A Simple and Effective CAPTCHA by Exploiting the Orientation of Sub-images Cropped from Whole-size Photos

Woo-Keun Chung, Seung-Hyun Ji, Jong-Woo Kim and Hwan-Gue Cho
Graphics Application Laboratory,
Dept. of Computer Science and Engineering
Pusan National University, Republic of Korea
{wkchung, shji, jwkim73, hgcho}@pusan.ac.kr

ABSTRACT

Automated detection of image orientation has previously been studied as an important problem in intelligent image processing and computer vision. For this problem, numerous methods and tools have been developed by adopting approaches such as objects segmentation, color feature analysis and machine learning e.g., Support Vector Machines(SVMS). But conversely, the difficulty of image orientation can be used to examine the robustness of a CAPTCHA(Completely Automated Public Turing test to Tell Computers and Human Apart). The automated image orientation problem previously only had been studied and solved using typical photos which almost include important semantic cues such as people, bright sky, dark ground and vertical edges. In this paper we propose a simple prototype CAPTCHA, which exploits the hardness of orienting sub-images cropped from a whole digital photo. Our CAPTCHA takes 8 sub-images from base-photos and rotates them randomly. Then we present them to the user, who is required to find the correct orientations of the 8 sub-images. The true orientation is easily obtained since most current high-end digital cameras have an automatic mechanism to store its orientation in EXIF. Thus we can simply and easily obtain the image orientation without applying complicated computation. For our experiment, we have collected about 1850 base photos that provide more than 100,000 different sub-images. Experiment showed that the accuracy of our CAPTCHA with humans is about 95%. We think this sub-image orientation is hard to solve by an automated procedure since all previous machine learning procedures have only considered whole photos with enough semantic cues, rather than partial image segments. Another advantage of our system is that user interaction is simpler(there are four choices) and more intuitive than a common text-based system or the previous image orientation method with arbitrary rotation. Experiment showed that common users performed at most two rotations for each sub-image. The total time to complete orienting the 8 sub-image orientation was less than 15 seconds which is significantly shorter than that of previous image-based CAPTCHAs.

Keywords: CAPTCHA, Sub-image, Image Orientation, Image Classification, Machine Learning.

1 ISSUES IN CAPTCHA

1.1 Motivation

Recently, there have been numerous automated software bots and automated scripts that exploit public web services. The problems caused by automated SPAM generators are becoming serious for public web bulletins. So the user is commonly required to solve a Turing test problem, namely a "Completely Automated Public Turing test to tell Computers and Humans Apart(CAPTCHA), before they are allowed" to use web services. Also HIP(Human Interaction Proof) terms are widely accepted in this subject. Therefore the main goal of a HIP or CAPTCHA is to discourage script attacks by raising the computation and development cost of breaking a HIP or CAPTCHA to an unprofitable level[10]. One criteria of a CAPTCHA is that each puzzle should be easy for most people to solve, but difficult for automated bots to solve. We can easily construct a CAPTCHA that meets the criteria, by customizing each CAPTCHA problem manually, such as "Find a very funny picture among these photos". But there is another criteria for a CAPTCHA: each problem should be efficiently generated and evaluated by an automated procedure[5]. This is a contradictory issue in every CAPTCHA system. This paper addresses one simple procedure to make an automated image-based CAPTCHA from a small set of user photos which can be efficiently collected and refined. Also our CAPTCHA system is hard to break using the previous tools depending on training-based machine learning tools.

A brief introduction to our system is provided by the following figure. The puzzle is to guess the correct orientation of a cropped sub-image from a whole-size photo. We define a whole-size photo as an image file totally recorded in a single shot by a modern digital camera. In general we assume that every whole-size photo includes a few meaningful objects we can recognize easily and immediately.

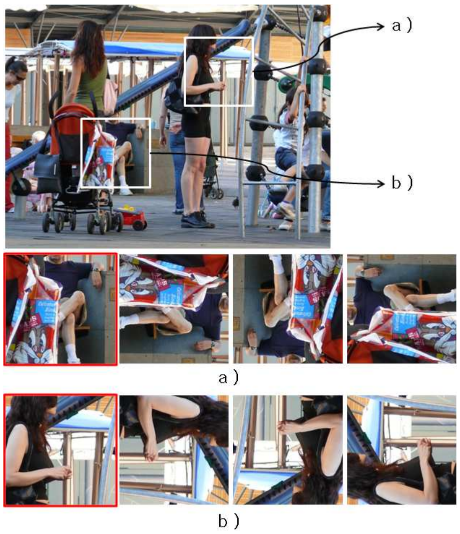

Figure 1: For a common given photo, we randomly crop sub-images. Next we show then to a human solver, who is required to find the correct orientation when the whole image is not shown.

