data from 400+ anonymous online raters, we quantify the human and machine performance on the same image set (961 images) in terms of their precision and recall rate under the variations of three parameters that are, respectively; rotation-symmetry threshold, reflection-symmetry threshold and the minimum number of symmetry labels required for a symmetry GT.
(2) On a new set of human raters, excluding those who participated in (1), we perform a validation test of "are you human or machine?" using a subset of the 961 images.

4.1. Performance of Human and Machine

True Positive: Given a labeled symmetry A (either a rotation or a reflection), a GT symmetry, a symmetry distance D_m, its corresponding threshold τ_m (Section 3.3.2), and B to represent all more prominent symmetry labels from the same rater on the same image where prominence is defined by the rater's ordering:

$$\text{TP} = \begin{cases} 1 & \text{if } \exists\, GT : (\forall\, m : D_m(GT, A) < \tau_m) \\ & \quad \neg\exists\, B : (\forall\, m : D_m(GT, B) < \tau_m) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

False Positives:

$$\text{FP} = \begin{cases} 1 & \text{if } \neg\exists\, GT : (\forall\, m : D_m(GT, A) > \tau_m) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

False Negatives:

$$\text{FN} = \begin{cases} 1 & \text{if } \neg\exists\, A : (\forall\, m : D_m(GT, A) < \tau_m) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

We vary three parameters to create the Precision-Recall (PR) curves. They are: rotation center distance threshold, reflection axis distance thresholds and the minimum number of labeled symmetries for defining a GT symmetry. We use the τ values automatically discovered in Section 3.3.2 (Figure 5) as a basis, and increase the threshold values up to 5 times the base value. A comparison of PR-surfaces and the differences between the human and the machine *et al.* [25] are shown in Figures 7.

4.2. Symmetry reCAPTCHA Validation

To evaluate the effectiveness of the symmetry reCAPTCHA mechanism, we automatically select a subset

