MI-CAPTCHA: Enhance the Security of CAPTCHA Using Mooney Images

Jingmeng Li, Lukang Fu, Surun Yang, Hui Wei*

Laboratory of Algorithms for Cognitive Models, Fudan University, Shanghai, China
{jmli21, lkfu23, sryang22}@m.fudan.edu.cn, weihui@fudan.edu.cn

Abstract

Completely automated public Turing test to tell humans apart (CAPTCHA) is an effective mechanism to protect websites and online applications from malicious bots programs. Image-based CAPTCHA is one of the most widely used schemes. However, deep learning techniques have significantly weakened the security of some image-based CAPTCHA schemes. Mooney images (MIs) are important research materials in the field of cognitive science. Compared to natural images, MI exhibits fewer visual cues, fragmented content, and greater ambiguity, leading to the perception of MI relying more on the iterative process between feedforward and feedback mechanisms. In this paper, we raise an intriguing problem: *can MIs be used to enhance the security of CAPTCHA?* Before this study, we first propose a novel framework HiMI that generates the high-quality MIs from natural images and also allows flexible adjustment of the perceived difficulty. Based on MI, we design two MI-CAPTCHA schemes related to object detection and instance segmentation tasks, respectively. We experimentally demonstrate that HiMI performs better than other baseline methods in terms of both image quality and application potential in two MI-CAPTCHA schemes. Additionally, we conduct experiments to explore the solving performance of humans and CAPTCHA solvers under different parameter settings of schemes, providing valuable reference for the practical application.

Introduction

Completely automated public Turing test to tell humans apart (CAPTCHA) is a defense mechanism against malicious scripts (Shi et al. 2020; Alqahtani and Alsulaiman 2020). According to the format of data, existing CAPTCHA mechanisms can be generally classified into three types: text, image, and audio. Text-based and image-based CAPTCHAs are the most widely used. Images are inherently more complex than text, making image-based CAPTCHA schemes more robust against automated attacks (Searles et al. 2023). In 2018, Google replaced the popular text-based reCAPTCHA v1 with the image-based reCAPTCHA v2 (**?**). However, deep learning techniques have achieved impressive successes in some computer vision tasks (e.g., object recognition), matching or even outperforming humans (He et al. 2015; Redmon et al. 2016). The security of some image-based CAPTCHAs have been significantly weakened.

CAPTCHAs essentially make use of the ability gaps between machines and humans to distinguish them (Hossen et al. 2020). As shown in Fig. 1(a), there are still significant gaps between deep neural networks and the human visual system in terms of robustness and generalization, and the closed-loop information processing mechanism is one of the reasons for this gap (Theeuwes 2010; Gilbert and Li 2013). Fig. 1(b) explains the process of closed-loop information processing for object recognition in the human visual system. It comprises two iterative components: feedforward/bottom-up and feedback/top-down (Theeuwes 2010; Cavanagh 1991; Roelfsema 2006). During the bottom-up process, the visual system integrates low-level visual signals through perceptual organization to obtain higher-level visual features and extract crucial cues (e.g., shape, texture). Our brain combines these cues with rich object knowledge to generate visual expectations. During the feedback process, the visual system actively adjusts perceptual organization by integrating missing information, filtering noise, and reducing redundancy to meet these expectations.

The Mooney image (MI) are a type of stylized image that consists of discrete speckles with irregular shapes and sizes, colored only in black and white (Mitra et al. 2009; Hegdé, Thompson, and Kersten 2007; Mooney 1957; Li and Wei 2024). Fig. 1(c) presents an example of MI. When some speckles are appropriately organized together, we can perceive a tiger. Compared to natural images, the MI contains fewer visual cues (e.g., color, luminance, texture), and its content is partial and discrete. These features enable it to play an important role in the fields of cognitive science (Van de Cruys et al. 2021; Hegdé and Kersten 2010; Li and Wei 2023) and web security (Mitra et al. 2009; Gao et al. 2015). The discrete speckles increase the difficulty of perceptual organization in the bottom-up process. In addition, MIs miss some visual content and cues, increasing its ambiguity. The emergence of perception on MI emphasizes the importance of iteration between feedforward and feedback in visual cognition. Here, we raise an intriguing question: *Can MIs be used to enhance the security of CAPTCHA?* Before answering it, we first consider how to generate high-

