The Visual Computer (2022) 38:963–975
https://doi.org/10.1007/s00371-021-02061-1

ORIGINAL ARTICLE

TICS: text–image-based semantic CAPTCHA synthesis via multi-condition adversarial learning

Xinkang Jia[1] · Jun Xiao[1] · Chao Wu[1]

Accepted: 3 January 2021 / Published online: 9 February 2021

Abstract

CAPTCHA is used to distinguish humans from automated programs and plays an important role in multimedia security mechanisms. Traditional CAPTCHA methods like image-based CAPTCHA and text-based CAPTCHA are usually based on word-level understanding, which can be easily cracked due to the recent success of deep learning techniques. To this end, this paper proposes a text–image-based CAPTCHA based on the cognition process and semantic reasoning and a novel model to generate the CAPTCHA. This method synthesizes three features: sentence, object, and location to generate a multi-conditional CAPTCHA that can resist the attack of the classification of CNN. A quantity of experiments has been conducted, and the result showed that the classification of ResNet-50 on the proposed TIC only achieves 3.38% accuracy.

Keywords Text–image-based CAPTCHA · security mechanism · Generative Adversarial Network · semantic image synthesis

1 Introduction

CAPTCHA (Completely Automated Public Turing Test to tell Computers and Humans Apart) is a standard multimedia security mechanism for distinguishing between human and automated programs. CAPTCHA is firstly invented in 1997 and widely used to improve the security for preventing the abuse of online services like phishing, bots, spam, and sybil attacks right now [32].

Existing CAPTCHA is generally classified into four categories: text-based CAPTCHA, image-based CAPTCHA, audio-based CAPTCHA, and video-based CAPTCHA, whereas the text-based and image-based are the two most popular methods. Text-based CAPTCHA schemes ask users to recognize a string of distorted characters with/without background, which is the most widely deployed and acceptable method up to now. Image-based CAPTCHA scheme is another popular method, which usually asks users to select one or more images with specific semantic meanings from a couple of candidate images.

The recent development of deep neural networks presents a remarkable performance in diverse perception tasks, which greatly enhance the CAPTCHA cracking techniques. The text-based CAPTCHA is defeated by some CAPTCHA solver based on machine learning such as generic attacks which targeted on multiple text-based CAPTCHA [2,22,34,35], and specialized attacks which targeted on one kind of text-based CAPTCHA [8]. The image-based CAPTCHA has the same problem either. Many attacks based on deep neural networks are proposed to defeat image-based CAPTCHA with high success, as a large number of reports present [3,29]. The defect existing in both text-based CAPTCHA and image-based CAPTCHA is the lack of the cognitive process. For the text-based CAPTCHA, it only requires users to input the corresponding characters in the CAPTCHA. For the image-based CAPTCHA, it requires users to choose the corresponding CAPTCHA images by the object. Both forms of CAPTCHA don't use cognitive ability. Hence, it's necessary to design a new robust and user-friendly CAPTCHA based on cognition to tackle the emerging challenge.

The difference between humans and machines is cognitive ability. The human can imagine a picture when they read a description, but machines can only classify images by extracting the features. To compare with the existing text-based and image-based CAPTCHA approaches focus-

✉ Chao Wu
chao.wu@zju.edu.cn

Xinkang Jia
xk_jia@zju.edu.cn

Jun Xiao
junx@cs.zju.edu.cn

[1] Zhejiang University, Hangzhou, China

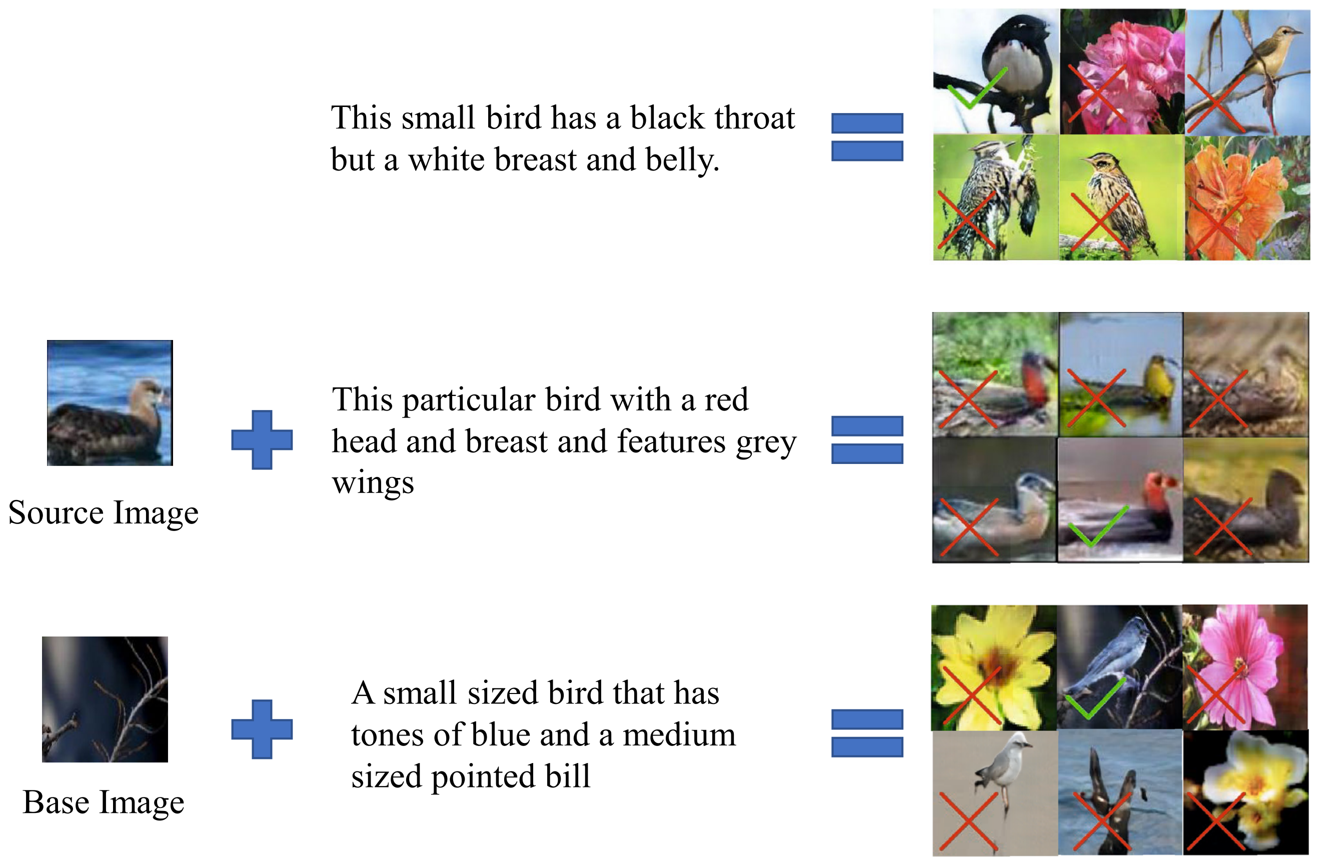
This small bird has a black throat but a white breast and belly.
Source Image
This particular bird with a red head and breast and features grey wings
Base Image
A small sized bird that has tones of blue and a medium sized pointed bill

Fig. 1 Example of the proposed text–image-based CAPTCHA. The top line is an example that corresponding CAPTCHA image directly synthesized with text description; the middle line is an example that corresponding CAPTCHA image is the attributed modification result of source image and text description; the bottom line is an example that corresponding CAPTCHA image is the object generation result of base image and text description

ing on the original matching process, we aim to generate the CAPTCHA by exploiting semantic common between the text and image, e.g., the CAPTCHA candidate images are highly related with a sentence description, and users need to make the selection via their cognitive ability and natural associative ability of cross-modality understanding between text and images.