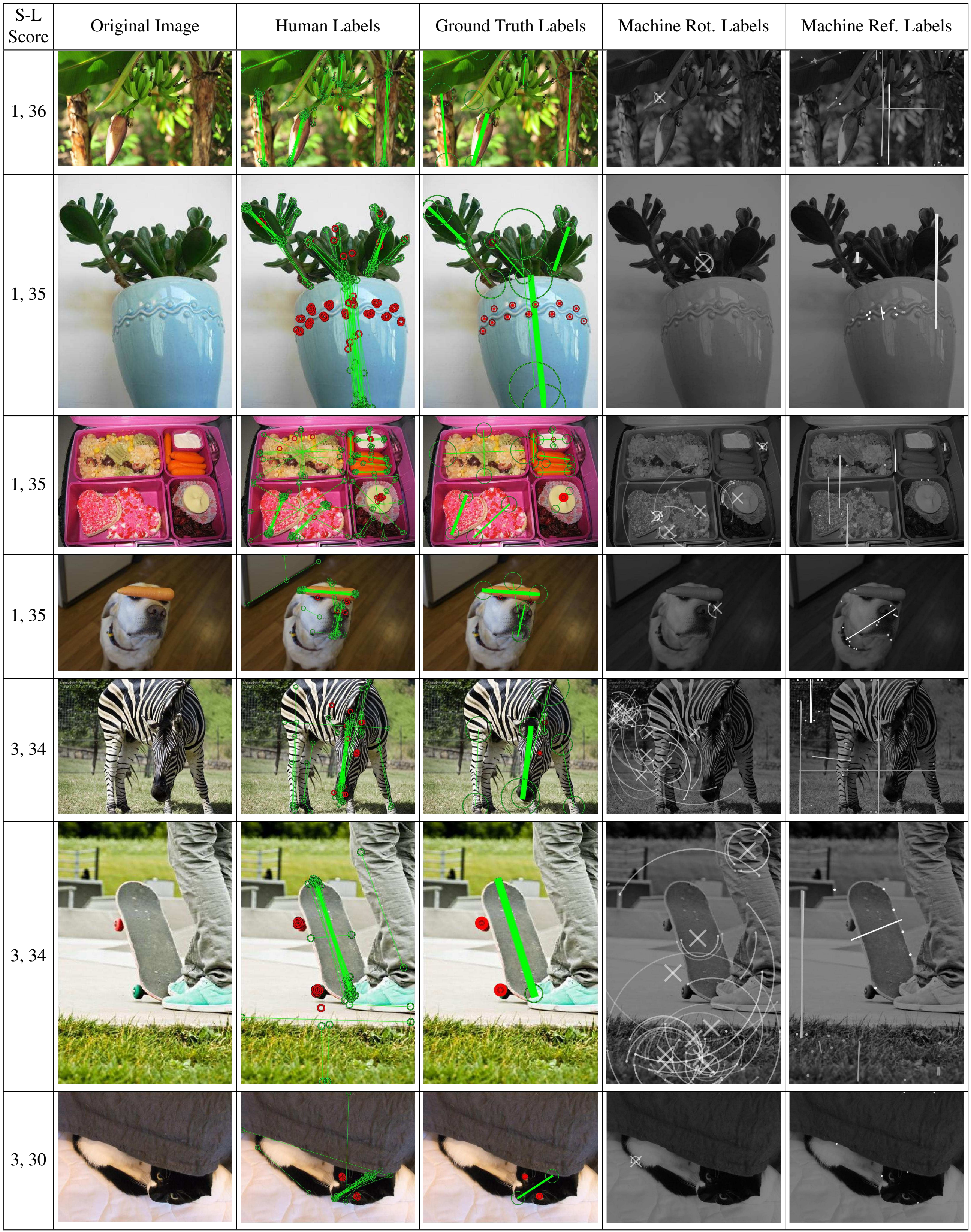


Figure 6. Seven randomly selected sample images where we index each image with its SL-score (skip-label score from Figure 3) and show the original image, all human labels on the image, the symmetry GTs automatically extracted, followed by the output of [25] showing the detected rotation and reflection symmetries where the detected symmetries are ranked from bright (top choice) to dark in intensities. This figure is best viewed on the computer.