The top image of Figure 1 shows a whole-size photo taken in a public children's park. We crop two sub-images (a) and (b) randomly, then we present them to the user, who is required to find the correct orientation. Each user can freely rotate the image 0, 90, 180 and 270 degrees in order to find the right position. Figure-1(a) and (b) shows four possible orientations for two square sub-images. As shown, a human observer can easily solve this problem (the leftmost one is correct) at a single glance. But previous automated tools for image orientation cannot deal with these partial images efficiently, since they have only considered the structures of whole photos, which include abundant semantically meaningful whole objects, rather than partial scenes. So the random chance to solve the two sub-images in Figure1 is 1/16.

1.2 Recent Advances on CAPTCHA

We simply survey the most recent work on CAPTCHAs for a comparison in terms of user-friendliness and system performance. Traditional CAPTCHAs asked users to identify a series of letters which were geometrically transformed to defeat a character recognition system[3, 10]. But recently, many smart character recognition softwares have succeeded in deciphering plain text-based CAPTCHAs. One way of breaking a noised text image is projection-based segmentation to clear artificial line noise, which have been demonstrated to break MSN and YAHOO[6]. And a careful shape analysis and machine learning approach also works well in deciphering text-based CAPTCHA [10, 12].

Since spammers are very eager to utilize academic algorithms to break CAPTCHAs, the defeating tools are easily exploited and distributed by spammers. Thus in order to defend against deciphering tools, text-based CAPTCHAs have increased the amount of noise and introduced different types of noise such as global warp into text images. But unfortunately this kind of hard-noised text puzzle makes it harder for humans as well as computers. Consequently a highly noised and wrapped text-based CAPTCHA leads to lower success rates(less than 60% accuracy) and frustrates common users[3, 5]. The drawbacks and pitfalls are well explained with typical examples in [2]. And the usability issues in character-based CAHPTCHAs were deeply discussed in one notable work[9].

In order to overcome the weak points of text-based CAPTCHAs, new Image-Based CAPTCHAs(IBCs) have been introduced recently. One great advantage of an image-based puzzle is that it is natural language-independent. There are many variants of IBCs[4]. These include selecting an appropriate label for an image and selecting an abnormal image for a set of subject images. One notable work has revealed the performance and disadvantages of image understanding CAPTCHAs [4]. Another interesting example of IBCw is identifying cats in 12 photos of both cats and dogs[7]. It was reported that humans solve this task 99.6% of the time in less than 30 seconds. Others showed that calculating a simple math problem can be used as a reasonable IBC[2].

Unfortunately most previous image-based CAPTCHAs have a basic problem in automatic generation, since they all require a priori knowledge such as the object name in the image or the true orientation of an image taken. For example if we ask for a good name(label) for a given image ,then we must specify the correct name(label) in advance or a priori, which is a burdensome work for a human generator. Also the name of the image may be dependent on the human generators. And if the size of the image database is not large or contains highly restricted objects, eg., cats and dogs only, then a well-known powerful machine learning model such as Support Vector Machines can break it after a moderate training and adaptation procedure[13], which insists that they can distinguish cats and dogs with more about 82.5% accuracy. Rather than a static image, Srikanth proposed a real-time image recognition CAPTCHA[16].

2 DETECTING IMAGE ORIENTATION

2.1 Image Orientation Problem

The image understanding problem has long been studied as a fundamental issue in computer vision. One typical application of image understanding is how to detect the human faces. Numerous computation models and tools and testing data on face detection have been well introduced in [21]. Or there is a specialized

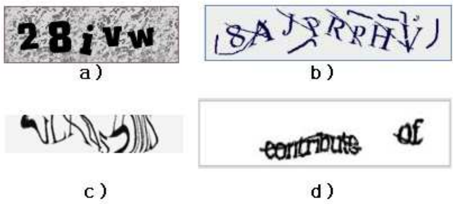

Figure 2: Typical types of text-based CAPTCHAs (a)text on noisy back ground. (b) noisy line segment added. (c) global warp applied. (d) warped after noise curve inserting.

cognition method for only one class of car objects[14]. As a subproblem of image understanding, automatic image orientation detection has been studied recently for content-based image organization and automated retrieval. But this type of image orientation detection is believed to be a very difficult problem, since state-of-the-art computer vision techniques still can not infer high-level abstraction of photo images[17]. For instance, it is very hard to recognize a dog partially hidden by a tree.