Since the CAPTCHA is the text–image-based CAPTCHA, text description and images are two main elements. For the traditional generation of CAPTCHA, the method that designing the corresponding images based on text description by humans is time-consuming. Inspired by the great success of GAN on exploiting the semantic relation between text and image, we propose to leverage GAN to design a text–image-based CAPTCHA generation method. We have tried to use GAN models to generate CAPTCHA images from text descriptions as the basis of our design. To improve the robustness of the proposed CAPTCHA design and make the model fit our task better, we provide two further improvements. First, we adopt a semantic GAN method, which requires a source image and a piece of text description. The corresponding generated CAPTCHA image would be the source image with attributes that are manipulated with the text description. Second, we adopt multi-condition to generate an object on a background image with the given text description. Both of these two methods could increase the variation space for synthesized images. We also apply the stacked architecture to the model to improve the image quality.

Generally, the core contribution of our work is that we propose a new mechanism of CAPTCHA, TICS, which requires the cognition process and semantic reasoning, and a novel CAPTCHA generation model to combine text, object, and location (Fig. 1). We have evaluated our model on the Caltech-200 bird dataset and ResNet-50, and the results show the quality of our images and it can resist the attack of the classification of ResNet-50.

Table 1 Text-based CAPTCHA

Text-based CAPTCHA	Source	Features	Breaking method
	Qihu360	Rotation, distortion, different font sizes, different font styles	[35]
	EZ-Gimpy	Obscuration	[22]
	Discuz!	Different font styles, different font sizes	[35]
	Google	Rotation, distortion, waving	[35]
	Alipay	Rotation, distortion	[35]

2 Related Work

Our work mostly relates to the CAPTCHA schema and GAN-based image synthesis, and we will describe each of them, respectively.

2.1 CAPTCHA schema

Text-based CAPTCHA and image-based CAPTCHA are the most popular schema due to the convenience.

Text-based CAPTCHA requires users to recognize the string of numbers or characters with an obfuscated background. Due to its high efficiency and simplicity, it is the most widely used type of CAPTCHA in the world. This type of CAPTCHA image is usually of low quality with different kinds of noise and degradation applied to it, as shown in Table 1. The early text-based CAPTCHA is easily cracked by dictionary attacks, which contain a limited number of words in them [22], and it is also easily defeated by OCR (Optical Character Recognition) program. To prevent the attack of OCR, splitting and rotating [12] are a traditional process applied to the CAPTCHA. The characters in the CAPTCHA are usually split and rotated heavily so that the OCR program can not recognize them normally because of the limitation of the OCR technology. However, due to the power of deep learning and the development of image classification, many methods based on deep learning are proposed to crack text-based CAPTCHA. And the performance of these methods is also impressive [3,19,34,35]. A few examples of text-based CAPTCHA are shown in Table 1.

Image-based CAPTCHA is another popular scheme which usually demands users to select one or more images with specific semantic information from a group of images [5]. Recently, some variants of image-based CAPTCHA are designed, such as slide-based CAPTCHA, which requires users to slide a puzzle to the right part of an image [1] and click-based CAPTCHA which requires users to pinpoint the specific semantic parts of an image [11]. Examples of image-based CAPTCHA are shown in Fig. 2. Generally, image-based CAPTCHA presents visual information that users need to identify. The visual information or patterns are usually easy to be recognized and understood by human but difficult for the machine to classify. Compared with the text-based CAPTCHA scheme, the image contains more information and has a larger variation space. Due to better safety and convenience, image-based CAPTCHA is used wider than text-based CAPTCHA. IMAGINATI-ON [6] is a system for the generation of image-based CAPTCHA, which produces controlled distortions on randomly chosen images and presents them to the user for annotation from a given list of words. There are also some other methods proposed to generate image-based CAPTCHA [4,15–17]. Finally, many other techniques crack this type of CAPTCHA with high success [29,38].

2.2 Generative adversarial network

Besides traditional CAPTCHA schemes, we are also inspired by the Generative Adversarial Networks model [9]. Recently, GAN gains success in several tasks, such as image generation [7,23,24,26,36], learning representations for semi-supervised learning [20], discriminate image features [18], and domain transfer [39]. GAN is a novel method of training generative models. The training procedure consists of a generator and a discriminator, in which the goal of optimization is the min-max game between the two components.

However, a challenge of traditional GAN is that GAN takes a random Gaussian noise as a seed to generate some output image. This causes the output of GAN is out of the user's control. Conditional generative adversarial network

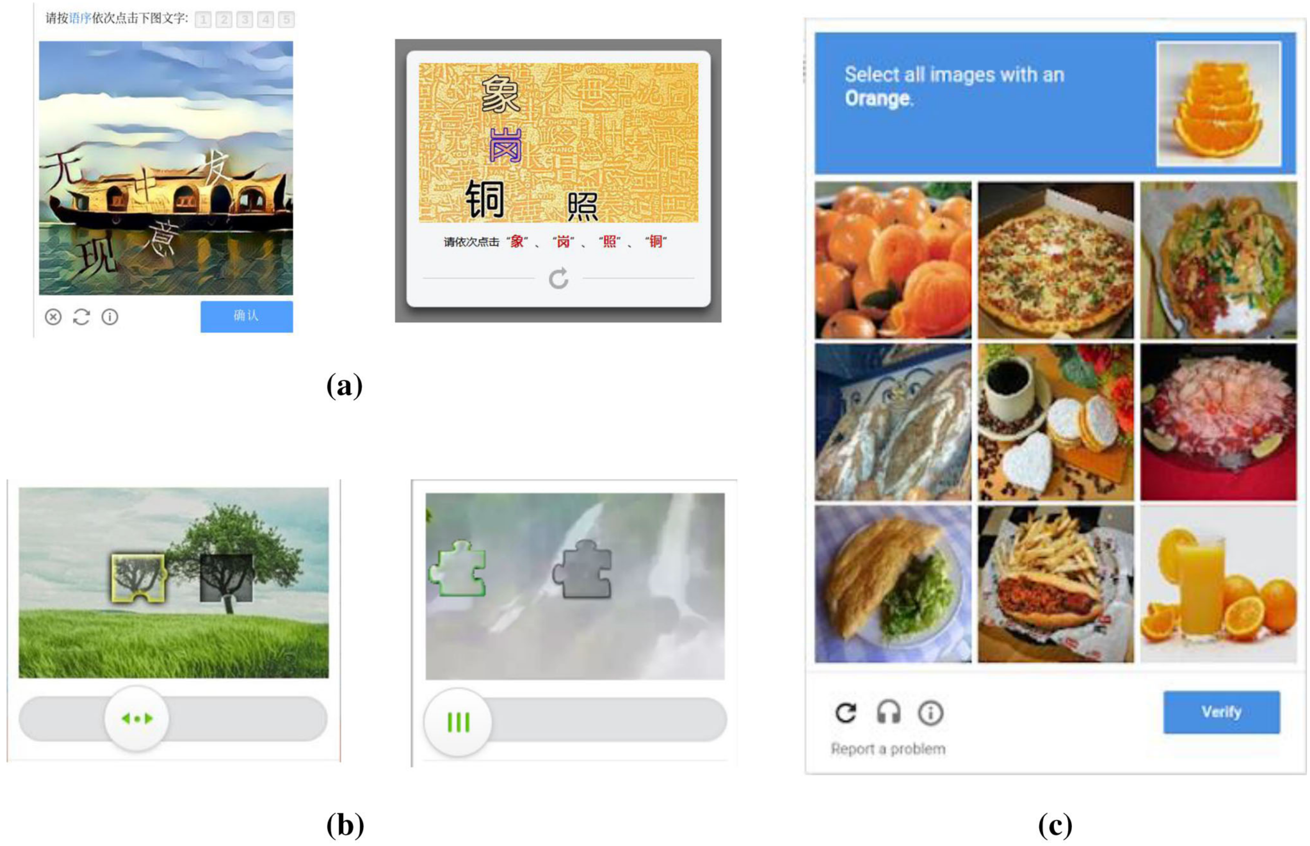
请按语序依次点击下图文字: 1 2 3 4 5
无 中 发 现 意
确认
象 岗 铜 照
请依次点击 "象"、"岗"、"照"、"铜"
(a)
(b)
Select all images with an Orange.
Verify
Report a problem
(c)

