HiEI: A Universal Framework for Generating High-quality Emerging Images from Natural Images

Jingmeng Li[1], Lukang Fu[1], Surun Yang[1], and Hui Wei*[1,2]

[1] Laboratory of Algorithms for Cognitive Models, Fudan University, Shanghai, China
{jmli21,lkfu23,sryang22}@m.fudan.edu.cn, and weihui@fudan.edu.cn
[2] Innovation Center of Calligraphy and Painting Creation Technology, MCT, China

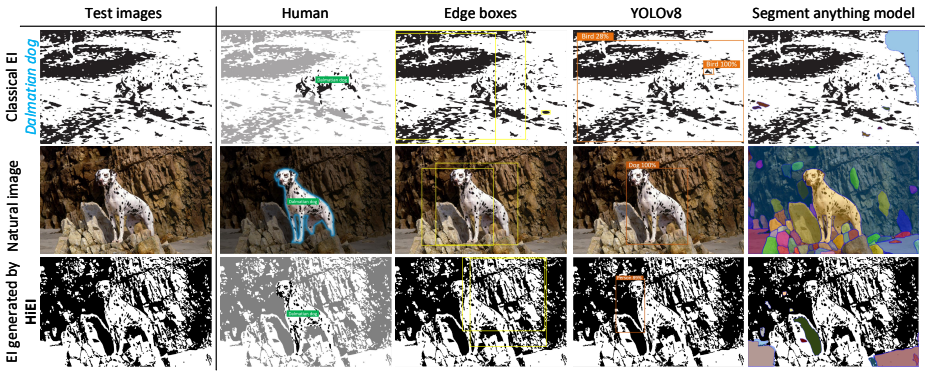

Fig. 1: Quick look at our study. We propose a high-quality EI generation framework HiEI and experimentally demonstrate that EIs generated by HiEI can effectively defend against attacks from deep vision models [15, 23, 24, 31, 35].

Abstract. Emerging images (EIs) are a type of stylized image that consists of discrete speckles with irregular shapes and sizes, colored only in black and white. EIs have significant applications that can contribute to the study of perceptual organization in cognitive psychology and serve as a CAPTCHA mechanism. However, generating high-quality EIs from natural images faces the following challenges: **1)** color quantization–how to minimize perceptual loss when reducing the color space of a natural image to 1-bit; **2)** perceived difficulty adjustment–how to adjust the perceived difficulty for object detection and recognition. This paper proposes a universal framework HiEI to generate high-quality EIs from natural images, which contains three modules: the human-centered color quantification module (TTNet), the perceived difficulty control (PDC) module, and the template vectorization (TV) module. TTNet and PDC modules are specifically designed to address the aforementioned challenges. Experimental results show that compared to the existing EI generation methods, HiEI can generate EIs with superior content and style quality while offering more flexibility in controlling perceived difficulty.

* Corresponding author.

In particular, we experimently demonstrate that EIs generated by HiEI can effectively defend against attacks from deep network-based visual models, confirming their viability as a CAPTCHA mechanism.

Keywords: Emerging image · CAPTCHA · Color quantization

1 Introduction

Emerging images (EIs), also known as Mooney images in the field of cognitive science [3,17,27], are a type of stylized image composed of discrete speckles with irregular shapes and sizes, featuring only black and white colors [30]. When certain speckles are appropriately organized together, humans can perceive meaning objects in EIs. For example, we can perceive a dalmatian dog from the classical EI (*Dalmatian dog*) depicted in the first row of Fig. 1.

The distinctive form endows them with significant application value. Due to the limited and fragmented visual cues in EIs, the human visual system relies on an iterative process of bottom-up perceptual organization and top-down active adjustment to perceive contents in EIs [5, 32, 38]. Therefore, EIs are often used to assist researches related to perceptual organization and closed-loop information processing mechanisms in cognitive psychology [9,17,25]. In addition, considering the differences in capabilities between visual models and the human visual system, EIs also serve as a CAPTCHA mechanism in the field of web security [2,11,26,30,34,37,41]. However, these related studies only demonstrated the effectiveness of EI-CAPTCHA defense against traditional vision models. Therefore, it is a valuable topic to explore whether EIs are effective in defending against vision models based on deep networks.

Generating high-quality EIs from natural images is crucial for above-mentioned research. Mitra et al. [30] proposed a scheme to generate EIs based on three-dimensional object templates in a simulated environment. It relies on a template library and only generates EIs with limited content. Yang et al. [41] introduced an approach to generate EIs from natural images. It utilizes superpixels as rendering primitives and edges as cues, but its generated results exhibit significant stylistic differences from EIs. Generative models [42] can be utilized for the task of stylized image generation, but they typically require a substantial amount of high-quality training data. In addition, universal style transfer (UST) models struggle to balance style loss and content loss when generating images with specific styles [6, 18, 21, 29].