*corresponding author

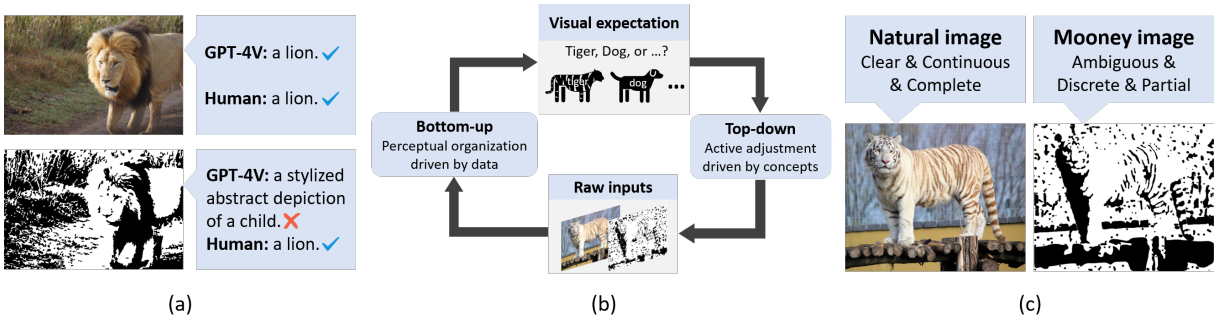

Figure 1: (a) Test on Chat GPT version 4 (GPT-4V) (Achiam et al. 2023), the current dominant large language model, using natural images and MIs. We input the image to GPT-4V along with the prompt "Tell me the name of the foreground object in the image". The test results show that humans can accurately perceive both natural image and MI, but the GPT-4V fails on the MI. (b) Explanation for the closed-loop information processing mechanism of human vision system. (c) Comparison between the natural image and the MI.

quality MIs from natural images. The poor performance and limitations of existing MI generation approaches are the first difficulty (Mitra et al. 2009; Yang, Kuo, and Chu 2016; Li et al. 2024, 2025). There are two challenges, color quantization and perceived difficulty adjustment. In comparison to the 24-bit full color space of natural images, MIs have only 1-bit. Existing color quantization methods perform poorly in extremely low-bit color space (Heckbert 1982; Gervautz and Purgathofer 1988; Hou, Zheng, and Gould 2020). In addition, the human visual system sometimes can not accurately perceive the foreground objects from MI, necessitating the flexible adjustment of perceptual difficulty.

In this paper, we first propose a novel framework HiMI that can use the natural image to generate high-quality MIs. HiMI consists of two steps: color quantization and perceived difficulty adjustment. In the first step, we present the end-to-end color quantization model CQNet to reduce the color space of the input image to 1-bit while preserving the abundant semantic content. Next, we consider two factors: object saliency and recognition cues, and set the corresponding parameters to control them, enabling HiMI to flexibly and effectively adjust the perceptual difficulty of the generated results. Inspired by reCAPTCHA v2, we design two MI-CAPTCHA schemes related to the object detection and instance segmentation tasks, respectively. In experiments, we first demonstrate that the MI generated by HiMI surpasses other baseline methods in terms of image quality and application potential in two MI-CAPTCHA schemes. We generate MI datasets corresponding to natural image datasets using HiMI, and then explore the impact of different parameter settings in two MI-CAPTCHA schemes on human users and CAPTCHA solvers. Experimental results provide valuable reference for future practical applications.

The main contributions are summarized as follows. (1) We present the color quantization model CQNet. It exhibits better quantization, especially in 1-bit color space. (2) We propose the novel MI generation framework HiMI. It can generate high-quality MIs from natural images and allows users to flexibly adjust the perceived difficulty of the generated results. (3) We design two MI-CAPTCHA schemes using the MIs generated by HiMI. Experimental results show that deep vision models perform significantly worse than human users in solving them, demonstrating their security. (4) We experimentally compare the performance of users and deep learning models in solving CAPTCHAs under different parameter settings, providing guidance on parameter configuration for the application.

Related Work

MI generation. J. Mitra et al. (Mitra et al. 2009) proposed a method for generating MIs based on 3D objects that quantifies the surface luminance and uses a threshold to control the difficulty level. However, this method suffers from two main issues: the quality of the generated results is significantly impacted by the viewing angle and illumination in the 3D simulation environment, and it relies on a specific 3D object library, which limits its practical application. Yang et al. (Yang, Kuo, and Chu 2016) introduced an approach to generate MIs from photographs. It utilizes superpixels as rendering primitives, and its generated results exhibit significant stylistic differences from MIs.

Universal style transform. Style transfer models are often used for image generation in specific styles and has two inputs: a style image and a content image, using the style pattern of the former to render the latter. These models can be classified into three categories: 1) models that can only render one style (Gatys, Ecker, and Bethge 2016); 2) models that can render multiple styles (Chen et al. 2017); 3) Universal style transfer (UST) models that can render arbitrary styles (Chandran et al. 2021; Li et al. 2017; Zhang et al. 2023b; Huang et al. 2023). UST models are particularly useful when training samples are scarce. In experiments, we will compare the generated results of our HiMI with UST models.

reCAPTCHA v2. reCAPTCHA v2 (Google-developers 2014) is a widely used and popular image-based scheme that distinguishes humans from web bots through challenges

based on visual tasks, such as image classification and object detection. In reCAPTCHA v2, there are two different but similar challenges. Both of them consist of a grid where the user should select the boxes according to the challenge instructions. However, the development of deep learning techniques has weaken its security. The current version adopts several some security enhancements over the earlier versions. It introduces anti-recognition techniques to render the challenge images unrecognizable to state-of-the-art deep vision models (Hossen et al. 2020). For example, it uses adversarial examples as a part of the anti-recognition mechanism. However, it also can be greatly reduced by fine-tuning the deep network model using challenge images that incorporate adversarial perturbations.