Fig. 2 Example of the image-based CAPTCHA. **a** Click image-based CAPTCHA. **b** Slide image-based CAPTCHA. **c** Traditional image-based CAPTCHA

[21] deals with this problem partly. Reed [26] proposes a method based on CGAN controlling the image generation by sentence. Kwon [15] proposes a CAPTCHA image generation system based on GAN. This system combines different characters in the CAPTCHA and generate a large number of CAPTCHAs in a short time, but it is used to generate the text-based CAPTCHA, which is an easy form to be attacked now. Cheng [4] applies neural style transfer to enhance the security for CAPTCHA design and proposes two image-based CAPTCHA, Grid-CAPTCHA, and Font-CAPTCHA. Grid-CAPTCHA asks the user to select all corresponding images according to the description from nine stylized images. Font-CAPTCHA asks users to click the right Chinese characters in an image according to the description. Kwon [16] also applies the style transfer method to deceive the machine while maintaining the human perception rate. The method combines the styles of different images while maintaining the content of the original CAPTCHA sample.

3 Methodology

In this part, we briefly introduce the whole workflow of our proposed CAPTCHA as Fig. 3 shows. After that, we introduce the image synthesis and CAPTCHA mechanism in detail.

In our workflow of the proposed CAPTCHA, we first take several types of inputs such as text, image, and location information to generate the candidate images using our designed image synthesis. Then, we form the final CAPTCHA from the candidate images with our CAPTCHA mechanism, and we introduce the mechanism in detail. Finally, we use the CAPTCHA for user authentication.

3.1 Text–image synthesis

A straightforward idea to generate an image matching a text description is to take text description as input by GAN. To our knowledge, this is the first attempt to apply the text to image translation in the CAPTCHA generation task. But the information of a semantic sentence is limited, which means that some attributes without description cannot be controlled. In order to improve our mechanism, we take a source image as

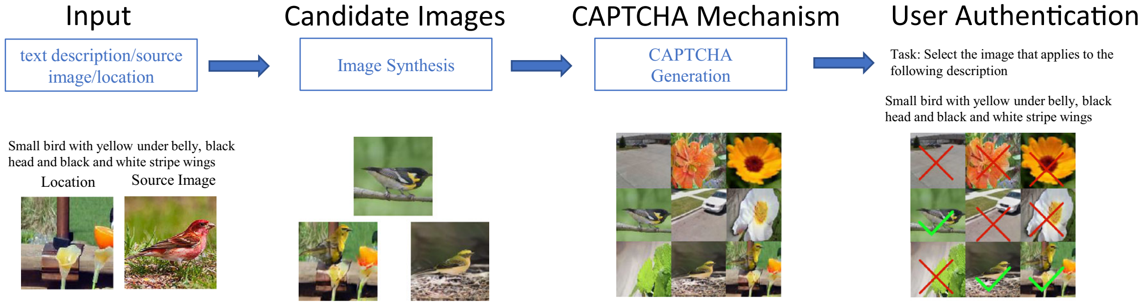
Input
text description/source image/location
Candidate Images
Image Synthesis
CAPTCHA Mechanism
CAPTCHA Generation
User Authentication
Task: Select the image that applies to the following description
Small bird with yellow under belly, black head and black and white stripe wings
Small bird with yellow under belly, black head and black and white stripe wings
Location
Source Image

Fig. 3 Workflow of the proposed CAPTCHA

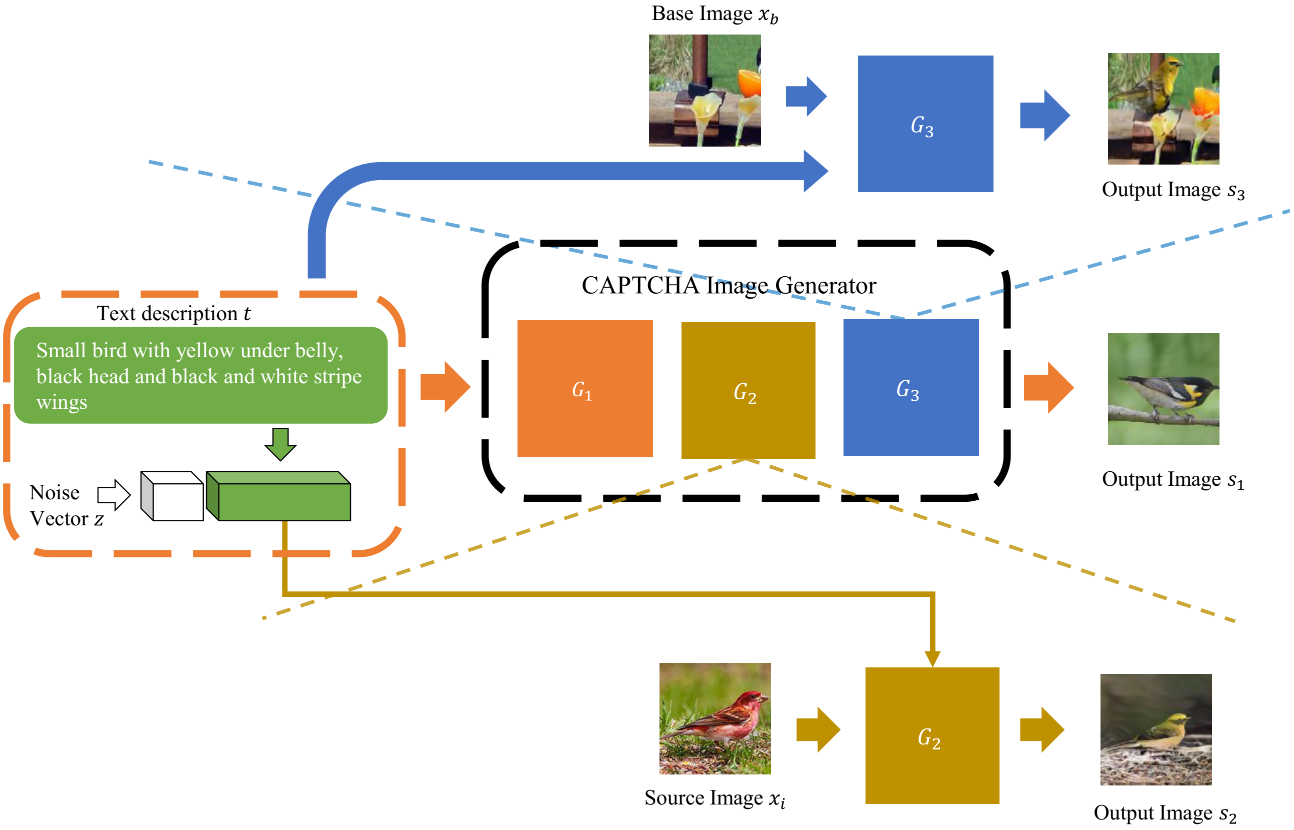

Fig. 4 Model architecture of the image-based CAPTCHA generation. The orange-colored network is the generation of the semantic images which take text description and noise as input. The yellow-colored network is the generation of the image-based semantic images which take real images or the synthetic images and text description as input. The blued-colored network is the generation of the location-based semantic image which take base image, noise, and text description as input