Generally speaking, a few well-established machine learning approaches such as MDL, LVQ, PCA, LDA(Linear Discriminant Analysis) and HDR(Hierarchical Discriminating Regression) are mainly applied for image orientation problem with a reliable set of training images. Others deeply studied the upper limit of orientation in terms of human psychological aspects [8]. Others have revealed the characteristics of image statistics in determining of image orientation[15], which was an essential factor leading to SVM-based automated image orienting system. Support Vector Machines are the most common learning tool for image orientation [11, 15, 19, 20].

It is worth to noting the psychophysical features in human image orientation perception[8]. For a randomly collected set of 1650 images five observers were asked to provide the correct orientation of the given image at various resolutions. The experimental results show that for typical images the orientation accuracy was close to 98% when all available semantic cues in high resolution photos were used. For coarser-resolution images, the accuracy was around 84%. Which means that upper limit of human orientation accuracy is about 84-95% depending on the image quality. Interestingly sky, ground and standing people are the most useful and reliable cues to find the correct orientation among other semantic cues including buildings, ceilings, color, grass, roads, texture and trees. This implies if the semantic cues are not observed or partially available in the image, then image orientation is a difficult problem.

The hardness of image orientation by computers can be a good basis for the usefulness of IBCs. Recently, detecting image orientation was used by a CAPTCHA system[5]. The IBC asks for an upright position of a circular form of image which was maximally inscribed in the image. This involved collecting a set of image s from a large image repository. Some images which have with an ambiguous upright orientation such as balls, guitars and general textures were excluded.

2.2 Machine Learning Approach for Image Orientation

Several machine learning approaches have long been applied to obtain the image orientation. The preprocessing step in this work involves transforming each photo into a vector with more than 1000 features[5]. For example, for an input feature vector, [5] used 1,965 means and 1,965 variances from 91 disjoint sub-regions. Next, the experimenters carefully collected a set of training photos in order to make a power classifier over unknown data. After applying numerous training steps, we finally obtained learned machine(program). In this step, it is very important to provide typical photos representing orientations that are very familiar to humans. Currently, Support Vector Machines are widely applied, and other methods combined SVMs with boosting models in order to obtain more correct and reliable results[18].

We describe the latent limitations in the machine learning approach for image orientation. First of all, input training images are commonly wide rectangular images with a height:width=ratio of 4:3. The preprocessing step for machine learning is where, photo images are usually divided. So we assume that we should divide a wide rectangle image into n by m grid-base sub-images. The top rows of an upper square sub-image are likely to be similar in general photos as was illustrated in the examples in Ref.[5, 17]. Thus if the training set includes this kind of plain images, it is easy to classify simply by observing the number of similar adjacent sub-grid images. Therefore, though the machine learning approach uses more than 2000 features of photosincluding RGB, YIQ, color intensity, vertical and horizontal edges, etc, we believe that the most crucial features of image vectors are smaller(30-50?) than we expected.

2.3 Problem of Sub-image Orientation

According to [8], the most important semantic cue(spatial region) is sky in the upper part of an image or grass or dark brown ground images in the lower part of a whole photo, as shown in Figure 1. So if the training images are not taken from a whole photo such as a partial region, then the most all kinds of machine learning approach fail, since a partial image hardly provides any kinds of semantic cues or environmental information helpful to obtaining the correct orientation. So if we restrict all the testing photos to exactly square photos, then the current machine learning approach does not show the good performance that was obtained in the previous work. We show another example of this issue in Figure 3. We know it is very easy to obtain the orientation of whole photos(top), but for their 4 sub-images(shown below)it is hard to obtain the correct orientation, since the sub-images do not include any whole faces. We have tested these four sub-images by applying well-known face detection tools and systems. Most face detection systems do not recognize the partial faces shown in Figure 3 (a)-(d).

Common users never take photos such as those shown in (a)-(d), and they were not used in any training set for machine learning tools where the sub-images do not provide any meaningful information to an automated machine. However, interestingly a human can recognize the right orientation of the 4 sub-images immediately, since a human has a very broad and abstract knowledge of partial face images. We have simply tested these 4 sub-images with 15 undergraduate students, who completely identified the orientation.