HiMI: High-quality Mooney Image

Overview

Fig. 2 depicts the pipeline of HiMI. Given a natural image, HiMI first uses CQNet to reduce its 24-bit color space to k-bit ($k \geq 2$, e.g., k=5) and 1-bit, respectively. Subsequently, it extracts eight-connected pixel clusters from the k-bit result based on color cues and constructs a graph structure describing the adjacency relationships of these pixel clusters (nodes). It then uses the 1-bit result to classify the nodes into two types: content (C) and non-content (NC), and combines them with the object segmentation results to divide the nodes to two sets: foreground (F) and background (B). The PDA module next uses the parameter α to adjust object recognition difficulty by merging and splitting of C-type and NC-type nodes in the F-set. The parameter β is used to adjust the number of C-type nodes in the B-set to control object saliency. Finally, C-type and NC-type nodes in PCAG are randomly assigned black (+) and white (-) to generate the corresponding a pair of binarized MIs.

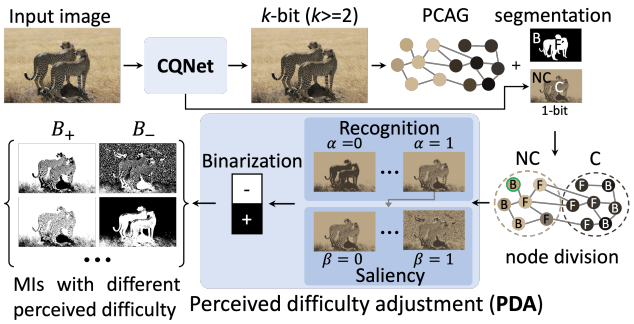

Figure 2: Pipeline of HiMI. It consists of two components: color quantization (CQNet) and perceived difficulty adjustment (PDA) module.

CQNet

Inspired by the work of Hou et al. (Hou, Zheng, and Gould 2020), we design the color quantization model CQNet as illustrated in Fig. 3. The first component is a UNeXt (Valanarasu and Patel 2022) auto-encoder that can extract abundant semantic information from the natural image I. The extracted information are fed into the linear convolution layer to generate the softmax probability map PM with s-channel, where s is the size of the color space. The corresponding set of values of pixel (x, y) in PM represents the contribution ratio of the pixel's color RGB values to the s colors. CQNet produce the color palette CP using PM. The RGB value of each quantized color is the weighted average of all pixels. The i-th quantized color in CP is defined as

$$CP_i = \frac{\sum_{(x,y)} I(x,y) \cdot PM(x,y,i)}{\sum_{(x,y)} PM(x,y,i)}. \quad (1)$$

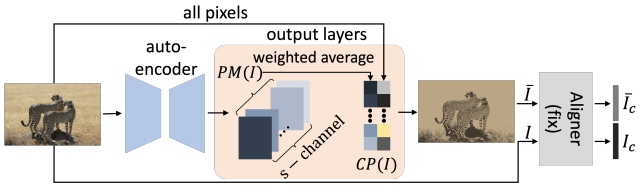

Figure 3: Architecture of CQNet. It consists of UNeXt auto-encoder, output layer, and aligner. The UNeXt auto-encoder extracts semantic information from natural images, and then the output layer designs the color palette and creates the quantizated result. Finally, the aligner enables CQNet to reduce the content difference between the quantizated image and the natural image.

Next, we use CP to map the pixels in I from the 24-bit full color space to the quantized color space (*e.g.*, 1-bit). The quantized image $\bar{I}$ is computed as

$$\bar{I} = \sum_{i=1}^{s} CP(i) \cdot PM(i), \quad (2)$$

where $PM(i)$ is used as the intensity of expression over entire quantized image $\bar{I}$.

Loss function. The human visual system perceives images by leveraging multiple cues, including color, texture, luminance, and edges. However, color quantification also leads to the loss of other visual cues, influencing the perception of the quantized image. To minimize the perceptual loss, we need to design a loss function that enables TTNet to preserve other visual cues when performing the color quantization. Here, the loss function is defined as

$$\mathcal{L} = \left\| \bar{I}_c - I_c \right\|_2, \quad (3)$$

where I_c and $\bar{I}_c$ are defined as

$$I_c = \frac{1}{N_l} \sum_{i=1}^{N_l} \Phi_i(I), \quad \bar{I}_c = \frac{1}{N_l} \sum_{i=1}^{N_l} \Phi_i(\bar{I}). \quad (4)$$

$\Phi_i(\cdot)$ denotes the features extracted from the i-th layer in the pre-trained VGG19, and N_l is the total number of network layers used for calculating content features. With the increasing depth of the network, there is an expansion in the scale of the receptive field, which consequently results in a more significant loss of local content details. Here, we set $N_l = 5$.

Perceived Difficulty Adjustment

The PDA module is designed to quantitatively adjust the the perceived difficulty of generated results, influencing the perception of humans on MIs. The key challenge is how to quantify the perceived difficulty, as the perceived difficulty is vague and subjective. Inspired by the psychological theories on the closed-loop visual perception mechanism as explained in Fig. 1(b), we split the perceptual process into two parts (sliency and recognition) and set two corresponding parameters to quantify the perceived difficulty.