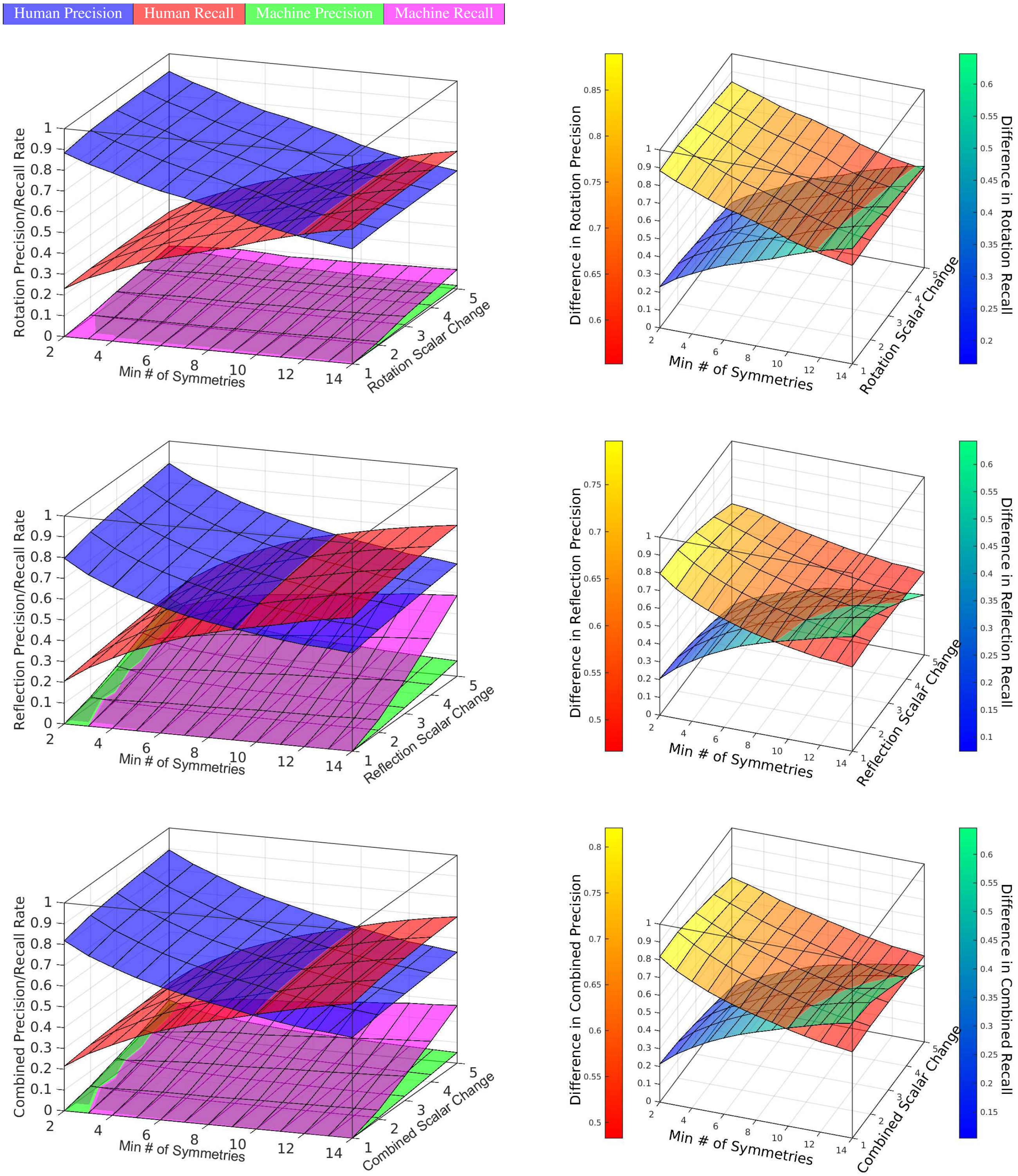


Figure 7. The performance PR-surfaces of all human raters (401) in comparison to those of a computer algorithm[25] on all 961 images. The three varying parameters are: (1) rotation symmetry center distance threshold; (2) reflection symmetry distance thresholds (four); and (3) the minimum number of symmetries required for a symmetry GT to be defined. With the increasing of minimum labeled symmetries required for a symmetry GT, one can observe a tradeoff between the human PR-curves due to down of symmetry GTs and up of FPs.

Total # of Test Images	40
Total # of Human Testers	118
Average # of Labelers / Image ± std	115(±3)
Average # Skippers / Image ± std	3(±3)
Total Rotation Symmetries Labeled	4,118
Total Reflection Symmetries Labeled	6,111
Average Symmetries Labeled / Image ± std	256(±83)
Average Rotation Symmetries Labeled / Image ± std	103(±57)
Average Reflection Symmetries Labeled / Image ± std	152(±53)

Table 3. Table of statistics on the performance of the prototype symmetry reCAPTCHA.

of test images which are most discriminative between human and machine perception of symmetries. Each image is selected based on human/machine performance evaluation (Section 4.1) according to the following criteria: (1) the computer algorithm fails to find any GT symmetries; (2) the total number of raters who successfully labeled at least one symmetry on the image (the higher the total, the more prominent the symmetry appears to humans); (3) the success rate (total TPs / total GTs) of human raters on the image; and (4) the number of human raters who failed to label the correct GT symmetry on the image: TP=0 while FP $\neq 0$. Based on these conditions, a total of 40 test images are chosen. By pairing each test image with a non-test image, we constructed a symmetry reCAPTCHA. In practice, the system would have a limitless supply of photos from the internet, labeled by users during reCAPTCHA, creating an ever moving target of symmetry image set.

A new set of human raters (exclusive from Section 4.1) are presented with our prototype symmetry reCAPTCHA, whose labels are used to determine whether the labeler is a human and compute the overall success rate (Table 3). For 118 new human raters, our symmetry reCAPTCHA prototype achieves a success rate of 96.31%, 95.05% and 95.81% using rotation symmetries alone, reflection symmetries alone and jointly, respectively. Using a one-sample t-test, with the null hypothesis that the human test result comes from a normal distribution with zero mean (or the same as machine performance), leads to p-value $= 0$.

5. Summary and Discussion

We have explored taking advantage of the discrepancy of symmetry perception between humans and machines to build a symmetry-based reCAPTCHA. We have carefully designed, implemented, and tested a user interface to solicit input of human-perceived imperfect symmetries on real-world photos from the MS COCO database. Our methods objectively and systematically extract user labeled symmetry ground truth as statistical consensus or viable clusters in the proper symmetry distance spaces. A 2-stage quantitative evaluation has shown that human performance on visual symmetry selection from real-world photos is indeed significantly superior to the best proven computer algorithm [25] thus far. A prototype symmetry reCAPTCHA generates promising initial results. In their extensive survey of image-based CAPTCHAs, Zhu *et al.* [43] proposed three guidelines for image-based CAPTCHAs. The first is to rely on the unambiguous semantics of the images. Symmetry reCAPTCHA relies on symmetry detection which is innate for humans across different cultures, and our experimental results support the observation that human symmetry perception is well-clustered and computable. The second criteria is allowing large variations: symmetry is a phenomenon independent of object class, shape, size and color. The last desirable criteria is the ease of making the image perceptions harder. Given the types and variances of possible symmetries (simple to complex, global to local, small to large ...), real-world symmetry detection from noisy data can be continuously challenged. Even without any intentional manipulations of the images, [25] fails to find any ground truth symmetries from human labels of the 961 photos from COCO database selected in this work. When comparing performance of humans versus machines on symmetry detection, we observe that human subjects are less distracted by clutters around real world symmetry objects (e.g. face, half of a moon, eyes of a dog); while the machine tends to have issues with near regular textures in the background or recurring patterns in the foreground. Since our focus is on rotation or reflection symmetries (excluding translation symmetry), the computer vision algorithm often fails to capture the real symmetry in a crowded image. Another common failure of machine is when the plane of symmetry is not parallel with the camera plane. These issues are illustrated in Figure 6.