But most face recognition tools and systems did not recognize these four face sub-images, since the computing principles used in these face recognition trained machines were only developed using whole, even more, straight-view human faces. In fact most of the sub-images do not have the other semantic cue that the upper one is significantly brighter than the lower one in most all photos, which is a significant human perception cue to decide the orientation[18].

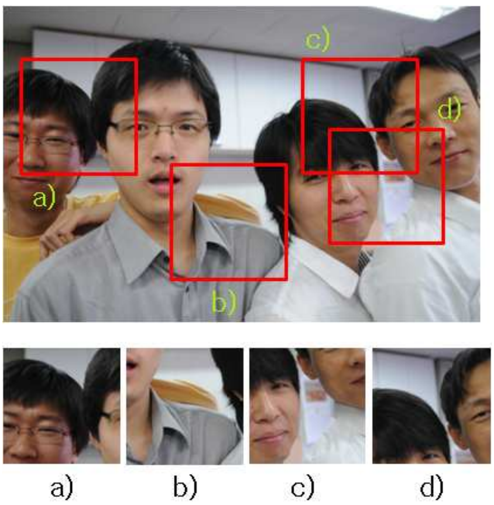

Figure 3: 4 different sub-images cropped from a photo. Though they do not have any single complete face, humans can easily obtain the correct orientation.

We provide another example to show that orienting a sub-image without whole images is hard for a computer, but easy for a human. See the other cropped sub-image which was taken from a travel photo in Figure 4. A Computer can easily orient this photo, since it has very typical semantic cues (one face and the a brighter upper region and darker lower region). But if only a sub-image of a partial face is used, then orientation is a very hard task because it does not contain enough semantic cues. But humans can do it without any hesitation.

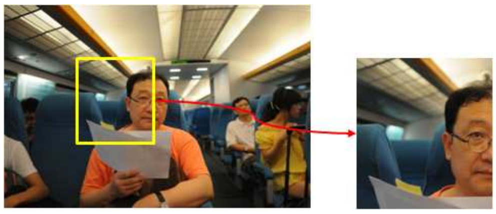

Figure 4: Computers can easily orient the left-hand whole photo but they cannot easily solve the right-hand partial subimages. Humans can do it immediately.

2.4 Image Orientation in EXIF of High-End Digital Camera

We believe the image orientation problem is easier due to the recent development of high-end digital cameras. Most of the high-end DSLR cameras from Canon, Nikon, Panasonic, Sony and SamSung have been equipped with internal automated orientation correction hardware. This information is stored in the EXIF field of each digital camera. EXIF denotes the Exchangeable file Image Format, which includes information about camera settings such as the timestamp, focal length, ISO, flash status etc. In the future, we will not have to apply a complicated machine learning system to obtain the orientation automatically, unless the photographer intentionally disturbs the image orientation. So in this paper, we assume that the true orientation can be easily obtained by retrieving the EXIF information of each digital photo. Thus in our CAPTCHA experiment, we only considered photos with orientation information recorded automatically by the mechanical sensor installed in digital cameras.

3 STRUCTURE OF OUR SYSTEM

3.1 Overview of Our System

In this section we explain the overview of our CAPTCHA system. Our system consists of four parts: the two photo databases DB1 and DB2, the random selector for image and cropping, the filtering module and the user interface. The n-plate is a problem including k sub-images cropped from our data base. If a human solver can provide at least k correct answers from n sub-images, then we say that a human can solve a k-plate CAPTCHA problem.