Preparation for PDA. We use HiMI to obtain the quantization results of the original image in both k-bit and 1-bit color spaces, and employ an image segmentation model (e.g., GFM (Li et al. 2022)) to process the original image. The color quantization results consist of 2^k pixel clusters, where each cluster contains pixels with the same RGB values. We extract eight-connected pixel clusters from the k-bit results as the speckle primitives controlling the perceived difficulty, and then construct a pixel cluster adjacency graph (PCAG) to describe the neighboring relationships between these clusters. In PCAG, each node represents an eight-connected cluster. We categorize the nodes of PCAG into two types based on the 1-bit results. One type comprises nodes (C-type) with RGB values corresponding to content pixels, while the other type comprises non-content nodes (NC-type). The segmentation model divides the original image into foreground and background regions. Based on the segmentation result, we further divide the nodes belonging to the foreground (F) and background (B) regions into two sets, namely F-set and B-set.

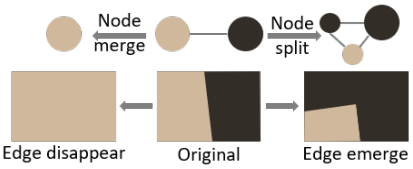

Figure 4: Explanation for recognition difficulty adjustment.

Recognition adjustment. Object recognition relies on various visual cues such as color, texture, shape, and luminance (Tanaka, Weiskopf, and Williams 2001; Wei and Li 2021). However, MI contains only black and white colors. Therefore, we perceive the content of MI relying more on cues other than color, with edges serving as the foundation for presenting these cues. Adjusting the number of edges in the object region can influence the perceived difficulty of MI. In PCAG, the nodes belonging to the F-set provide the content of foreground objects in the 1-bit result. As illustrated in Fig. 4, edge cues are represented by the connections between C-type and NC-type nodes. When we merge two nodes that have a connection, the edge cue represented by the connection disappear. If a node is split into two different types (C-type and NC-type), some edge cues disappear while additional noisy edges are introduced, interfering with visual perception. Here, we set the parameter α to adjust the difficulty of object recognition. We first count the number of C-type nodes (speckles) in the F-set, denoted as Num_C^F. If we select α proportion of cues, we randomly select $Num_C^F \cdot (1-\alpha)$ C-type nodes to merge into the NC-type nodes in the F-set.

Saliency adjustment. Before recognizing the target object, the visual system needs to perform figure-ground segregation (Wagemans et al. 2012). Research indicates that visual attention (including endogenous spatial attention and exogenous spatial attention) influences the occurrence of figure-ground segregation (Kimchi et al. 2016). Vecera et al. demonstrated that exogenous spatial attention influenced the role of bottom-up Gestalt cues in figure-ground segregation (Vecera and Farah 1994; Vecera 2000). The pattern (e.g., texture) differences between the foreground and background are closely related to exogenous spatial attention. After the process of the recognition adjustment, the speckle density of the foreground D_F is $\frac{Num_C^F \cdot (1-\alpha)}{Area(F)}$, where $Area(F)$ is the area of the foreground region. To control the saliency of the object, We set the parameter β to adjust the contrast of speckle density between the foreground and the background. The speckle density of the background D_B is $D_F \cdot \beta$, and the number of C-type node in the B-set is Num_C^B. Theoretically, the number of speckles contained in the background should be $D_B \cdot Area(B)$. If $D_B \cdot Area(B) > Num_C^B$, then we need to add some speckles to the background. We do not generate speckles but select some ones from the speckles contained in the foreground object. We randomly select $D_B \cdot Area(B) - Num_C^B$ C-type nodes from the F-set and place them within empty areas of the background.

Result binarization. We binarize the result of PDA moudle to generate the final MI. Rubin Vase (Fisher 1968), a well-known ambiguous image, can be perceived either as the black silhouette of two faces looking at each other or as a white vase, but not both simultaneously. Inspired by this, we assign black and white colors to C-type and NC-type nodes, generating a pair of complementary MIs to enhance ambiguity. If C-type nodes are black and NC-type nodes are white, it is denoted as B_+, otherwise B_-.

MI-CAPTCHA

CAPTCHA schemes. Inspired by reCAPTCHA v2, we design two MI-CAPTCHA schemes based on object detection and instance segmentation tasks, as illustrated in Fig. 5. We use HiMI to generate the corresponding MI datasets from natural image datasets. In Scheme 1, four MIs are randomly selected from the MI dataset, including M MIs containing objects of the target category. The challenge instruction is "*Please select all images containing a specific type of object*". In Scheme 2, a single MI is randomly selected from the MI dataset. We place four red dots in this MI based on the ground truth of instance segmentation for the original image, with one red dot must locate within the foreground object region. The challenge instruction is "*Please select all red dots located within the object region*".

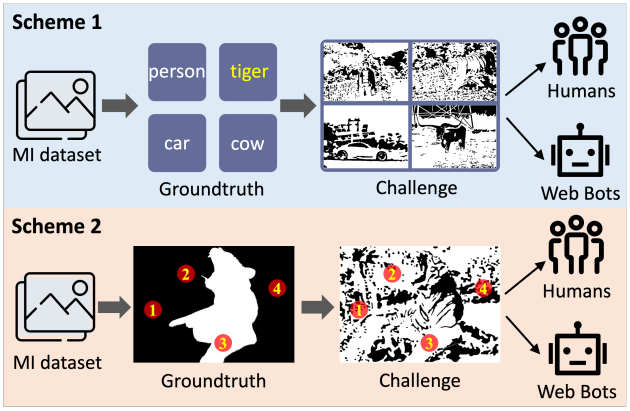

Figure 5: Illustration of two MI-CAPTCHA schemes. The challenge instruction of Scheme 1: please select all images containing the tiger. The challenge instruction of Scheme 2: please select the red dots located in the area of tiger.