There are two main challenges, low-bit color quantization and perceived difficulty adjustment, in generating high-quality EIs from natural images. In comparison to the 24-bit full color space of natural images, EIs have only 1-bit. We first need to perform color quantization on the natural image. However, existing color quantization methods perform poorly in low-bit color spaces. In addition, the applications of EIs span across various fields such as psychology and web security. Especially in the cognitive psychology research, researchers need to present test images with different perceived difficulty levels to participants.

In this paper, we propose HiEI, a universal framework for generating high-quality EIs from natural images. In particular, we employ this framework to explore an interesting problem: the potential of EIs as a CAPTCHA mechanism. HiEI consists of three components: human-centered color quantization module (TTNet), perceived difficulty control (PDC) module, and template vectorization (TV) module. TTNet learns to preserve rich visual cues in a limited color space by minimizing perceptual loss, producing human-centered color quantization results. Subsequently, the results are fed to the PDC module, where three parameters are set to quantitatively adjust the perceived difficulty and output the rendering templates. Finally, the TV module utilize Bézier curves to reconstruct the shape of the speckles in the rendering template, relizing the vectorization of the EIs.

The main contributions are summarized as follows:

- We propose HiEI, a universal framework for generating high-quality EIs from natural images. HiEI offer users the flexibility to control the perceived difficulty of the generated EIs, enhancing its practical applicability.
- We present a human-centered color quantization model TTNet to minimize the perceptual loss by preserving essential visual cues. Experimental results demonstrate that TTNet can perform better than other methods especially in the low-bit color space.
- We experimentally demonstrate that EIs generated by HiEI can effectively defend against attacks from the visual models based on deep networks, confirming its feasibility as a CAPTCHA scheme. In experiments, EIs significantly reduce the performance of multiple deep networks on tasks of object detection and image classification.

2 Related Work

Color quantization. MedianCut [16] proposed by Heckbert et al. is the first color quantization algorithm. The main idea is to recursively sort the pixels by color space and divide it along the median. Many variations exists of MedianCut algorithm, such as Center-cut [22]. Octree algorithm [13] proposed by Gervautz et al. is the first agglomerative color quantization algorithm and is based on the octree, a tree data structure in which each internal node has eight children. The popularity algorithm first builds a coarse color histogram of the input image using bit-cutting and then takes the set of most frequent colors in this histogram as the color palette. Hou et al. proposed a deep network-based color quantization model, ColorCNN [19], to minimize the accuracy loss by preserving the essential structur for deep networks. It demonstrates a higher classification accuracy than human-centered methods.

Universal style transfer. 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. Recent years have seen the rise of end-to-end style transfer models, with Gatys et al. [12] being among the pioneers who used CNNs for this task. These models can be

classified into three categories according to their evolutionary development: 1) models that can only render one style [12]; 2) models that can render multiple styles [7]; 3) Universal style transfer (UST) models that can render arbitrary styles [6, 18, 20, 21, 29, 43]. UST models are particularly useful when training samples are scarce. In the experimental section, we will compare the generated results of our HiEI with UST models.

3 Methodology

This section presents the proposed framework HiEI for generating high-quality EIs from natural images. It first describes the overall flow of HiEI, and then presents the implementations of three components: TTNet, the PDC module, and the TV module.

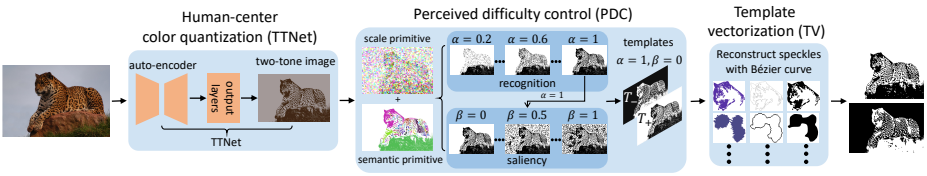

Fig. 2: Overview of HiEI. It consists of three modules, TTNet, PDC module, and TV module. TTNet is responsible for extracing the key visual content in 1-bit color space from the given natural image. The PDC module quantitatively adjusts the perceived difficulty of the rendering template through two parameters α and β. The TV module vectorizes the rendering templates.

3.1 Overview

Fig. 2 depicts the overall processing pipeline of HiEI. Given a natural image, HiEI first uses TTNet to reduce the color space of the image to 1-bit (the two-tone image) with the minimal perceptual. Subsequently, the PDC module extracts speckles from the two-tone image as the semantic primitive to controll the perceived difficulty. It further divides the two-tone image into patches (scale primitive). The division is made with approximately equal areas, using the edges from the two-tone images as the constraint. The PDC module controls object recognition difficulty by adjusting the proportion of patches using the parameter α. The parameter β is used to adjust the density contrast of patches between the foreground and the background, thus controlling object saliency. The TV module processes the output from the PDC module and produces two complementary rendering templates T_+ and T_-. It then employs Bézier curves to smoothly fit the contour of speckles in T_+ and T_- to generate a pair of vectorized EIs.