a condition into the model and modify one or more attributes by a text description and maintain the irrelevant parts of the source image. On the semantic level, it is acceptable for a human to understand the image, and it is hard for a machine to extract the features of all attributes directly. Furthermore, we take the location as a new condition to generate the object on an image as a background. The input conditions of our model are progressive from text, image to location. Therefore, our model is able to improve the variety of the image, where we can add text information as cognition challenges.

As shown in Fig. 4, the proposed model's architecture consists of three components based on GAN:

$$s_{1,2,3} = \begin{cases} G_1(t, z) \\ G_2(t, x_i) \\ G_3(t, z, x_b) \end{cases} \tag{1}$$

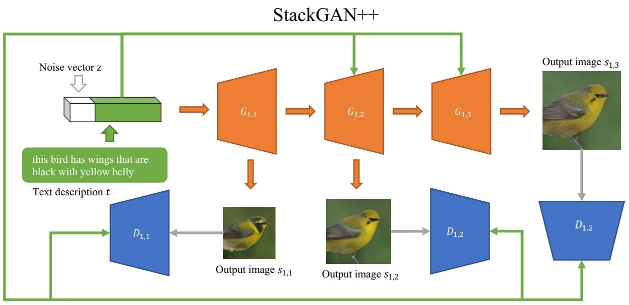

Fig. 5 Model architecture of semantic image generator

As shown in Eq. 1, G_1, G_2, and G_3 stand for the basic Semantic Image Generator (SIG), Image-based Semantic Image Generator (ISIG), and Location-based Semantic Image Generator (LSIG). The z, t, x_i, and x_b stand for the noise, text description, source image, and the background image.

3.1.1 Semantic Image Generator

In this subsection, we introduce our Semantic Image Generator that aims to generate image-based CAPTCHA that contains the semantic information from sentences. In addition, the created images should also promote image quality, which means the result must be clear enough for users to recognize.

As shown in Fig. 5, we adopt StackGAN++ [37] model to tackle this issue, which is able to generate images containing complex semantic information from a sentence with high resolution. StackGAN++ [37] consists of three stages of generation tasks. The first stage of generation task $G_{1,1}$ is to take a noise z and text description t as input pair to generate a low-resolution image $s_{1,1}$, which focuses on the rough shape and basic color of the object. The second and third stages of generation task $G_{1,2}$ and $G_{1,3}$ are to correct defects in the low-resolution images from the previous generators and complete details of the object by understanding the sentence t again to produce a high-resolution image $s_{1,3}$.

The model of the first stage of basic Semantic Image Generator is based on conditional GAN [21] framework conditioning on the text description. As for the process of the generator, the text description t is captured by the method named conditioning augmentation [36]; then, it concatenates with a random noise z as the input. The generator produces images $s_{1,1}$ based on text features t by a series of up-sampling blocks.

The generator of the second and third stages of StackGAN++ are based on an encoder–decoder architecture with several residual blocks [10]. The hidden features from the previous generators contain the image features like shape, color, or other basic information of the object. In the second and third stage generators, these hidden features concatenated with text features t again are fed into several residual blocks to learn multi-modal representations across image and text features again. The image generation of these two stages is the same as the former stages by a series of up-sampling blocks.

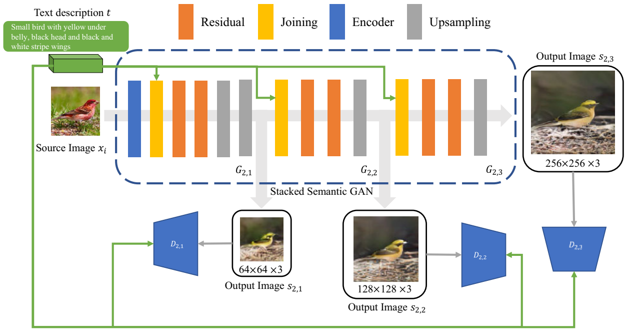

Fig. 6 Model architecture of image-based semantic image generator

For the discriminator, it extracts the features from images $s_{1,1}$, $s_{1,2}$, and $s_{1,3}$, respectively, by a series of down-sampling blocks and then distinguishes between the real images and fake images based on semantic features t.

3.1.2 Image-based Semantic Image Generator

In order to expand the variety of images and improve the cognition challenge of the synthetic images, we propose a model called stacked semantic GAN based on the method [7] to design our Image-based Semantic Image Generator, where Fig. 6 shows the general model architecture. This method takes an image x_i as input to generate realistic images s_2 matching text description t, and the image also maintains other image features that are irrelevant to the text description.

As shown in Fig. 6, the stacked semantic GAN consists of three generators in a tree-like structure. Different branches of the model generate different scales of images from low resolution to high resolution. The stacked semantic GAN takes the source images x_i as the input and has three generators to produce images varied from low resolution to high resolution. The network architecture of the first generator is based on an encoder–decoder structure. The encoder architecture is a pre-trained convolution network based on VGG network [28] to extract the features from the source images, such as background, object. The decoder consists of joining blocks, two residual blocks, and one up-sampling blocks. After the extraction of the encoder, the image features join with the text features in the joining blocks. We use several residual blocks to make the network deeper for better extracting the hidden features behind the feature maps. Finally, we use up-sampling blocks to compute the hidden features $h_{2,1}$ for the later image generation of high resolution. The decoder then generates the low resolution 64 × 64 × 3 images based on both text features and image features. The second and third generators are similar to the first one, but they take the hidden features $h_{2,1}$ and $h_{2,2}$ as inputs, which are computed by the first stage and second stage of generators, respectively. And scales of images generated by the second generator and third generator are 128 × 128 × 3 and 256 × 256 × 3.

The choice of the source images also influences the performance of the stacked semantic GAN. On the one hand, compared with the StackGAN++ [37], the latent representation can be extracted directly from the real images, which is more plausible and reasonable. Therefore, the images features which are irrelevant to the text description are more realistic and easy to be understood for humans, which can be used to improve cognition challenges. On the other hand, the semantic GAN can also take the output of StackGAN++ as the input to expand the diversity of images heavily and avoid some failing outputs of StackGAN++, which do not match the text description or mode collapse by using some normal synthetic images to generate again in semantic GAN, because the semantic GAN can extract the hidden features again and concatenate the hidden features with the same or other text features, which can make it to generate a new plausible image again. Therefore, compared with the output of the first CAPTCHA generator based on StackGAN++ [37], the variation space and image quality of the images from our modified semantic GAN can be improved again.