Experiments

Experimental Settings

Settings. CQNet is implemented in Python with PyTorch and is trained on a Linux server with NVIDIA 3090 GPU. The rests are implemented in MATLAB 2021a on a machine with Intel Core i5-3470 CPU @ 3.20 GHz and 16 GB of main memory. We adjust the perceived difficulty of generated results by parameters α, β and Binarization B including B_+ and B_-. Here, we use $\text{HiMI}_{(\alpha,\beta,B)}$ to denote HiMI with different settings. In Table 1, we take some examples to explain the meanings of the notations related to these three parameters.

Parameter	Notation	Meaning
α, β	1	set parameter to 1
	$\{0, 0.5, 1\}$	set parameter to 0, 0.5, 1 in order
	$[0, 1]$	randomly set it to a value within $[0, 1]$
B	$+$	only use the template B_+
	$-$	only use the template B_-
	$+/-$	randomly use a template B_+ or B_-

Table 1: Meanings of notations realted to parameters α, β and B in the following experiments.

Datasets. We use the following two publicly available datasets to conduct experiments. Animal 2K (Li et al. 2022) is used in image matting and consists of 2,000 high-resolution images, with 1,800 images in the training set and 200 images in the test set. MS COCO 2017 (Lin et al. 2014) is a classical dataset for multiple computer vision tasks such as object detection and instance segmentation. It contains 80 classes of objects with a total of 123,287 images (118,287 samples in training set and 5,000 samples in the val set).

Evaluation strategy. Currently, there is no common evaluation strategy for MIs. As described in the section of Introduction, MI is a stylized image with significant application value in cognitive psychology and CAPTCHAs. The application potential of MI generation methods can reflect their effectiveness. MI applications involve human perception and defense against computer vision models. A reasonable strategy for measuring application potential includes two aspects: image quality and defense capability. For evaluating image quality, we use methods commonly used in style transfer research: quantitative evaluation based on deep encoders and qualitative evaluation based on user studies. The defense capability of MI can be measured by the performance differences of various deep networks on some vision tasks between natural images and MIs.

Comparison between HiMI and Baseline Methods

Baseline methods. We compare HiMI with the traditional MI generation methods PhotoMI (Yang, Kuo, and Chu 2016) and four state-of-the-art UST models: AdaConv (Chandran et al. 2021), WCT (Li et al. 2017), QuantArt (Huang et al. 2023), InST (Zhang et al. 2023b). AdaConv transfers the global statistics and spatial structure of the style image to the content image. WCT model uses a whitening and coloring transformation to align the second-order statistics of content and style features. QuantArt aims to generate the stylized image with high visual-fidelity by pushing the latent representation of the generated artwork toward the centroids of the real artwork distribution with vector quantization. InST is a diffusion-based method. Its key idea is to learn the artistic style directly from a single painting and then guide the synthesis.

Generation results. There are significant color differences between the results. Specifically, the outputs of PhotoMI and HiMI are binary images, while those of the UST models are colorful or grayscale. For the fair comparison, we first use CQNet to reduce the color space of the UST model results to 1-bit. Here, '+bw' indicates the binarization operation. Fig. 6 presents several examples of the generated results from HiMI and five baseline methods. Considering the differences in the perception processes of humans and deep neural networks, we conduct further experiments to quantitatively evaluate the image quality of these results from the perspectives of deep neural network and human perception, respectively.

Quantitative evaluation of MI quality. Following studies (Huang and Belongie 2017; Li et al. 2017), we use two metrics including the content loss and the style loss to quantitatively evaluate the generated results. The content loss denotes the difference between the generated result and the content image and the style loss is the difference between the generated result and the style image. The pre-trained VGG19 is used to extract features from the generated result at 256×256 resolution, and then we calculate the content loss and style loss. Table 2 lists the quantitative comparison results. HiMI exhibits significantly lower content loss and style loss than other methods. We find that although the generated results of $\text{HiMI}_{(1,0,+)}$ and $\text{HiMI}_{(1,0,-)}$ have the same edge information, $\text{HiMI}_{(1,0,-)}$ is slightly lower in content loss and style loss than $\text{HiMI}_{(1,0,+)}$. This is attributed to the