3.2 Human-centered color quantization

TTNet architecture. Inspired by the work of Hou et al. [19], we design the color quantization model TTNet as illustrated in Fig. 3. The first component is a U-net [33] auto-encoder that can extract abundant semantic information from the natural image I. The extracted information are fed into the linear convolutional 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. TTNet creates 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

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

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

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

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

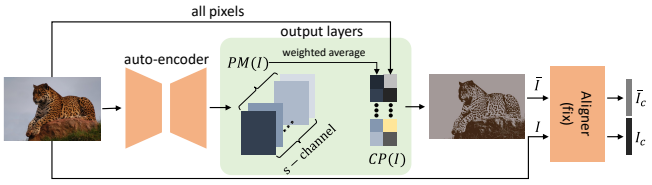

Fig. 3: Architecture of TTNet. It consists of U-net auto-encoder, output layer, and aligner. The U-net auto-encoder extracts semantic information from natural images, and then the output layer designs the color palette and creates the quantizated image. Finally, the aligner enables TTNet to reduce the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

Finally, we apply a pre-trained VGG19 ercoder [35] to serve as a fixed image content aligner. It takes the natural image I and the quantized image $\bar{I}$ as inputs and then outputs their content features I_c and $\bar{I}_c$. During the training process, TTNet will optimize the color quantization results by learning to reduce the difference between I_c and $\bar{I}_c$.

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

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

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

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

$\Phi_i(\cdot)$ 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. Fig. 4 depicts the visualized feature maps extracted from the first five layers of the pre-trained VGG19. 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 $N_l = 5$. We also conduct an experimental analysis to assess the impact of varying N_l settings on results of color quantization in supplementary materials.

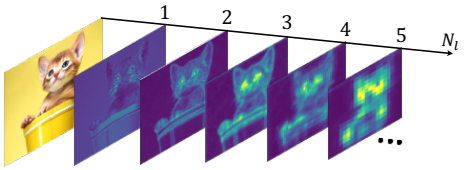

Fig. 4: Explanation of parameter N_l setting in the loss function of TTNet.

3.3 Perceived difficulty control (PDC)

The PDC module is designed to quantitatively adjust the the perceived difficulty of generated results, influencing the perception of humans on EIs. The challenge of PDC is how to quantify the perceived difficulty of EI, as the perceived difficulty is vague and subjective. Inspired by the psychological experiments on the perception of EI conducted by Li et al. [25], we split the perceptual process into two parts (object detection and object recognition) and take two different control primitives to quantify the perceived difficulty.

Control primitives. We utilize the image matting model GFM [28] to separate the foreground I_f and the background I_b of I. Next, we obtain a set of speckles E by extracting the four-connected regions from the two-tone image. Subsequently, we take the edge information contained in the two-tone image as the constraint to divide I into a set of patches with approximately equal areas [1],

denoted as A. The speckle and the patch will serve as the semantic primitive and the scale primitive, respectively. We use I_f and I_b to divide E into E_f and E_b within the foreground and background regions, respectively.

Recognition control. Humans use surface information (color, texture, luminance) and shape for object recognition [39]. Early object recognition theory [4] prioritizes shape over surface information. However, this perspective falls short in elucidating the distinction in recognizing animals with similar forms, such as horses and zebras. The limitation becomes evident when the shape of a zebra is presented in isolation, leading observers to erroneously identify it as a horse. The "shape + surface" theory [36] suggests that the importance of shape and surface information depends on structural differences between objects. Before adjusting the perceived difficulty, we need to determine which cues are more critical for recognizing foreground objects.

Two representative examples shown in Fig. 5. The edges within the foreground region play a role in this process. We first frame the foreground object with a bounding box and divide it into $M * N$ grids, and then count the number K of grids that contain edges. If $\frac{K}{M*N} \geq \omega$ (*e.g.*, $\omega = 0.4$), the texture is more important in object recognition (Case 1); otherwise, the shape (Case 2). We set the parameter α to adjust the proportion of key discriminative cues. For Case 1, we keep the α-proportion patches randomly selected from each speckle in $E_f = \{e_{f_1}, ..., e_{f_n}\}$. For example, we randomly select $m * \alpha$ patches from the i-th speckle $e_{f_i} = \{a_1, ..., a_m\}$. For Case 2, we perform α-proportion sampling on each speckle that contains the contours of the object.