3.1.3 Location-based Semantic Image Generator

For the Image-based Semantic Image Generator, we aim to improve the background information from real images and modify attributes of images by text descriptions. But it doesn't add more information to the image. In order to expand the variety of images again and improve the cognition challenge impressively, we crop a location from a real image as a base image and generate object on the base image, which means we can add many objects on different locations of one base image. We adopt the MC-GAN [23] to achieve this goal, and it can maintain the latent representations of base images and generate images from a random noise meeting the text description on the base images.

As shown in Fig. 7, the model of Location-based Semantic Image Generator is built on conditional GAN [21] with feature extractor and synthesis block. Similar to Semantic Image Generator, Location-based Semantic Image Generator also concatenates text features t with a noise vector z as the seed feature map. After that, the seed feature map is fed into the synthesis block, which generates the images s_3 with text description t and combines the features extracted from base image x_b. The synthesis block is used to maintain the background features extracted from the base image and generate the objects on the location of the base image. For the discriminator D_3, it extracts the features from images s_3 and s_m by a series of down-sampling blocks and then distinguish between the real images and fake images based on semantic features t.

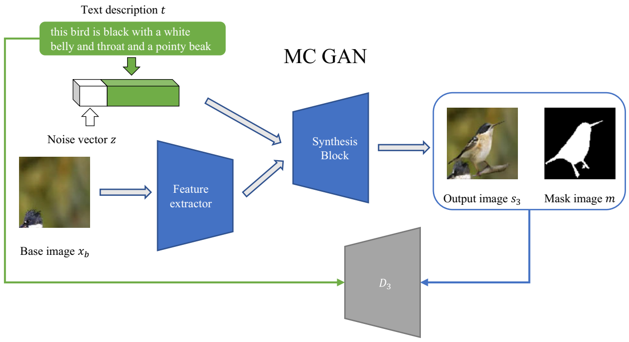

Fig. 7 Model architecture of location-based semantic image generator

3.2 Text–image-based CAPTCHA

The CAPTCHA mechanism is one of the most important parts of the workflow of our proposed CAPTCHA. As shown in Fig. 8, our CAPTCHA generation process is based on material images and generation rules. Firstly, we use offline text to image synthesis which are StackGAN++ [37], Stacked Semantic GAN and MCGA-N [23] to generate a number of images as the CAPTCHA materials. But the diversity of images generated by a text description is very high which may cause that some images generated by similar text descriptions look totally different from each other. In order to mitigate these problems and control the detail of images, we use the image relationship based on the text description attribute to control the details of the images. We manipulate some specific attributes in the same text description to generate some similar images but different in the edited attributes. Besides, we also adopt some images as basic disturbance terms that are irrelevant to the text description at all. Then, we combine these three parts as our generation rule to get the correct CAPTCHA form finally. The formal CAPTCHA will contain the only image matching the text description, similar images but not matching the text description and the irrelevant images, and users need to select the only images matching the text description.

4 Experiments

In this section, we first introduce the dataset used in the experiments and the training and implementation details, then introduce our evaluation metrics, and finally analyze the performance of our model.

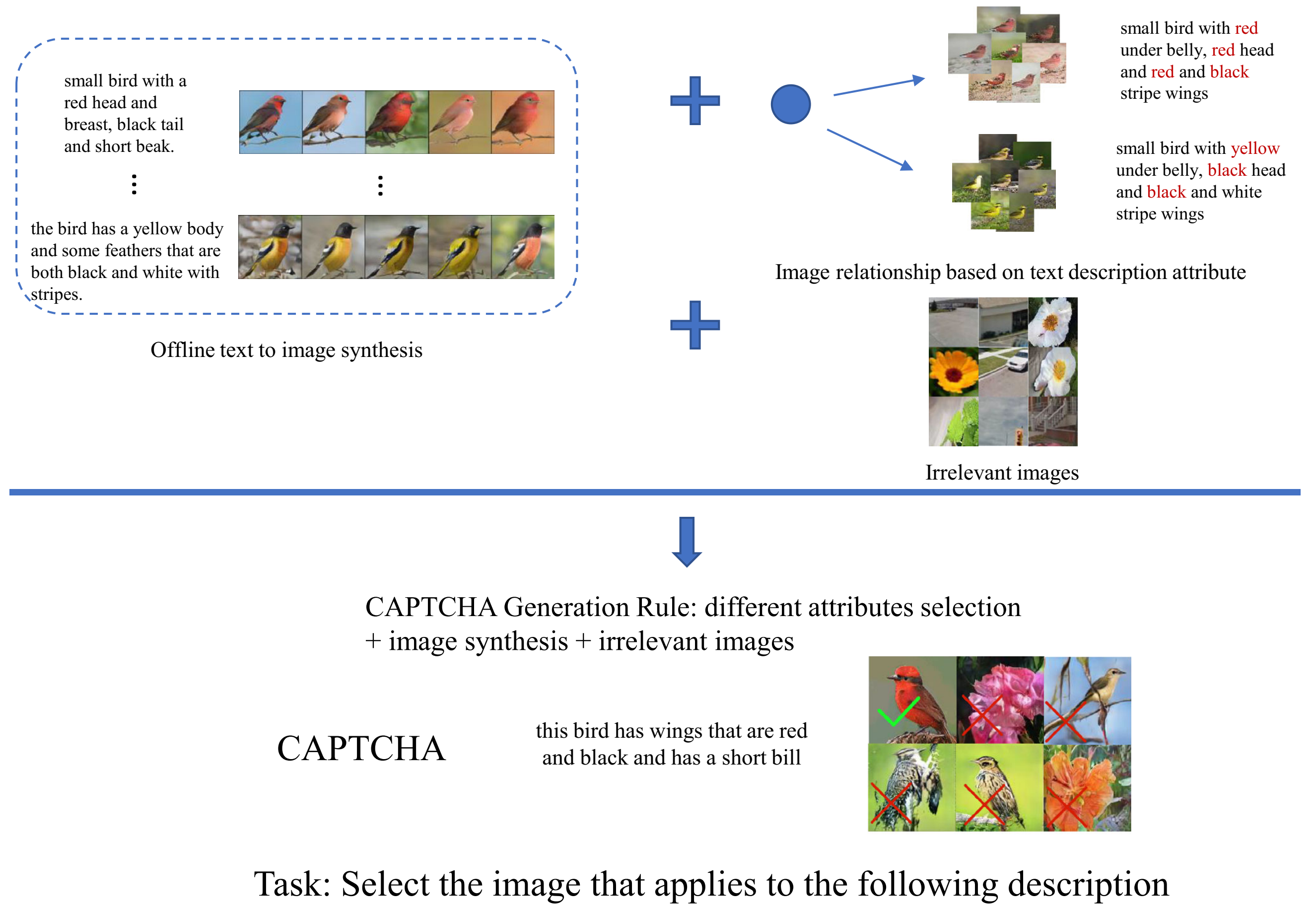
small bird with a red head and breast, black tail and short beak.
the bird has a yellow body and some feathers that are both black and white with stripes.
Offline text to image synthesis
small bird with red under belly, red head and red and black stripe wings
small bird with yellow under belly, black head and black and white stripe wings
Image relationship based on text description attribute
Irrelevant images
CAPTCHA Generation Rule: different attributes selection + image synthesis + irrelevant images
CAPTCHA
this bird has wings that are red and black and has a short bill
Task: Select the image that applies to the following description

Fig. 8 Workflow of CAPTCHA generation process

4.1 Datasets and implementation details

To fairly evaluate the performance of the proposed method, We conduct experiments on the publicly available datasets (Caltech-200 bird) [33] with many images and enough labeled sentences. Next, we introduce the detail of the dataset.