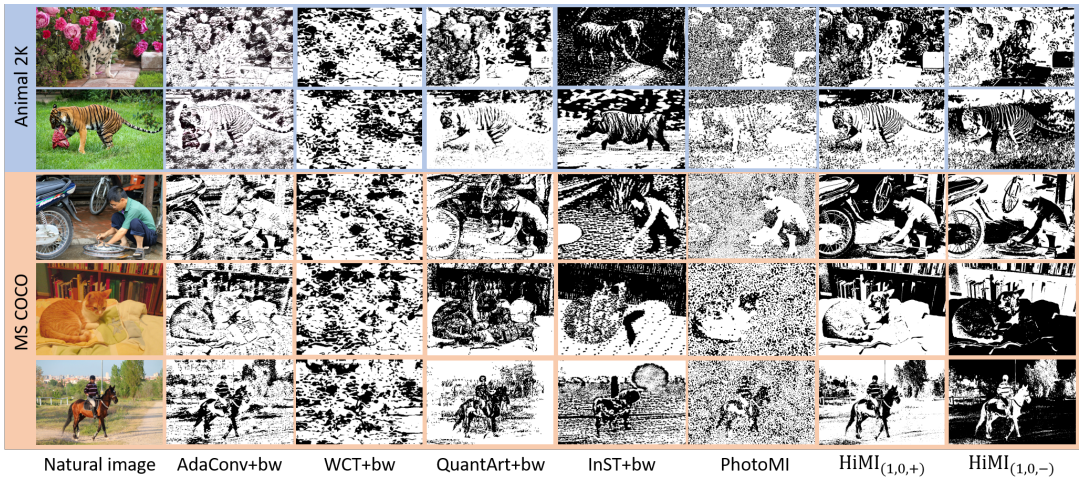

Figure 6: Qualitative comparison of generated MIs between HiMI and other five baseline methods.

Methods	AdaConv	WCT	QuantArt	InST	PhotoMI	$\text{HiMI}_{(1,0,+)}$	$\text{HiMI}_{(1,0,-)}$
Content loss	1.16	1.34	1.39	1.46	1.36	**1.02**	1.05
Style loss	0.28	0.74	0.68	0.31	0.78	**0.19**	0.24

Table 2: Quantitative comparisons between the binarized results of four UST models and the results of PhotoMI and HiMI.

exchange of foreground and background colors, which increases the loss of luminance features in MIs.

User study on MI quality. In the study by Searles et al. on evaluation of image-based CAPTCHA schemes (Searles et al. 2023), solving time and accuracy are important metrics. Therefore, we conduct user studies to collect data on the recognition time and accuracy of MI generated by different methods, and then evaluate the application potential for two MI-CAPTCHA schemes. We recruit 120 participants without visual cognitive impairments and divide them into six groups. We randomly select 80 natural images from the Animal 2K and 100 natural images from the COCO dataset, and also divide them into six groups, each containing 30 images. For each image, we generate the corresponding MIs using $\text{HiMI}_{(1,0,+)}$ and five baseline methods, followed by the operation '+bw'. We conduct a total of six rounds of experiments. In each round, we present the six methods' MIs to the six groups of participants, respectively. Participants are required to recognize the foreground objects of MIs, and both the time cost and the recognition result are recorded. Table 3 lists the statistical results from the user study, where MIs generated by HiMI exhibits the shortest perception time and highest recognition accuracy. These experimental results demonstrate that the MI generated by HiMI exhibits the best image quality from both deep network and human perspectives.

Defense capability. Scheme 1 of MI-CAPTCHA is based on the object detection task. We first measure the defense capability of MI against the deep object detection models. YOLO (Redmon et al. 2016), the classical one-stage model, performs impressive results on the task of object detection, and it is also used in the Google reCAPTCHA v2 solver. In addition, DINO (Zhang et al. 2023a) is a strong end-to-end object detector and improves over previous Transformer-based models in performance and efficiency. An important design principle for image-based CAPTCHA is to reduce machine performance while minimizing the impact on human visual perception. According to the results in Table 3, some baseline methods perform poorly, inconsistent with this principle. Therefore, we compare our HiMI with the best-performing traditional method, PhotoMI, and the top UST model, AdaConv. We first generate the corresponding MI datasets from two natural image (NI) datasets. For each natural image in the training set, we use $\text{HiMI}_{(1,\{0,0.5,1\},+/-)}$ to generate its three MIs with different perceived difficulty. For each image in the validation set, we use $\text{HiMI}_{([0.5,1],[0,1],+/-)}$ to generate its MI. We use the available pre-trained YOLO v8 and DINO, and then finetune them on both MI datasets. For training stage, we use a batch size of 16 and train the model for 300 epochs with an initial learning rate of 0.01. The image in COCO contains one or more objects, and we select the one with the largest area as the target object. In the final detection result, if there exists a bounding box labeled as target object class, then the task model successfully solves the sample. Table 4 lists the accuracy of two object detection models. The experimental results demonstrate that the MIs generated by HiMI can enable Scheme 1 to better defend against deep object detection models. Additionally, we use the YOLO and the Segment Anything Model (SAM) (Kirillov et al. 2023) as task models to compare the application potential of MIs generated by

Methods	AdaConv	WCT	QuantArt	InST	PhotoMI	HiMI$_{(1,0,+/-)}$
Time (s)	35	71	37	32	29	**13**
Accuracy	0.359	0.023	0.338	0.104	0.325	**0.937**

Table 3: Statistical results of the user study.

COCO				
Models	NI	MI		
		AdaConv	PhotoMI	HiMI
YOLO	0.673	0.412	0.368	0.258
DINO	0.669	0.404	0.347	0.249
Animal 2K				
Models	NI	MI		
		AdaConv	PhotoMI	HiMI
YOLO	0.748	0.461	0.412	0.324
DINO	0.736	0.452	0.407	0.311

Table 4: Comparison of defense capability against deep object detection models.