Real-world symmetries have presented computational challenges to computer vision for decades. We hope to motivate more research in this direction such that better algorithms can break symmetry reCAPTCHA in the near future. Meanwhile, we continue our effort on learning how humans perceive noisy symmetries. A better understanding of human perception at both the behavioral and neural levels will help our design of more robust algorithms for symmetry detection. We will release the full dataset of images to help others test and improve their symmetry algorithms.

6. Acknowledgement

This work is supported in part by an NSF CREATIV grant (IIS-1248076).

References

[1] L. V. Ahn, M. Blum, N. J. Hopper, and J. Langford. CAPTCHA: Using Hard AI Problems for Security. pages 294–311, 2003. 1

[2] K. Chellapilla and P. Y. Simard. Using Machine Learning to Break Visual Human Interaction Proofs (HIPs). *Advances in neural information processing systems*, 17:265–272, 2005. 1

[3] M. Chew and J. D. Tygar. Image Recognition CAPTCHAs. *Information Security: 7th International Conference, ISC 2004, Palo Alto, CA, USA, September 27-29, 2004. Proceedings*, pages 268–279, 2004. 1, 2

[4] R. Chow, P. Golle, M. Jakobsson, L. Wang, and X. Wang. Making CAPTCHAs Clickable. In *Proceedings of the 9th Workshop on Mobile Computing Systems and Applications*, HotMobile '08, pages 91–94, New York, NY, USA, 2008. ACM. 1

[5] A. D. Clarke, F. Halley, A. J. Newell, L. D. Griffin, and M. J. Chantler. Perceptual Similarity: A Texture Challenge. In *Proceedings of the British Machine Vision Conference*, 2011. 1

[6] R. W. Conners and C. T. Ng. Developing a quantitative model of human preattentive vision. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(6):1384–1407, 1989. 1

[7] R. Datta, J. Li, and J. Z. Wang. IMAGINATION: A Robust Image-based CAPTCHA Generation System. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, pages 331–334, New York, NY, USA, 2005. ACM. 1

[8] R. Datta, J. Li, and J. Z. Wang. Exploiting the Human-Machine Gap in Image Recognition for Designing CAPTCHAs. *IEEE Transactions on Information Forensics and Security*, 4(3):504–518, 2009. 1, 2

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 2

[10] J. Deng, K. Li, M. Do, H. Su, and L. Fei-Fei. Construction and analysis of a large scale image ontology. In *Vision Sciences Society*, volume 186, 2009. 2

[11] J. Deng, O. Russakovsky, J. Krause, M. S. Bernstein, A. Berg, and L. Fei-Fei. Scalable Multi-label Annotation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 3099–3102, New York, NY, USA, 2014. ACM. 2

[12] D. D'Souza, P. C. Polina, and R. V. Yampolskiy. Avatar CAPTCHA: Telling Computers and Humans Apart via Face Classification. In *Electro/Information Technology (EIT), 2012 IEEE International Conference on*, pages 1–6. IEEE, 2012. 2

[13] J. Elson, J. R. Douceur, J. Howell, and J. Saul. Asirra: A CAPTCHA that Exploits Interest-aligned Manual Image Categorization. *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pages 366–374, 2007. 1, 2

[14] ESP-PIX. Esp-pix. http://www.captcha.net/. 1, 2

[15] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Knowledge Discovery and Data Mining (KDD)*, volume 96, pages 226–231, 1996. 4

[16] P. Golle. Machine Learning Attacks Against the Asirra CAPTCHA. *Proceedings of the 15th ACM Conference on Computer and Communications Security*, pages 535–542, 2008. 2

[17] I. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet. Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks. In *International Conference Learning Representations, (ICLR2014)*, 2014. 1, 2