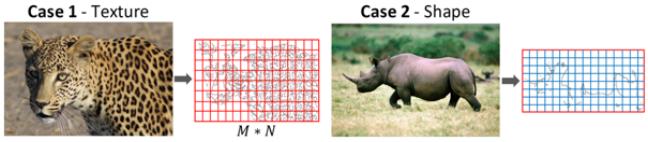

Fig. 5: Illustration of two representative cases. The edges are used to determine which cues (texture and shape) are more critical for recognizing the foreground object.

Saliency control. After the process of the recognition control, the updated set of speckles located within the foreground is denoted as $E_f^\alpha = \{e_{f_1}^\alpha, ..., e_{f_n}^\alpha\}$, and the density of patches within the foreground I_f is defined as

$$D_f = \frac{\sum_{i=1}^{n} Num(e_{f_i}^\alpha)}{Area(I_f)}, \tag{5}$$

where $Num(\cdot)$ is the function to obtain the number of elements in a speckle, and $Area(\cdot)$ is the function to calculate the area of a region. To control the saliency of the object, We set the parameter β to adjust the contrast of patch density between the foreground and the background. The patch density of the

background D_b is calculated as

$$D_b = D_f \cdot \beta. \tag{6}$$

The number of patches originally contained in the background $E_b = \{e_{b_1}, ..., e_{b_k}\}$ is $\sum_{i=1}^{k} Num(e_{b_i})$.Theoretically, the number of patches contained in the background should be $D_b \cdot Area(I_b)$. If $D_b \cdot Area(I_b) > \sum_{i=1}^{k} Num(e_{b_i})$, 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 sort the speckles in E_b in descending order based on the number of patch, and randomly place speckles of the sorted E_b in empty areas of the background. Finally, the updated set of speckles in background is denoted as E_b^{β}.

Rendering templates. After setting the parameters α and β, we obtain a binary image. Inspired by Rubin vase [14], we obtain two complementary rendering templates T_+ and T_- by inversing the color of the binary image. Although two rendering templates contain the same edge information, variations in color significantly influence the figure-groud segregation process. Therefore, observers will have different difficulties in perceiving the target object from the EIs generated using these two templates.

3.4 Template vectorization (TV)

The distortion problem occurs when the user zooms in on a rendering template obtained based on a low-resolution natural image, influencing the perceptual quality of the observer. In HiEI, we solve the template distortion problem by vectorization. The method is the same for both rendering templates. Here, we take the template T_+ with $\alpha = 1$ and $\beta = 0$ shown in Fig. 6(a) to explain this process. For each speckle (*e.g.*, the one in Fig. 6(b)) in $E_f^{\alpha} \bigcup E_b^{\beta}$, we replace it with its patches shown in Fig. 6(c) and then extract the contour shown in Fig. 6(d). Next, we calculate the contour curvature. As shown in Fig. 6(e), we select two pixels, p_0 and p_1, on the contour from each side of the current position $p_c(x_c, y_c)$, and define the approximate curvature of p_c as $len_{(p_0,p_1)}/dis_{(p_0,p_1)}$, where $len_{(p_0,p_1)}$ is the length of the contour segment between p_0 and p_1, and $dis_{(p_0,p_1)}$ is the distance between them. We then obtain the sampled contour pixels shown in Fig. 6(f) based on the normalized curvature. Finally, we fit these sampled pixels using Bézier curves to obtain the vectorized sepckle shown in Fig. 6(g).

4 Experiment

4.1 Experiment setup

Settings. In HiEI, TTNet 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

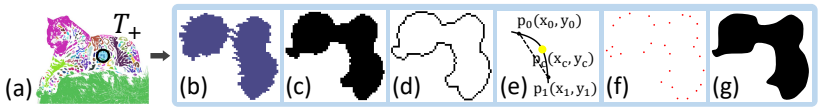

Fig. 6: Explaination for the process of template vectorization. (a) rendering template T_+; (b) an example of spckle in T_+; (c) patches contained in (b); (d) contour of (c); (e) approximate curvature; (f) sample pixels based on curvatures; (g) vectorized speckle.

of main memory. We adjust the perceived difficulty of generated results by parameters α, β and rendering templates T including T_+ and T_-. Here, we use $\text{HiEI}_{(\alpha,\beta,T)}$ to denote HiEI with different settings. In Table 1, we take some examples to explain the meanings of the notations related to these three parameters.

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

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 $[0, 1]$
T	$+$	only use the template T_+
	$-$	only use the template T_-
	$+/-$	randomly use a template T_+ or T_-

Datasets. We conduct experiments on the following three publicly available datasets. Animal 2K [28] 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. PASCAL VOC2012 [10] is a classical dataset for multiple computer vision tasks such as image classification, object detection, and image segmentation. It contains 20 classes of objects with a total of 11,530 images. STL-10 [8] is a dataset for the tasks of image classification. It has 10 classes of object images with 96×96 low resolution, 500 training images and 800 test images for each class.