4.1.1 Caltech-200

Caltech-200 bird dataset [33] consists of 200 categories of 11,788 bird images and gives the ground truth segmentation mask maps for all the bird images. Each category of bird images has a training group and a test group. For the text description, we use the captions dataset from Reed et al. [26] which contains 10 captions for each image. Every caption describes the specific information of a bird such as shapes, colors, and parts of birds. For the base image, we use the image patches from MC-GAN [23] which are cropped from the Caltech-200 dataset [33] excluding the bird part.

4.1.2 Implementations

In the training, all the hyperparameters and other parameters we use in Semantic Image Generator and Location-based Semantic Image Generator are all the same as the StackGAN++ [37], and MC-GAN [23]. And the implementation details of Semantic Image Generator and Image-based Semantic Image Generator are also the same as the StackGAN++ [37] and MC-GAN [23]. For the Image-based Semantic Image Generator, the input image x_i is first transformed to a $16 \times 16 \times 512$ feature tensor by the encoder layer and we use the VGG-16 [28] here. Then, this feature tensor is gradually transformed to $64 \times 64 \times 256$, $128 \times 128 \times 128$, and eventually $256 \times 256 \times 64$ tensors at different layers of the network by four up-sampling blocks. We use three 3×3 convolution layers to generate images from these features tensors. Text embedding t is also fed into the joining layer to ensure the condition information is not omitted and keep the hidden feature vector highly related to the text description. Between every two generators, we use two residual blocks to extract the image features deeply. As for the optimizer, we use the ADAM [13] optimizer and set $beta1 = 0.5$ with a learning rate of 0.0002 for the whole model.

4.2 Evaluation metrics

In this subsection, we introduce the settings of our experiments first and then analyze the performance of our model.

4.2.1 Image quality evaluation

It is difficult to evaluate the performance of a generative model. The recent proposed assessment approach "Inception Score" is widely accepted as an evaluation method [27]. And we take this equation for image quality evaluation.

$$\text{IS} = \exp\left(\mathbb{E}_{\mathbf{x}\sim p_g} D_{KL}(p(y \mid \mathbf{x}) \| p(y))\right) \tag{2}$$

where the $\mathbf{x} \sim p_g$ denotes one synthetic image sampled from the distribution of synthetic images p_g, $D_{KL}(p\|q)$ is the KL-divergence between the distributions p and q, $p(y \mid \mathbf{x})$ is the conditional class distribution, and $p(y) = \int_{\mathbf{x}} p(y \mid \mathbf{x}) p_g(\mathbf{x})$ is the marginal class distribution. And the $\mathbb{E}_{\mathbf{x}\sim p_g}$ is the expectation of the distribution of the synthetic images p_g. The intuition of this metric is that good models should generate diverse and clear images, which can be classified into one specific class by Inception model. Therefore, the KL istance between the marginal distribution $p(y)$ and the conditional distribution $p(y|\boldsymbol{x})$ should be large. In our experiments, we fine-tune a pre-trained Inception model [31] for each of our datasets. And we evaluate this metric on almost 30k synthetic samples for each model, and these samples are all in the 256 × 256 resolution.

4.2.2 Semantic image evaluation

In this part, we design a classifier to check whether our synthetic CAPTCHA images can resist the attack of classification. With the expansion of the variety in synthetic images, it is hard for the classifier to classify similar object correctly. In this experiment, the classifier does a multi-classification task. We train our classifier on the bird dataset and test it on the sample images which are generated from our model. Note that, in the practical situation, CAPTCHA solvers cannot acquire the CAPTCHA image in advance from the CAPTCHA service. So we do not train the classifier on our sample images but test it directly. Besides, we also test different scales of images which are generated by the Semantic Image Generator and Image-based Semantic Image Generator.

For the classifier, we use a pre-trained resnet-50 model [10] as the base classifier and then we train this classifier on the training set of bird dataset for 50 epochs. After that, we validate our classifier on the test set of the bird dataset. Finally, we test the classifier on the sample generated from our model.

For the test sample, we randomly choose 16 images for each sentence from the three generators and collect these images from the samples generated from the three generators as semantic image set, image-based semantic image set, and location-based semantic image set. We also set up three resolution sets for the Semantic Image Generator, image-based semantic generator, respectively.

Table 2 Image quality evaluation

Model	Inception score on birds
SIG	5.28 ± 0.09
ISIG	5.66 ± 0.03
LSIG	3.42 ± 0.05

Table 3 Semantic image evaluation

Model	Accuracy top 1 (%)	Accuracy top 5 (%)
Real image	76.99	94.12
SIG	3.38	11.82
ISIG	4.71	16.12
LSIG	4.54	16.71

4.3 Quantitative comparison

As Table 2 shows, the inception score of images of birds is high, which shows that the quality of our images is clear enough for the Inception Model [27] to classify. Moreover, our modified image-based semantic generator (Stacked Semantic GAN) gets a higher inception score than the StackGAN++ [37], which shows that the features from the real images can improve the image quality. The inception score of LSIG (MC-GAN) is much lower than the other two models because of the lack of stacked architecture, which shows that stacked architecture can improve the image quality too.

As Fig. 9 shows, the StackGAN++ may generate a failure image, but our Stacked Semantic GAN can take this failure image as the source image with a similar sentence as input to generate a successful image. It proves that the Stacked Semantic GAN can extract the useful visual information from a failure image and use another similar sentence to fix the failure image, and generate a successful image matching the text description.

As Table 3 shows, we train the classifier on the real dataset of birds to do a multi-classification task on top1 and top5 with 76.99% and 94.12%, respectively. The result of the classification of our synthetic images is much lower than the real image. The result of the SIG is the lowest with 3.38%. Because it can generate the objects which are totally differ-

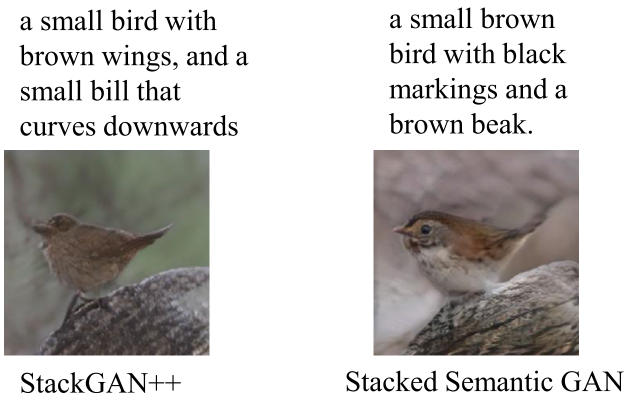
a small bird with brown wings, and a small bill that curves downwards
a small brown bird with black markings and a brown beak.
StackGAN++
Stacked Semantic GAN

Fig. 9 Example of the fix of stacked semantic GAN

ent from the real images according to the descriptions. The classifier which learns from the real dataset does not have the knowledge to deal with the new data. The results of the ISIG and LSIG are slightly higher than the SIG with 4.71% and 4.54%, respectively. The inputs of the ISIG and LSIG are based on real images, so the output may contain some similar elements like texture, shape. These elements can also be learned from the real images for the classifier. So the classifier can have a better classification on the images from LSIG and ISIG than the SIG. However, the features and attributes of synthetic images are not totally same as those of real images, so the result cannot be improved a lot. All samples used in this test are 256 × 256 × 3 resolution.