[18] R. Gossweiler, M. Kamvar, and S. Baluja. What's Up CAPTCHA?: A CAPTCHA Based on Image Orientation. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 841–850, New York, NY, USA, 2009. ACM. 1, 2

[19] P. J. Kohler, A. Clarke, A. Yakovleva, Y. Liu, and A. M. Norcia. Representation of Maximally Regular Textures in Human Visual Cortex. *The Journal of Neuroscience*, 36(3):714–729, 2016. 1

[20] M. Korayem, A. A. Mohamed, D. Crandall, and R. V. Yampolskiy. Learning visual features for the avatar captcha recognition challenge. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, volume 2, pages 584–587. IEEE, 2012. 2

[21] M. Leyton. *Symmetry, Causality, Mind*. Cambridge, Massachusetts. MIT Press, 1992. A Bradford book. 1

[22] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. *CoRR*, abs/1405.0312, 2014. 3

[23] J. Liu, G. Slota, G. Zheng, Z. Wu, M. Park, S. Lee, I. Rauschert, and Y. Liu. Symmetry Detection from RealWorld Images Competition 2013: Summary and

Results. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 200–205. IEEE, 2013. 1, 2, 3

[24] Y. Liu, H. Hel-Or, C. S. Kaplan, and L. Van Gool. Computational Symmetry in Computer Vision and Computer Graphics. *Foundations and Trends® in Computer Graphics and Vision*, 5(1-2):1–195, 2010. 2

[25] G. Loy and J.-O. Eklundh. Detecting Symmetry and Symmetric Constellations of Features. In *Computer Vision – ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part II*, pages 508–521. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. 2, 3, 5, 6, 7, 8

[26] N. J. Mitra, H.-K. Chu, T.-Y. Lee, L. Wolf, H. Yeshurun, and D. Cohen-Or. Emerging Images. *ACM Transactions on Graphics*, 28(5):163:1–163:8, Dec. 2009. 2

[27] G. Mori and J. Malik. Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA. In *Computer Vision and Pattern Recognition, 2003. CVPR 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–134 – I–141. IEEE, 2003. 1

[28] D. Morrison, S. Marchand-Maillet, and É. Bruno. TagCaptcha: annotating images with CAPTCHAs. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 44–45. ACM, 2009. 1

[29] G. Moy, N. Jones, C. Harkless, and R. Potter. Distortion estimation techniques in solving visual CAPTCHAs. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–23–II–28. IEEE, 2004. 1

[30] Picatcha. Picatcha. http://picatcha.com/captcha/. 1, 2

[31] S. A. Ross, J. A. Halderman, and A. Finkelstein. Sketcha: A Captcha Based on Line Drawings of 3D Models. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 821–830, New York, NY, USA, 2010. ACM. 1, 2

[32] Y. Rui and Z. Liu. ARTiFACIAL: Automated Reverse Turing Test Using FACIAL Features. In *Proceedings of the Eleventh ACM International Conference on Multimedia*, MULTIMEDIA '03, pages 295–298, New York, NY, USA, 2003. ACM. 1, 2

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 2

[34] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: A Database and Web-Based Tool for Image Annotation. *International Journal of Computer Vision*, 77(1-3):157–173, May 2008. 2

[35] Y. Sasaki, W. Vanduffel, T. Knutsen, C. Tyler, and R. Tootell. Symmetry activates extrastriate visual cortex in human and nonhuman primates. *Proceedings of the National Academy of Sciences of the United States of America*, 102(8):3159–3163, 2005. 1

[36] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on*, pages 166–171, June 2011. 4, 5

[37] B. Stone. Breaking Google CAPTCHAs for some extra cash. *New York Times*, March 13, 2008. 1, 2

[38] Y. Sun, A. Singla, D. Fox, and A. Krause. Building Hierarchies of Concepts via Crowdsourcing. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 844–851. AAAI Press, 2015. 2

[39] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations*, 2014. 2

[40] C. Tyler (Ed.). *Human Symmetry Perception and its Computational Analysis*. VSP, Utrecht, The Netherlands, 1996. 1

[41] L. von Ahn, M. Blum, and J. Langford. Telling Humans and Computers Apart Automatically. *Communications of the ACM*, 47(2):56–60, Feb. 2004. 1

[42] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science*, 321(5895):1465–1468, 2008. 1, 2

[43] B. B. Zhu, J. Yan, Q. Li, C. Yang, J. Liu, N. Xu, M. Yi, and K. Cai. Attacks and Design of Image Recognition CAPTCHAs. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, pages 187–200, New York, NY, USA, 2010. ACM. 8