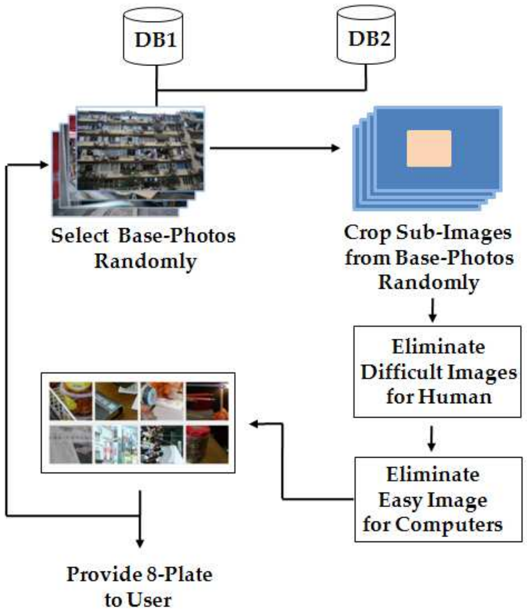

Figure 5: Overview of Our CAPTCHA and Data flow

DB1 was established by selecting photos which were previously taken during travel. And we intentionally took about 650 photos for use in our CHAPTCHAs, which were stored in DB2. We did not include any regions of sky, ground, paths and human faces. Also, we tried to avoid objects having too many vertical or horizontal edges such as straight buildings. If the inclusion of straight edges was inevitable, then we slightly rotated the viewing angle so as to avoid detection by edge segmentation tools automatically. But the rotation was limited to +10 to - 10 degrees to avoid frustrating a human solver's perceptions. One simple way to make DB2 photos is taking a scene with 5 to 10 small familiar stationeries, which are placed randomly on a desk within a distance of 1.0-2.0m from the camera lens. So after gathering around 600 photos we manually refined them to obtain the final photos. In the following, the base photos are the set of all photos stored in DB1 and DB2.

3.2 Eliminating Bad Sub-images

Though we carefully selected the base photos, bad subimages can be included in the final 8-plate problem, since a smaller region of a sub-image may contain a scene that is very hard for human solvers. When we crop subimages from the base photos, we automatically exclude those with one of following two conditions, since they do not provide any partial semantic cues to human.

- A photo with a large color segmentation region, which is called a monotone region in the following paper. Thus a monotone sub-image implies that it has no semantic objects. A plain texture without any object may belong to this category. Since a monotone region does not provide any orientation cues, it cannot be solved either by a computer or a human. Then the remaining problem is how to decide if a photo has a monotone region. We quantized the whole image into regions of 25 different natural colors by the color segmentation algorithm of [1]. They cleverly proposed and defined 25 representative colors including Black, Sea green, Light green, Aqua. Rise Yellow, Pink, Orange and White, for general photos favoring human perceptual sensing. So we can decompose a whole photo into at most 25 different colored regions. If the maximal area of one(from 25)-colored color segment is larger than 20% of the whole photo's area, then we discard the photo, since it is a hard problem for humans. The elimination criteria can be improved by experiment, but in this paper we do not address this problem further and we adopt a very simple rule. For instance, all photos including a larger sky region or ground(grass) region or water and wide walls are discarded(See 6).

(a) (b)

Figure 6: Two whole-size photos including larger monotone segments. If a red square is cropped, then it is not possible for a human to obtain the orientation due to the absence of meaningful objects(even partial ones)

- A photo including numerous human faces at a recognizable size and resolution. The most crucial semantic cue in a general photo is a human face, since the human eyes and month completely determine the image orientation in most cases(except when people are lying on the ground). Numerous face detection systems are available and can be obtained easily, such as OpenCV tools. We have utilized the tools released from *Applied Device* available on http://www.applieddevice.com/facedb/fs.php. If the size of the human face segments detected by the above tool is smaller than the area of the sub-image cropped by our system, then we disregard the photo, since the cropped sub-image completely contains the face image segment. Figure shows two photos belonging to this category. We exploited available codes to eliminate images with faces.

Figure 7: Examples of sub-images eliminated by our system. The left-hand two images were eliminated by the monotone region and the right-hand two were eliminated due to the faces it contained.

4 EXPERIMENT

4.1 Human Accuracy Analysis

In order to evaluate our system, we performed several user studies. We collected 350 base-photos which were taken in during overseas travel in Shanghai and Europe. Also we intentionally took more than 350 photos for use with this CAPTCHA. After refinement we finally initialize DB1 and DB2 with 1650 photos. Since we took the sub-images from the base photos, we can took more than 10,000 sub-images which were mutually different more than 70% of the time in terms of the spatial region.

We have evaluated our system with 10 human testers. We gave each tester, we gave 50 "8-plates", which consist of 8 different sub-images in on screen. We regard the answer as correct if a human finds more than 7 correct orientations from the given 8 images. The probability to break this 8-plate problem by random chance is given $\binom{8}{8} \cdot (1/4)^8 \cdot (3/4)^0 + \binom{8}{7} \cdot (1/4)^7 * (3/4)^1 = 25/(2^{16}) \approx 0.000381$

In order to find the optimal image size, we varied the crop size by 0.1*height of the base photos. In the following Table, *SIZE* denotes the size of the sub-images cropped from the whole-size photos. $SIZE = 01$ means the width and height of cropped sub-image is $0.3 * height$ of the whole-size photos. So if the width:height ratio of a photo is 3:2, then the area of a $SIZE = 0.3$ sub-image is $1/15 = 0.06$. So it is 6% of the area of a whole-size image, which is quite small. So if we can exclusively extract sub-images from a photo with $SIZE = 0.3$, we obtain get 15 totally different(disjoint) sub-images. If we allow some overlaps in the cropped sub-images, then we can get more than 30 sub-images, which is one of the advantages of our system.