4.2 Comparison with other EI generation methods

We compare HiEI with two existing EI generation methods: ArtThres [40] and EdgeEI [41], and five state-of-the-art UST models: AdaConv [6], WCT [29], AdaIN [21], QuantArt [20], InST [43]. 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. The core of AdaIN is an adaptive instance normalization layer, which aligns the mean and variance of content image features with those of style image 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.

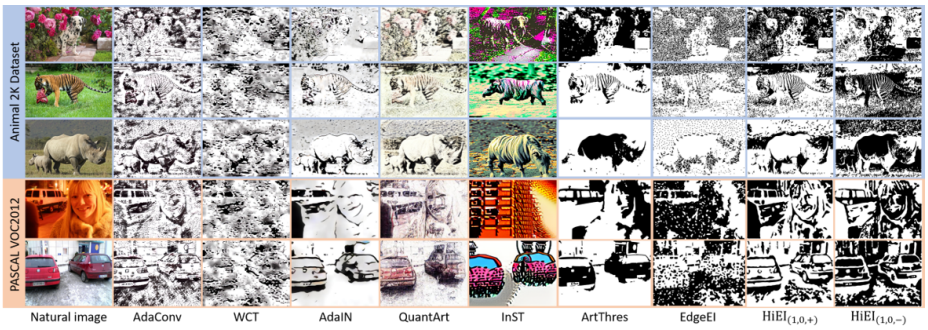

Fig. 7: Qualitative comparison of generated EIs between our HiEI and other seven methods including five state-of-the-art UST models and two original EI generation methods.

We observed color differences from the generated results in Fig. 7. The results of ArtThres, EdgeEI, and HiEI are binary images, but those of UST models are colorful or gray images. To fairly compare these methods, we first utilized TTNet to reduce the color space of results to 1-bit. Fig. 8 depicts the qualitative comparison between the binarized results of five UST models and the results of ArtThres, EdgeEI, and HiEI. '+bw' denotes the binarization operation on results of UST models. Given the distinctions in the perception processes of humans and deep neural networks, we further conduct experiments to quantitatively evaluate these binary results based on the deep neural network and the human, respectively.

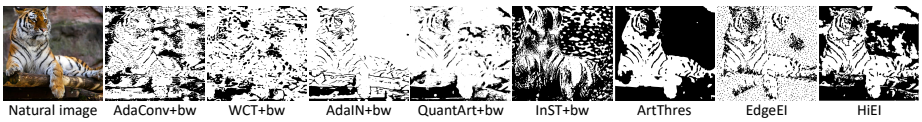

Fig. 8: Qualitative comparison between the binarized results of five UST models and the results of ArtThres, EdgeEI, and HiEI.

Evaluation based on deep network. The algorithm of stylized image generation has two inputs: a style image and a content image, using the style pattern of the former to render the latter. Following studies [21, 29], we use two metrics including the content loss and the style loss to 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 *Dalmatian Dog*. 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. HiEI exhibits significantly lower content loss and style loss than other methods. We find that although the generated results of $\text{HiEI}_{(1,0,+)}$ and $\text{HiEI}_{(1,0,-)}$ have the same edge information, $\text{HiEI}_{(1,0,-)}$ is slightly lower in content loss and style loss than $\text{HiEI}_{(1,0,+)}$. This is attributed to the exchange of foreground and background colors, which amplifies the loss of luminance features in EIs.

Table 2: Quantitative comparisons between the binarized results of five UST models and the results of ArtThres, EdgeEI and HiEI.

Methods	AdaConv	WCT	AdaIN	QuantArt	InST	ArtThres	EdgeEI	$\text{HiEI}_{(1,0,+)}$	$\text{HiEI}_{(1,0,-)}$
Content loss	1.16	1.34	1.28	1.39	1.46	1.29	1.36	**1.02**	<u>1.05</u>
Style loss	0.28	0.74	0.82	0.68	0.31	0.59	0.78	**0.19**	<u>0.24</u>

User study. We recruited 100 participants in the study (mean age = 22.8 years; 50 female). The participants come from the School of Computer Science, School of Psychology, School of Life Sciences, and School of Mathematics. None of the participants had visual cognitive impairment. We prepared the style image *Dalmatian Dog*, 100 natural images from the Animal 2K dataset and 100 natural images from the PASCAL VOC2012, and their corresponding binary results as shown in Fig. 8. We presented subjects with the style image and one content image at a time, as well as seven generated results with a randomly disrupted order. The participants were required to complete the following two tasks. 1) Content preference: please select the one that is closest to the natural image in content. 2) Style preference: please select the one that is closest to the *Dalmatian Dog* in style. Finally, we collect 40,000 feedbacks from subjects, and the statistical results are listed in Table 3. HiEI outperforms the other UST models in terms of both the style preference and the content preference.