Besides of classification test, we also conduct a human evaluation to evaluate the semantic correlation of image. Ten candidates are recruited and required to rank the semantic correlation of images generated by different generators. The images from different generators are divided into 3 sets for ten candidates. We mix the one image with randomly picked target text with other images in the test set and ask the candidate to choose the correct one. After choosing the image, the candidates rank the images from 3 sets (3 for best, 1 for worst). The rank is based on the following criteria:

- Whether candidate choose the correct image form the test set.
- Whether the synthetic images match the text description
- Whether the synthetic images are realistic

Then, we average all human ranks to calculate the scores of our synthetic images, as shown in Table 4. The result of accuracy shows that our synthetic images can be selected easily by a human, which proves that our images are user-friendly. And the scores of these synthetic images also show that our synthetic images match the text description and realistic.

Table 4 Human evaluation on synthetic images

Model	Score	Accuracy (%)
SIG	2.3	93
ISIG	2.4	94
LSIG	1.3	91

5 Discussion

In this section, we discuss the intuition behind our method, security performance, the limitation, and the practical application for our CAPTCHA.

5.1 Intuition

The intuition behind our method is that we use the cognition ability of the human to distinguish human and machine. Cognition ability is essential for human to learn the cognitive skills which are needed in the acquisition of knowledge, manipulation of information, and reasoning. And the core part of our method is a semantic reasoning test for the human. The users using our CAPTCHA need to understand the meaning of our description firstly, then they need to imagine a similar image in their mind. Finally, they need to choose the right image from the CAPTCHA. The whole process needs cognitive ability which machine does not have. First, the understanding of the description is the acquisition of knowledge. Second, imagining the images of the description is the manipulation of information. Third, the process of choosing the correct image is the reasoning part. So we can use this ability to distinguish from the people and machines. The machine can only extract the specific feature from the image, and it cannot imagine what they have not seen. However, humans can imagine what they have not seen from the semantic information, for example, people can imagine what a red bird looks like if they have seen a bird.

5.2 Security performance

For the text-based CAPTCHA, most attack methods are based on the segmentation and recognition method. Because our CAPTCHA needs the understanding of the description and selecting the corresponding image, which requires the cognition ability of humans, our CAPTCHA can avoid the attacks of traditional text-based CAPTCHA.

However, recent studies show that the proposed CAPTCHAs have challenges like image retrieval and image caption techniques [14,25]. Image retrieval is used to browse, search, and retrieve images from a group of images. Most traditional and common methods of image retrieval require labor work of preparing metadata such as captioning, keywords, or descriptions to the image. As our CAPTCHA presents that users need to select the corresponding image with the description sentence, the image retrieval seems to be the most direct attack method. But, it is still very hard for image retrieval to crack our CAPTCHA for three reasons. First, in the practical situation, the attackers can only get open datasets. These datasets are maybe similar to the training datasets we use, but not the same. So it is hard for them to get a same generator as ours. Second, if the crack team wants to train their model, they have to collect our generative CAPTCHA image as a training set. It may be threatening to our CAPTCHA when they collect a quantity of our CAPTCHA image, but it is very time consuming for them, which also decreases the possibility of being defeated. Last but not least, if the crack team does spend time collecting images and training their model to defeat our CAPTCHA, it is still very flexible for us to update our CAPTCHA. We can change our image training set and description training set and the generative CAPTCHA will be totally different from that before.

5.3 Limitation

There is still room for improvement in future work and we discuss some points here.

5.3.1 Image quality

The synthetic image result of our model is acceptable for the usage of CAPTCHA. But it still has the problem like low resolution, limited variety, and mode collapse, which means some generated samples contain the same texture pattern and shape. There are more and more new kinds of GAN to generate images with higher resolution, but for a CAPTCHA image, the higher resolution doesn't mean better. First, the resolution of our CAPTCHA is low, but it is acceptable for a human to recognize and understand. Second, in practice, the provider needs to save millions of CAPTCHA images, and low-resolution CAPTCHA images can save a lot of disk space. The mode collapse is a typical problem of Generative Adversarial Network. Usually, we want to use GAN to generate a wide variety of outputs, for example, a bird image for every random noise input to the generator. However, if the generator successfully produces a plausible output, the generator may only learn to produce that output. And the goal of the generator is always to find the most plausible output for the discriminator. If the generator produces the same output or a small set of outputs repeatedly, the discriminator should learn to reject the output. But if the discriminator gets stuck into a local minimum, then the generator will find the most plausible output for the discriminator. Because the generator always produces the same or similar outputs, the variety of images will be reduced. It means that the variety of our CAPTCHA will be lowered. Although there is no perfect solution for the mode collapse problem, we can use the source image as input to control the variety partly, which can lower the impact of the mode collapse, because we can use the real image to expand the variety of the images generated from the generator.

5.3.2 Adversarial samples

A recent study shows that adversarial samples can confuse the machine learning classifiers [30]. By inserting some special perturbation into the image, it can mislead the classifier, and in the meantime, the perturbation doesn't modify the contents of the image. But the perturbation is also tightly tied to the model of classifier and its parameters. In practice, the crack team do not release their CAPTCHA solver and a small change in CAPTCHA solver structure is also enough to invalid the adversarial samples. So the dependence on the information of the solver is a problem that adversarial samples need to solve.

5.3.3 Real and synthetic datasets

Our method provides a possible way to generate images about some birds which may not exist in the real world. In other words, we also create a synthetic dataset about the unreal objects. As for the synthetic dataset, it can reduce the dependence on the labeling of data and generate the data which is difficult or impossible to capture in the real world. It can also generate the data which happened infrequently in nature but essential for the model training. Besides, synthetic data can also reduce security problems and privacy issues. However, compared with the real datasets, the synthetic dataset also has its shortages. There is a difference in data distribution between synthetic data and real data, and it is difficult to assess the difference between synthetic data and real data. So the choice of real dataset and synthetic dataset should depend on the practical situation. When we are training models in the traditional or common problem, the real dataset is more representative. However, for some uncommon situations and lack of datasets, the synthetic dataset can provide us with more useful data for the training of models.

5.4 Practical application

The CAPTCHA we proposed can also be utilized in practical situations. In the productive environment, various aspects have to be considered, such as the performance, the security, the user experience. For the performance, a pre-trained model can generate a number of candidate images in a short time. For security, these images are all generated from the model and are unique in the world. So, it is impossible for others to collect these CAPTCHA images on the Internet. For the user experience, TIC we proposed uses the cognitive ability of humans which is natural. The authentication process is that humans read the text description and choose the corresponding picture. Compared with those CAPTCHA like slide CAPTCHA or puzzle CAPTCHA, TIC is more user-friendly. However, there are still many people who cannot read or cannot understand the description precisely. For those people, TIC may be hard for them to use, but we can use a simple description and simple image for those people to reduce the difficulty. Besides, we can also change the form of CAPTCHA. Instead of choosing the correct images matching the description, we can provide a correct image for the user and require the user to choose the other images which are similar to the image we provide.

6 Conclusion

In this work, we propose a new mechanism of CAPTCHA. The major challenges of image-based CAPTCHA and text-based CAPTCHA are resistance ability of attacks based on deep learning, limitation of labor works for the source image, and the lack of cognition process. In order to address such problems, we propose a novel model to generate a large number of semantic synthetic images that add cognition challenge and semantic reasoning process. Specifically, our model consists of three components, Semantic Image Generator, Image-based Semantic Image Generator, and Location-based Semantic Image Generator. These generators generate synthetic images and provide a progressive variety. Extensive experiments on the publicly available dataset and the attack of ResNet-50 demonstrate the quality of synthetic images and the resistance ability of the attack of the classification of Neural Network.