We us show the main experiment plate of our system in Figure 8 which shows six different sub-image $SIZE = 0.1, 0.15, 0.2, 0.3, 0.35, 0.4$. The overall accuracy rate was more than 90% $SIZE > 0.3$. Since it is generally believed that a CAPTCHA with higher than 90% human accuracy is acceptable, we insist that $SIZE = 0.35$ provides quite good performance with an accuracy 97%, which so satisfactory in practice.

Detail statistics for the 10 testers are provided in Table 1. We did not apply the Partial Credit Algorithm introduced in [7], since it may weaken security as was pointed out in [7].

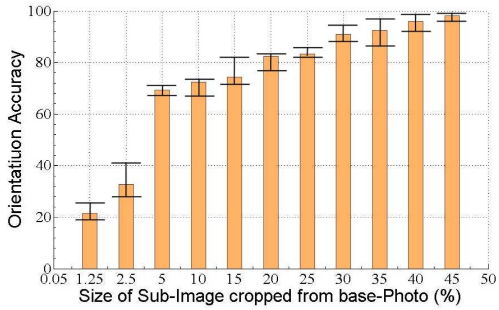

Figure 8: Experiment Graph

SIZE	0.1	0.15	0.20	0.25	0.30	0.35	0.40
tester 1	77.4	80.2	85.7	84.9	94.9	97.6	98.5
tester 2	73.4	76.4	80.7	82.1	88.8	98.6	98.1
tester 3	78.4	79.1	80.1	87.2	88.1	97.7	100.0
tester 4	75.1	74.5	83.9	86.3	91.5	98.4	98.2
tester 5	72.6	81.2	80.1	83.2	91.1	96.8	98.1
tester 6	75.9	78.2	79.4	86.9	93.5	97.5	100.0
tester 7	74.9	75.1	82.4	85.7	88.3	98.7	98.5
tester 8	75.5	74.3	81.2	81.2	94.8	96.4	100.0
tester 9	73.9	80.2	78.5	81.2	88.1	96.4	98.1
tester 10	74.6	77.3	79.2	87.1	93.1	97.2	98.4
Average	75.1	77.6	81.1	84.4	91.2	97.4	98.7

Table 1: 10 testers have tried to solve around 30-50 $8-plate$ problems at their work place for sub-images SIZE.

Figure 9: Example an 8-plate with SIZE=0.10

We explain the resolution of the base-photos as follows. The base-photo resolutions are various, ranging from a low resolution 1000*1000 photo to a high 4000*4000 one which was taken from a high-end DSLR. If the size of a sub-image is greater than 0.30, then the orientation accuracy does not depend on the image resolution. The accuracy is likely dependent on the relative size, rather than the absolute size(=the number of pixels) of the cropped sum-images. Figures 9 and Figure 10 show examples of 8-plate snapshots where SIZE=0.1 and 0.30. Figure-11 is the corresponding solution of the problem Figure=10.

Figure 10: Example an 8-plate with SIZE=0.10

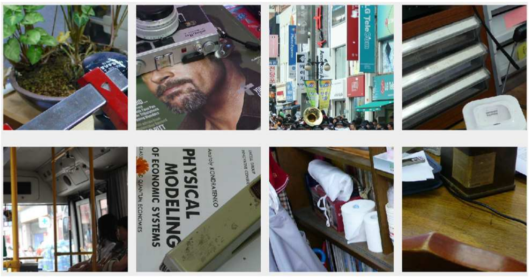

Figure 11: Corresponding correct orientations of 8-plate subimages in Figure 10

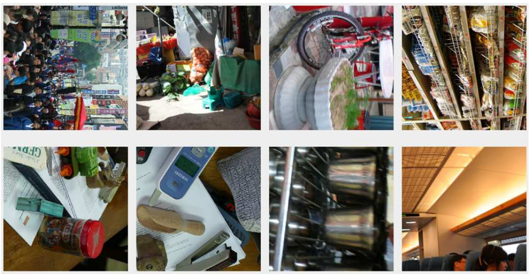

Figure 12: A plate of eight sub-images cropped with SIZE=0.40. From top-left, the photos show a crowded street during rush hour(more than 20 tiny faces), flee market, bicycle leaning against a street wall, shelf in a small market, five objects on a desk, objects(phone and stapler) on a desk, cups arranged on a restaurant shelf and a meeting room with a bright ceiling.