AdaConv, PhotoMI, and HiMI for Scheme 2. For each MI, we compute the IoU between the mask generated by the task models and ground truth. Table 5 lists the mean IoU of two task models. Experimental results show that HiMI achieves the greatest reduction in the performance of two instance segmentation models.

Models		YOLO	SAM
NI		0.974	0.986
MI	AdaConv	0.293	0.152
	PhotoMI	0.278	0.139
	HiMI	0.216	0.104

Table 5: Comparison of defense capability against instance segmentatioin models.

Configuration of MI-CAPTCHA Schemes

Parameter analysis. In Scheme 1, we randomly select 4 MIs from the MI dataset for each instance, with M of them containing the target class object. How do different parameter settings of M affect the performance of human users and CAPTCHA solvers? We set M to 1, 2, 3, and 4 sequentially, generating 100 CAPTCHA challenge samples for each parameter setting. For each sample, 120 participants are required to select all MIs containing the target class object from the four provided MIs, while the CAPTCHA solver (YOLO) performs object detection on the four MIs. We record the time and success rate (SR) of the participants in completing all samples, as well as the SR of YOLO. The experimental results in Table 6 show that as M increases, both the participants' and YOLO's SR decrease. Additionally, because we do not inform participants of the value of M, they need to observe all MIs before making a decision, resulting in little change in time. When M is set to 2, participants have the largest performance gap with YOLO while maintaining a high success rate. Therefore, $M = 2$ is the optimal configuration for Scheme 1.

M	Humans (SR / Time (s))	YOLO (SR)
1	0.895 / 49.2	0.313
2	0.763 / 51.5	0.105
3	0.631 / 53.3	0.041
4	0.379 / 53.9	0

Table 6: Effect of different parameter M settings on solving Scheme 1 of user and solver (YOLO).

Ablation Study

In this section, we demonstrate the effectiveness of the two modules: CQNet and PDA.

CQNet. To demonstrate the effectiveness of CQNet, we replaced it with three other representative color quantization methods (*i.e.*, OcTree (Gervautz and Purgathofer 1988), MedianCut (Heckbert 1982), and ColorCNN (Hou, Zheng, and Gould 2020)). The qualitative comparison demonstrates that the MI generated using CQNet is closest to the natural images from the visual perception. Additionally, the results of quantitative evaluation using content loss shows that CQNet enables HiMI to achieve lower content loss.

PDA moudle. The PDA module is designed to enable HiMI to generate MIs with different perceived difficulties. Here, we use HiMI$_{(\{1,0.7,0.5\},\{0,0.2,0.4,0.6,0.8,1\},+)}$ to generate MIs. The quantitative results demonstrate that the differences between these MIs and the natural images gradually increase as the parameters are adjusted, and substantial differences exist between these MIs.

Conclusion

In this paper, we investigate whether Mooney images (MIs) can enhance CAPTCHA security. We propose HiMI, a novel framework for generating high-quality MIs from natural images with adjustable difficulty. We then design two MI-CAPTCHA schemes based on object detection and instance segmentation tasks. Experimental results show that HiMI-generated MIs outperform existing methods in both image quality and CAPTCHA application potential. Additionally, we conduct parameter analysis to assess the impact of different settings on the performance of human users and CAPTCHA solvers, offering practical insights for implementation.

Acknowledgements

This work was supported by the NSFC Project (Grant number: 61771146).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alqahtani, F. H.; and Alsulaiman, F. A. 2020. Is image-based CAPTCHA secure against attacks based on machine learning? An experimental study. *Computers & Security*, 88: 101635.

Cavanagh, P. 1991. What's up in top-down processing. *Representations of vision: Trends and tacit assumptions in vision research*, 295–304.

Chandran, P.; Zoss, G.; Gotardo, P.; Gross, M.; and Bradley, D. 2021. Adaptive convolutions for structure-aware style transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7972–7981.

Chen, D.; Yuan, L.; Liao, J.; Yu, N.; and Hua, G. 2017. Stylebank: An explicit representation for neural image style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1897–1906.

Fisher, G. H. 1968. Ambiguity of form: Old and new. *Perception & Psychophysics*, 4: 189–192.

Gao, S.; Mohamed, M.; Saxena, N.; and Zhang, C. 2015. Emerging image game CAPTCHAs for resisting automated and human-solver relay attacks. In *Proceedings of the 31st Annual Computer Security Applications Conference*, 11–20.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2414–2423.

Gervautz, M.; and Purgathofer, W. 1988. A simple method for color quantization: Octree quantization. In *New Trends in Computer Graphics: Proceedings of CG International'88*, 219–231. Springer.

Gilbert, C. D.; and Li, W. 2013. Top-down influences on visual processing. *Nature Reviews Neuroscience*, 14(5): 350–363.

Google-developers. 2014. Choosing the type of recaptcha. https://developers.google.com/recaptcha/intro.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

Heckbert, P. 1982. Color image quantization for frame buffer display. *ACM Siggraph Computer Graphics*, 16(3): 297–307.

Hegdé, J.; and Kersten, D. 2010. A link between visual disambiguation and visual memory. *Journal of Neuroscience*, 30(45): 15124–15133.