4.3 Feasibility of EI-CAPTCHA

Image-based CAPTCHA uses vision tasks to determine whether the current user is a human or a malicious program. For example, in famous Google reCAPTCHA v2, the task related to object detection is "*select all squares containing some parts of an object*", and the task related to image classification is "*select all*

Table 3: Statistical results of the user study.

Methods	AdaConv	WCT	AdaIN	QuantArt	InST	ArtThres	EdgeEI	$\text{HiEI}_{(1,0,+/-)}$
Content preference	4.27%	0.14%	1.33%	2.65%	0.03%	0.48	1.16%	**89.94%**
Style preference	5.42%	13.15%	6.85%	16.57%	3.53%	2.81%	2.96%	**48.71%**

images containing a specific class of objects". Inspired by these two CAPTCHA tasks, we explored the feasibility of EI-CAPTCHA by conducting the following experiments on EIs generated by HiEI.

Object detection on EIs. Object detection methods can be broadly classified into traditional and end-to-end approaches. Traditional approaches usually rely on manually extracted features, including edges and color. Fig. 9 displays the performance of Edge Boxes [44], a classical traditional method, on natural images and EIs. The detection result marked with the yellow frame in Fig. 9(a) indicates that Edge Boxes can successfully detect the dog in the natural image. However, the result in Fig. 9(b) demonstrates that it fails to detect the object in the EI. In Fig. 9(c), the edge extraction results of the EI are fragmented. It is difficult to separate the object edges from the noise.

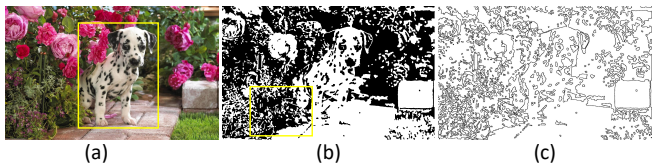

Fig. 9: Edge Boxes fails to detect the object in the EI. (a) Detection result on the natural image. (b) Detection result on the EI. (c) Edges of the EI in (b).

In the past decade, end-to-end deep visual models have become the mainstream in object detection. YOLO, a classical one-stage detection model, has performed impressive results on the task of object deteciton, and it is also used in the Google reCAPTCHA v2 solver. In this experiment, we test the ability of YOLO v8 [31] to detect objects on EIs. We firstly generate the corresponding emerging-style PASCAL VOC2012, denoted as $\text{PASCAL}_{\text{EI}}$. For each natural image in the training set, we use $\text{HiEI}_{(1,\{0,0.5,1\},+/-)}$ to generate its three EIs with different perceived difficulty. For each image in the test set, we use $\text{HiEI}_{(1,[0,1],+/-)}$ to generate its EI. We use the officially available pre-trained YOLO v8, and then fine-tune it on PASCAL VOC2012 and $\text{PASCAL}_{\text{EI}}$, respectively. 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 PSACAL contains one or more objects and we select the one with the largest area as the target object. In the final detection result, if the

IoU (Intersection over Union) is greater than 0.6 between the bounding box and the ground truth of the target object, and the recognition result is correct, then YOLO successfully solves this sample. Table 4 lists the success rate of YOLO in solving this task. YOLO_{NI} and YOLO_{EI} refer to the models fine-tune on PASCAL and $\text{PASCAL}_{\text{EI}}$, respectively. On the test set ($\text{testval}_{\text{EI}}$) of $\text{PASCAL}_{\text{EI}}$, YOLO_{NI} performs significantly worse compared to its performance on the original test set (testval) of PASCAL. Despite a slight improvement in the performance of YOLO_{EI} on $\text{testval}_{\text{EI}}$, its accuracy remains low at 29.1%. We also conduct tests to evaluate human performance on EIs. We randomly select 100 images from both testval and $\text{testval}_{\text{EI}}$ for participants to find the target objects. Participants are required to mark the target object with only one wireframe on a task image. We use the same criteria as YOLO to judge these results, and the success rate of human reached 89%. We conclude that EIs can effectively defend against the attacks from the deep network-based CAPTCHA solver with little influence on human vision.

Table 4: Performance of YOLO on object detection with the PASCAL and the $\text{PASCAL}_{\text{EI}}$.