We show an example an 8-plate with SIZE=0.4 in the following Figure 12.

Intuitively we observe that an 8-plate of SIZE=0.4 is easier to solve by visual inspection than one with SIZE=0.30, since the larger sub-image may contain more semantic cues to help obtain the correct orientation.

4.2 Security and User Interaction

We discuss the usability of our system. There is no solid security analysis model for CAPTCHA, since it depends on very diverse human solvers.

The generally accepted figure in the previous work is that the system should admit automated bots with less

Image SIZE	Accuracy %	Clicks	Time(Sec.)
0.10	75.1	0.81	6.2
0.15	77.6	1.12	10.4
0.20	81.1	1.15	14.3
0.25	84.4	1.46	16.4
0.30	91.2	1.41	13.7
0.35	97.4	1.14	9.2
0.40	98.7	1.11	6.5

Table 2: Average success rate and average user clicks for each sub-image and average time for completing an 8-plate according to SIZE=0.1 - 0.5

than 1/10,000 probability within 30 seconds and only a small amount of human effort it is needed. Let the $(n,k)-plate$ problem denote that we regard the answer as right if a responder identified more than k sub-images from a total of n sub-images in the n-plate. The probability of random chance for the $(8,8)-plate$ is $1/4^8 = 1/65,536$, which is quite acceptable in practice. The probability of a random attack for the $(8,7)-plate$ is $(24+1)/4^8 \approx 3/10,000$. In practice we hope to use the $(9,8)-plate$, for which the random chance to break it is $(1+27)/4^9 \approx 1/10,000$.

We consider the user-friendliness of our system. We have checked the number of total clicks and elapsed time to clear an $(8,7)-plate$ in practice in Table 2. The average time for an 8-plate is around 13 seconds, which is faster than the previous text-based CAPTCHAs and the other IBCs, because the user only interacts with one mouse click, which rotates the image 90 degrees clockwise. It is interesting that solving the plate of with the small SIZE=0.1 and big SIZE=0.40 requires shorter time since a human does not need to rotate the sub-image due to the absence of any semantic cue objects or even partial images. Contrary to the small sub-image, about 8/4 sub-images in an 8-plate can be in a correct orientation if it is bigger(SIZE > 0.35). Thus it takes a shorter time to complete a plate.

Experiment showed that our 4-way discrete orientation CAPTCHA is quite efficient for a user compared with sliding rule-based image rotation[5]. Less then 2 clicks are needed to obtain the correct orientation, which implies that the interactions of our system are very human-friendly. So our system is very compatible to with mobile devices, e.g., touch-based cellular phones, where the user is allowed only a simple touch panel.

5 CONCLUSION

This paper introduced another kind of CAPTCHA by exploiting the orientation of cropped sub-images from whole-size base photos. We summarize the notable contributions of our idea.

- One fundamental disadvantage of the previous work on image-based CAPTCHAs is that they assume *a* *priori* knowledge such as image labels and semantic meanings. But in our model, the correct orientation is obtained entirely from the EXIF information of digital photos without any human intervention.

- It is hard for an automated procedure to obtain the correct orientation of a sub-image of a whole-size digital photo, because it has only partial semantic cues, which can be very difficult to learn in the machine learning model. But experiment showed that only 10% of a whole-size photo region is good enough for a human to easily and successfully provide the correct orientation with around 98% accuracy.

- The probability of random chance to break the $(9,8) - plate$ is about $1/10000$, which is low enough to be used in practice.

- Unlike the previous image-based CAPTCHA systems, we can generate more than $20 * N$ sub-images easily from N base-photos. Thus 1,000 base-photos provide more than 20,000 sub-images. Also, it is quite computationally expensive for an attacker to reconstruct a whole base-photo by gathering the sub-images used.

- The user interaction of our system is quite easy and simple, since the user chooses one from four possible orientations, taking one from four possible rotations, which can be done within 10 seconds for an 8-plate problem. So our model is quite effective, especially in a mobile environment where no keywords and control icon boxes are given.