Hegdé, J.; Thompson, S.; and Kersten, D. 2007. Identifying faces in two-tone ('Mooney') images: A psychophysical and fMRI study. *Journal of Vision*, 7(9): 624–624.

Hossen, M. I.; Tu, Y.; Rabby, M. F.; Islam, M. N.; Cao, H.; and Hei, X. 2020. An Object Detection based Solver for {Google's} Image {reCAPTCHA} v2. In *23rd international symposium on research in attacks, intrusions and defenses (RAID 2020)*, 269–284.

Hou, Y.; Zheng, L.; and Gould, S. 2020. Learning to structure an image with few colors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10116–10125.

Huang, S.; An, J.; Wei, D.; Luo, J.; and Pfister, H. 2023. Quantart: Quantizing image style transfer towards high visual fidelity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5947–5956.

Huang, X.; and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, 1501–1510.

Kimchi, R.; Yeshurun, Y.; Spehar, B.; and Pirkner, Y. 2016. Perceptual organization, visual attention, and objecthood. *Vision Research*, 126: 34–51.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. *arXiv preprint arXiv:2304.02643*.

Li, J.; Fu, L.; Yang, S.; and Wei, H. 2025. HiEI: A Universal Framework for Generating High-quality Emerging Images from Natural Images. In *European Conference on Computer Vision*, 129–145. Springer.

Li, J.; and Wei, H. 2023. Important Clues that Facilitate Visual Emergence: Three Psychological Experiments. In *Annual Meeting of the Cognitive Science Society (CogSci)*, volume 45.

Li, J.; and Wei, H. 2024. Make Use of Mooney Images to Distinguish between Machines and Humans. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46.

Li, J.; Wei, H.; Yang, S.; and Fu, L. 2024. Emerging image generation with flexible control of perceived difficulty. *Computer Vision and Image Understanding*, 240: 103919.

Li, J.; Zhang, J.; Maybank, S. J.; and Tao, D. 2022. Bridging composite and real: towards end-to-end deep image matting. *International Journal of Computer Vision*, 130(2): 246–266.

Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; and Yang, M.-H. 2017. Universal style transfer via feature transforms. *Advances in neural information processing systems*, 30.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Mitra, N. J.; Chu, H.-K.; Lee, T.-Y.; Wolf, L.; Yeshurun, H.; and Cohen-Or, D. 2009. Emerging images. *ACM transactions on graphics (TOG)*, 28(5): 1–8.

Mooney, C. M. 1957. Age in the development of closure ability in children. *Canadian Journal of Psychology/Revue canadienne de psychologie*, 11(4): 219.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.

Roelfsema, P. R. 2006. Cortical algorithms for perceptual grouping. *Annu. Rev. Neurosci.*, 29: 203–227.

Searles, A.; Nakatsuka, Y.; Ozturk, E.; Paverd, A.; Tsudik, G.; and Enkoji, A. 2023. An Empirical Study & Evaluation of Modern CAPTCHAs. In *32nd USENIX Security Symposium (USENIX Security 23)*, 3081–3097.

Shi, C.; Ji, S.; Liu, Q.; Liu, C.; Chen, Y.; He, Y.; Liu, Z.; Beyah, R.; and Wang, T. 2020. Text captcha is dead? a large scale deployment and empirical study. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, 1391–1406.

Tanaka, J.; Weiskopf, D.; and Williams, P. 2001. The role of color in high-level vision. *Trends in cognitive sciences*, 5(5): 211–215.

Theeuwes, J. 2010. Top–down and bottom–up control of visual selection. *Acta psychologica*, 135(2): 77–99.

Valanarasu, J. M. J.; and Patel, V. M. 2022. Unext: Mlp-based rapid medical image segmentation network. In *International conference on medical image computing and computer-assisted intervention*, 23–33. Springer.

Van de Cruys, S.; Damiano, C.; Boddez, Y.; Król, M.; Goetschalckx, L.; and Wagemans, J. 2021. Visual affects: Linking curiosity, Aha-Erlebnis, and memory through information gain. *Cognition*, 212: 104698.

Vecera, S. P. 2000. Toward a biased competition account of object-based segregation and attention. *Brain and Mind*, 1: 353–384.

Vecera, S. P.; and Farah, M. J. 1994. Does visual attention select objects or locations? *Journal of Experimental Psychology: General*, 123(2): 146.

Wagemans, J.; Elder, J. H.; Kubovy, M.; Palmer, S. E.; Peterson, M. A.; Singh, M.; and Von der Heydt, R. 2012. A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure–ground organization. *Psychological bulletin*, 138(6): 1172.

Wei, H.; and Li, J. 2021. Computational Model for Global Contour Precedence Based on Primary Visual Cortex Mechanisms. *ACM Transactions on Applied Perception (TAP)*, 18(3): 1–21.

Yang, C.-H.; Kuo, Y.-M.; and Chu, H.-K. 2016. Synthesizing emerging images from photographs. In *Proceedings of the 24th ACM international conference on Multimedia*, 660–664.

Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L.; and Shum, H.-Y. 2023a. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. In *The Eleventh International Conference on Learning Representations*.

Zhang, Y.; Huang, N.; Tang, F.; Huang, H.; Ma, C.; Dong, W.; and Xu, C. 2023b. Inversion-based style transfer with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10146–10156.