Data	YOLO_{NI}	YOLO_{EI}	Human
testval	0.709	-	1
$\text{testval}_{\text{EI}}$	0.103	0.291	0.89

Image classification on EIs. We employ AlexNet [24], VGG19 [35] and Resnet34 [15] as the task networks for the image classification. The accuracy of the top-1 classification was used to evaluate their classification performance. We first used $\text{HiEI}_{(\{0.5,1\},0,+/-)}$ to generate the EIs on STL-10 dataset, denoted as $\text{STL-10}_{\text{EI}}$. We trained these three networks using the training set of STL-10, and then fine-tuned them on the train set of $\text{STL-10}_{\text{EI}}$. The results in Table 5 demonstrate that these three task networks exhibit poor classification performance even after being fine-tuned on EIs. We conclude, from the substantial performance disparities observed between deep neural networks and humans in the tasks of object detection and image classification, that EIs can serve as a CAPTCHA for distinguishing between humans and bots.

4.4 Ablation study

In this section, we conduct the ablation studies to justify the effectiveness of three modules in HiEI. Due to space constraints, the ablation experiments for the PDC and TV modules are presented in the supplementary materials.

TTNet. To demonstrate the effectiveness of TTNet, we replaced it with three other representative color quantization methods (*i.e.*, OcTree [13], MedianCut

Table 5: Performance of three task networks (AlexNet, VGG19, Resnet34) on the task of image classification with the STL-10 and the STL-10_{EI}.

Datasets	AlexNet	VGG19	Resnet34
STL-10	0.759	0.691	0.878
STL-10_{EI}	0.237	0.205	0.265

[16], and ColorCNN [19]) to provide two-tone images for HiEI. We conducted this experiment on 200 natural images in the test set of Animal 2K dataset. A qualitative comparison of four EIs generated with the same image is shown in Fig. 10(a). The EI generated using TTNet is closest to the natural images from the visual perception. Fig. 10(b) shows the results of quantitative evaluation based on content loss. TTNet enables HiEI to achieve lower content loss. In addition, we also set different sizes of color quantization space to test TTNet in supplementary materials. The experimental results demonstrate that TTNet exhibit superior results to the other three methods.

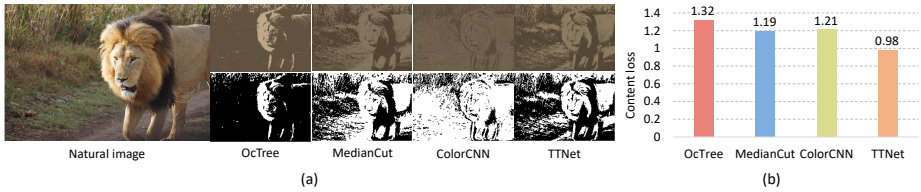

Fig. 10: (a) Qualitative comparison of EIs generate by HiEI combining four different color quantization methods. (b) Quantitative evaluation of EIs generated by HiEI equipped with these four methods.

5 Conclusion

In this paper, we proposed a universal framework, HiEI, to generate high-quality EIs from natural images. HiEI is equipped with the human-centred colour quantization module, the perceived difficulty control module, and the template vectorization module. These three modules not only enable HiEI to improve the quality of generated EIs, but also enhance its practicality. Based on the EIs generated by HiEI, we validated the feasibility of the EI-CAPTCHA on two vision tasks, object detection and image classification. Experimental results demonstrate that EIs generated by HiEI can significantly weaken the detection and classification performance of deep network-based visual models.

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

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. IEEE transactions on pattern analysis and machine intelligence **34**(11), 2274–2282 (2012)
2. Alqahtani, F.H., Alsulaiman, F.A.: Is image-based captcha secure against attacks based on machine learning? an experimental study. Computers & Security **88**, 101635 (2020)
3. Anderson, J.R.: Cognitive psychology and its implications. Macmillan (2005)
4. Biederman, I.: Recognition-by-components: a theory of human image understanding. Psychological review **94**(2), 115 (1987)
5. Cavanagh, P.: What's up in top-down processing. Representations of vision: Trends and tacit assumptions in vision research pp. 295–304 (1991)
6. Chandran, P., Zoss, G., Gotardo, P., Gross, M., Bradley, D.: Adaptive convolutions for structure-aware style transfer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7972–7981 (2021)
7. Chen, D., Yuan, L., Liao, J., Yu, N., Hua, G.: Stylebank: An explicit representation for neural image style transfer. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1897–1906 (2017)
8. Coates, A., Ng, A.Y., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: Gordon, G.J., Dunson, D.B., Dudík, M. (eds.) Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011. JMLR Proceedings, vol. 15, pp. 215–223. JMLR.org (2011), `http://proceedings.mlr.press/v15/coates11a/coates11a.pdf`
9. Van de Cruys, S., Damiano, C., Boddez, Y., Król, M., Goetschalckx, L., Wagemans, J.: Visual affects: Linking curiosity, aha-erlebnis, and memory through information gain. Cognition **212**, 104698 (2021)
10. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
11. Gao, S., Mohamed, M., Saxena, N., Zhang, C.: Emerging image game captchas for resisting automated and human-solver relay attacks. In: Proceedings of the 31st Annual Computer Security Applications Conference. pp. 11–20 (2015)
12. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2414–2423 (2016)
13. Gervautz, M., Purgathofer, W.: A simple method for color quantization: Octree quantization. In: New Trends in Computer Graphics: Proceedings of CG International'88. pp. 219–231. Springer (1988)
14. Hasson, U., Hendler, T., Bashat, D.B., Malach, R.: Vase or face? a neural correlate of shape-selective grouping processes in the human brain. Journal of cognitive neuroscience **13**(6), 744–753 (2001)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