Acknowledgements This work is supported by Fundamental Research Funds for the Central Universities, Artificial Intelligence Research Foundation of Baidu Inc., Zhejiang University and Cybervein Joint Research Lab, Zhejiang Natural Science Foundation (LY19F020051, R19F020009, LZ17F020-001), National Natural Science Foundation of China (61572-431, U19B2042), Key R&D Program of Zhejiang Province (2018C01006), Program of China Knowledge Center for Engineering Sciences and Technology, Program of ZJU and Tongdun Joint Research Lab, Program of ZJU and Horizon Robotics Joint Research Lab, Joint Research Program of ZJU and Hikvision Research Institute, and Major Scientific Research Project of Zhejiang Lab (No. 2018EC0ZX01-1), CAS Earth Science Research Project(XDA19020104).

References

1. Aleksandrovich, P.N., Alekseevich, N.I., Vladimirovich, V.M., Igorevich, N.A., Borisovna, P.V., Igorevna, N.O.: Image-based captcha system. US Patent App. 13/528,373 (2012)
2. Bursztein, E., Martin, M., Mitchell, J.: Text-based captcha strengths and weaknesses. In: Proceedings of the 18th ACM conference on Computer and communications security, pp. 125–138. ACM (2011)
3. Chen, J., Luo, X., Guo, Y., Zhang, Y., Gong, D.: A survey on breaking technique of text-based captcha. Secur. Commun. Netw. **2017** (2017)
4. Cheng, Z., Gao, H., Liu, Z., Wu, H., Zi, Y., Pei, G.: Image-based captchas based on neural style transfer. IET Inf. Secur. **13**(6), 519–529 (2019)
5. Chew, M., Tygar, J.D.: Image recognition captchas. In: International Conference on Information Security, pp. 268–279. Springer (2004)
6. Datta, R., Li, J., Wang, J.Z.: Imagination: a robust image-based captcha generation system. In: Proceedings of the 13th Annual ACM International Conference on Multimedia, pp. 331–334. ACM (2005)
7. Dong, H., Yu, S., Wu, C., Guo, Y.: Semantic image synthesis via adversarial learning. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5706–5714 (2017)
8. Gao, H., Wang, W., Qi, J., Wang, X., Liu, X., Yan, J.: The robustness of hollow captchas. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, pp. 1075–1086. ACM (2013)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
10. 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)
11. Hwang, K.F., Huang, C.C., You, G.N.: A spelling based captcha system by using click. In: 2012 International Symposium on Biometrics and Security Technologies, pp. 1–8. IEEE (2012)
12. Ince, I.F., Yengin, I., Salman, Y.B., Cho, H.G., Yang, T.C.: Designing captcha algorithm: splitting and rotating the images against OCRS. In: Third International Conference on Convergence and Hybrid Information Technology, 2008. ICCIT'08, vol. 2, pp. 596–601. IEEE (2008)
13. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
14. Kiros, R., Salakhutdinov, R., Zemel, R.S.: Unifying visual-semantic embeddings with multimodal neural language models. arXiv preprint arXiv:1411.2539 (2014)
15. Kwon, H., Kim, Y., Yoon, H., Choi, D.: Captcha image generation systems using generative adversarial networks. IEICE Trans. Inf. Syst. **101**(2), 543–546 (2018)
16. Kwon, H., Yoon, H., Park, K.W.: Captcha image generation using style transfer learning in deep neural network. In: International Workshop on Information Security Applications, pp. 234–246. Springer (2019)
17. Kwon, H., Yoon, H., Park, K.W.: Robust captcha image generation enhanced with adversarial example methods. IEICE Trans. Inf. Syst. **103**(4), 879–882 (2020)
18. Lipton, Z.C., Tripathi, S.: Precise recovery of latent vectors from generative adversarial networks. arXiv preprint arXiv:1702.04782 (2017)
19. Liu, F., Li, Z., Li, X., Lv, T.: A text-based captcha cracking system with generative adversarial networks. In: 2018 IEEE International Symposium on Multimedia (ISM), pp. 192–193. https://doi.org/10.1109/ISM.2018.000-9 (2018)
20. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial autoencoders. arXiv preprint arXiv:1511.05644 (2015)
21. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
22. Mori, G., Malik, J.: Breaking a visual captcha. Unpublished manuscript (2002)

23. Park, H., Yoo, Y., Kwak, N.: MC-GAN: multi-conditional generative adversarial network for image synthesis. In: The British Machine Vision Conference (BMVC) (2018)
24. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks (2016)
25. Reed, S., Akata, Z., Lee, H., Schiele, B.: Learning deep representations of fine-grained visual descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 49–58 (2016)
26. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text-to-image synthesis. In: Proceedings of The 33rd International Conference on Machine Learning (2016)
27. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. In: Advances in Neural Information Processing Systems, pp. 2234–2242 (2016)
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
29. Sivakorn, S., Polakis, I., Keromytis, A.D.: I am robot: (deep) learning to break semantic image captchas. In: IEEE European Symposium on Security and Privacy (EuroS&P), pp. 388–403. IEEE (2016)
30. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
31. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
32. Von Ahn, L., Blum, M., Hopper, N.J., Langford, J.: Captcha: using hard AI problems for security. In: International Conference on the Theory and Applications of Cryptographic Techniques, pp. 294–311. Springer (2003)
33. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD birds-200-2011 dataset. Technical report (2011)
34. Wang, Y., Lu, M.: An optimized system to solve text-based captcha. arXiv preprint arXiv:1806.07202 (2018)
35. Ye, G., Tang, Z., Fang, D., Zhu, Z., Feng, Y., Xu, P., Chen, X., Wang, Z.: Yet another text captcha solver: a generative adversarial network based approach. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 332–348. ACM (2018)
36. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5907–5915 (2017)
37. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: Stackgan++: realistic image synthesis with stacked generative adversarial networks. IEEE Trans. Pattern Anal. Mach. Intell. **41**(8), 1947–1962 (2018)
38. Zhu, B.B., Yan, J., Li, Q., Yang, C., Liu, J., Xu, N., Yi, M., Cai, K.: Attacks and design of image recognition captchas. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, pp. 187–200. ACM (2010)
39. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2223–2232

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Xinkang Jia is currently a second-year master student at the College of Software Technology, Zhejiang University, under the supervision of Prof. Jun Xiao and Prof. Chao Wu. He received his B.S. degree in electronic and information engineering at the Zhejiang University of Technology in July 2018. He is interested in computer vision, deep learning, and federated learning. His major research interest is focusing on using the meta-learning to improve the personalization performance of federated learning.

Dr. Jun Xiao is currently a Full Professor with the College of Computer Science, Zhejiang University. He received the B.E. and Ph.D. degrees from the College of Computer Science, Zhejiang University, Hangzhou, China, in 2002 and 2007, respectively. His research interests include cross-media analysis, computer vision, and machine learning. He has served as reviewers for a rich set of forums, e.g., TMM, TPAMI, Neurocomputing, and Information Sciences.

Chao Wu is a Research Professor at School of Public Affairs, Zhejiang University. He is now the director of Computational Social Science Research Center (ZJUCCS, accessiable with ZJU VPN). He is also an Honorary Research Fellow of Data Science Institute and Department of Computing, Imperial College London, and the Director of Big Data Research Center –ZJPTCC. His main research interests are distributed machine learning, privacy protection, and computable social analytics. A list of selected publications is available here. He is also the creator of Mo, which is web-based platform for both AI modelling and education.