We need to evaluate our system and rigorously compare it with the previous machine learning-based breaking tools using $k-$ plates. For practical applications, we are preparing more than 5,000 base-photos by filtering out bad images for humans and good images for computers. A web-based testing platform will be announced in the near future.

6 ACKNOWLEDGEMENT

This work was supported by the IT R&D program of MKE/MCST/IITA (2008-F-031-01, Development of Computational Photography Technologies for Image and Video Contents).

REFERENCES

[1] Prasad B. G., Biswas K. K., and Gupta S. K. Region-based image retrieval using integrated color, shape, and location index.

[2] H. Castro C. Javier and A. Ribagorda. Pitfalls in captcha design and implementation: The math captcha, a case study. pages 1–17, July 2009.

[3] Kumar Chellapilla, Kevin Larson, Patrice Simard, and Mary Czerwinski. Designing human friendly human interaction proofs (hips). In *Proc. of the SIGCHI conference on Human factors in computing systems*, pages 711–720, 2005.

[4] Monica Chew and J.D. Tygar. Image recognition captchas. In *Information Security Conference*, volume 3225, pages 268–279, 2004.

[5] Rich Gossweiler, Maryam Kamvar, and Shumeet Baluja. What's up captcha?: a captcha based on image orientation. In *Proc. of the 18th international conference on World wide web*, pages 841–850, 2009.

[6] Bell G. Huang S.Y., Lee Y.K. and Ou Z.H. A projection-based segmentation algorithm for breaking msn and yahoo captchas. In *Proc. of the International Conference of Signal and Image Engineering*, 2008.

[7] J. Howell J. Elson, JR. Douceur and J. Saul. Asirra: a captcha that exploits interest-aligned manual image categorization. In *Proc. of the 14th ACM conference on Computer and communications security*, pages 366–374, 2007.

[8] A. Singhal M. Boutell J. Luo, D. Crandall and R. Gray. Psychophysical study of image orientation perception. 16(5):429–457, 2003.

[9] Yan Jeff and El Ahmad Ahmad Salah. Usability of captchas or usability issues in captcha design. In *Proc. of the 4th Symposium on Usable Privacy and Security*, pages 44–52, 2008.

[10] P Simard K Chellapilla. Using machine learning to break visual human interaction proofs (hips). In *Advances in Neural Information Processing Systems 17*, 2004.

[11] Hong-Jiang Zhang Lei Zhang, Mingjing Li. Boosting image orientation detection with indoor vs. outdoor classification. In *Proc. of the Sixth IEEE Workshop on Applications of Computer Vision*, pages 95–99, 2002.

[12] G. Mori and J. Malik. Recognizing objects in adversarial clutter: breaking a visual captcha. In *Proc. of Computer Vision and Pattern Recognition.*, June 2003.

[13] Golle Philippe. Machine learning attacks against the asirra captcha. In *Proc. of the 15th ACM conference on Computer and communications security*, pages 535–542, 2008.

[14] B. Leung S. M. Bileschi and R. M. Rifkin. Towards component-based car detection. In *In ECCV Workshop on Statistical Learning and ComputerVision*, 2004.

[15] Lyu Siwei. Automatic image orientation determination with natural image statistics. In *Proc. of the 13th annual ACM international conference on Multimedia*, pages 491–494, 2005.

[16] V. Srikanth, C. Vishwanathan, Udit Asati, and N. Ch. Sriman Narayana Iyengar. Think-an image based captcha mechanism (testifying human based on intelligence and knowledge). In *Proc. of the International Conference on Advances in Computing, Communication and Control*, pages 421–424, 2009.

[17] A. Vailaya, H. Zhang, Changjiang Yang, Feng-I Liu, and A.K. Jain. Automatic image orientation detection. *IEEE Trans. on Image Processing*, 11(7):746–755, July 2002.

[18] Lei Wang, Xu Liu, Lirong Xia, Guangyou Xu, and Alfred Bruckstein. Image orientation detection with integrated human perception cues (or which way is up). In *Proc. of ICIP*, volume 2, pages 539–42, Sept. 2003.

[19] Yongmei Wang and Hongjiang Zhang. Content-based image orientation detection with support vector machines. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, 2001.

[20] Yongmei Michelle Wang and Hongjiang Zhang. Detecting image orientation based on low-level visual content. *Computer Vision and Image Understanding*, 93(3):328 – 346, 2004.

[21] Ming-Hsuan Yang, David J. Kriegman, and Narendra Ahuja. Detecting faces in images: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.