16. Heckbert, P.: Color image quantization for frame buffer display. ACM Siggraph Computer Graphics **16**(3), 297–307 (1982)
17. Hegdé, J., Kersten, D.: A link between visual disambiguation and visual memory. Journal of Neuroscience **30**(45), 15124–15133 (2010)
18. Hong, K., Jeon, S., Yang, H., Fu, J., Byun, H.: Domain-aware universal style transfer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14609–14617 (2021)
19. Hou, Y., Zheng, L., Gould, S.: Learning to structure an image with few colors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10116–10125 (2020)
20. Huang, S., An, J., Wei, D., Luo, J., Pfister, H.: Quantart: Quantizing image style transfer towards high visual fidelity. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5947–5956 (2023)
21. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE international conference on computer vision. pp. 1501–1510 (2017)
22. Joy, G., Xiang, Z.: Center-cut for color-image quantization. The Visual Computer **10**, 62–66 (1993)
23. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
24. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012)
25. Li, J., Wei, H.: Important clues that facilitate visual emergence: Three psychological experiments. In: Proceedings of the Annual Meeting of the Cognitive Science Society. vol. 45 (2023)
26. Li, J., Wei, H.: Make use of mooney images to distinguish between machines and humans. In: Proceedings of the Annual Meeting of the Cognitive Science Society. vol. 46 (2024)
27. Li, J., Wei, H., Yang, S., Fu, L.: Emerging image generation with flexible control of perceived difficulty. Computer Vision and Image Understanding **240**, 103919 (2024)
28. Li, J., Zhang, J., Maybank, S.J., Tao, D.: Bridging composite and real: towards end-to-end deep image matting. International Journal of Computer Vision **130**(2), 246–266 (2022)
29. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Universal style transfer via feature transforms. Advances in neural information processing systems **30** (2017)
30. Mitra, N.J., Chu, H.K., Lee, T.Y., Wolf, L., Yeshurun, H., Cohen-Or, D.: Emerging images. ACM transactions on graphics (TOG) **28**(5), 1–8 (2009)
31. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
32. Roelfsema, P.R.: Cortical algorithms for perceptual grouping. Annu. Rev. Neurosci. **29**, 203–227 (2006)
33. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. pp. 234–241. Springer (2015)

34. Shi, C., Ji, S., Liu, Q., Liu, C., Chen, Y., He, Y., Liu, Z., Beyah, R., Wang, T.: Text captcha is dead? a large scale deployment and empirical study. In: Proceedings of the 2020 ACM SIGSAC conference on computer and communications security. pp. 1391–1406 (2020)
35. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
36. Tanaka, J., Weiskopf, D., Williams, P.: The role of color in high-level vision. Trends in cognitive sciences **5**(5), 211–215 (2001)
37. Tang, M., Gao, H., Zhang, Y., Liu, Y., Zhang, P., Wang, P.: Research on deep learning techniques in breaking text-based captchas and designing image-based captcha. IEEE Transactions on Information Forensics and Security **13**(10), 2522–2537 (2018)
38. Theeuwes, J.: Top–down and bottom–up control of visual selection. Acta psychologica **135**(2), 77–99 (2010)
39. Wei, H., Li, J.: Computational model for global contour precedence based on primary visual cortex mechanisms. ACM Transactions on Applied Perception (TAP) **18**(3), 1–21 (2021)
40. Xu, J., Kaplan, C.S.: Artistic thresholding. In: Proceedings of the 6th international symposium on Non-photorealistic animation and rendering. pp. 39–47 (2008)
41. Yang, C.H., Kuo, Y.M., Chu, H.K.: Synthesizing emerging images from photographs. In: Proceedings of the 24th ACM international conference on Multimedia. pp. 660–664 (2016)
42. Yang, S., Wang, Z., Wang, Z., Xu, N., Liu, J., Guo, Z.: Controllable artistic text style transfer via shape-matching gan. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4442–4451 (2019)
43. Zhang, Y., Huang, N., Tang, F., Huang, H., Ma, C., Dong, W., Xu, C.: Inversion-based style transfer with diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10146–10156 (2023)
44. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 391–405. Springer (2014)