Improving Robustness and Visibility of Adversarial CAPTCHA Using Low-frequency Perturbation

Improving Robustness and Visibility of Adversarial CAPTCHA Using Low-frequency Perturbation

Takamichi Terada, Vo Ngoc Khoi Nguyen, Masakatsu Nishigaki and Tetsushi Ohki

Abstract CAPTCHA is a type of Turing test used to distinguish between humans and computing machine. However, image-based CAPTCHAs are losing their function as Turing tests owing to the improvement of image recognition using machine learning. This paper proposes an Adversarial CAPTCHA that provides attacking resistance to CAPTCHAs by using Adversarial Example (AE) as well as maintaining visibility by reducing image degradation. The proposed CAPTCHA maintains the difficulty of solving CAPTCHAs using computing machine by adding resistance against the attack using a machine learning classifiers. The proposed CAPTCHA is evaluated using three evaluation experiments, i.e., the attack using a machine learning classifier, the image quality, and the solving workload. The three evaluation experiments show that an Adversarial CAPTCHA is resistant to the attack by machine learning and is as convenient as the existing CAPTCHA.

1 Introduction

CAPTCHA is a fully automated Turing test that can distinguish between humans and computing machines[14]. Previous studies have applied CAPTCHAs to tasks that are easy for humans to identify but difficult for computing machines, such as complex image recognition tasks[5, 2]. However, along with recent advances in image recognition algorithms, such as deep neural network-based image recognition algorithms, computing machines have shown a high recognition accuracy for image-based CAPTCHAs[3, 13]. The fact that the recognition accuracy of image-based CAPTCHAs no longer differs between humans and computing machines indicates

Takamichi Terada · Vo Ngoc Khoi Nguyen · Masakatsu Nishigaki
Tetsushi Ohki
Shizuoka University, Graduate School of Integrated Science and Technology
3-5-1 Johoku, Naka-ku, Hamamatsu, Shizuoka, 432-8011, Japan.

Takamichi Terada, e-mail: terada@sec.inf.shizuoka.ac.jp

that CAPTCHAs are losing their ability to determine humans from computing machines.

We propose a novel CAPTCHA algorithm for generating CAPTCHA images that are difficult for machine learning classifier to solve but easy for humans uses. Osadchy et al. proposed DeepCAPTCHA as an attempt to create a CAPTCHA that is difficult to attack using a machine learning classifier[10]. With DeepCAPTCHA, an adversarial example (AE) is applied to make it difficult for machine learning classifiers to attack the CAPTCHA[4]. However, DeepCAPTCHA applies perturbations multiple times, and when the perturbations are applied at a level where the image is sufficiently resistant to machine learning classifiers, the image quality tends to degrade, which may make it difficult for humans to solve.

Considering the problems, we limit the number of perturbations based on a basic iterative method. In addition, we propose a low-frequency CAPTCHA image generation algorithm that limits the perturbations to low-frequency components. We evaluated the proposed CAPTCHA from three perspectives, i.e., the attack resistance by machine learning classifiers, visibility, and convenience, and demonstrated its effectiveness and reliability. We then show that the proposed algorithm is not only able to prevent a degradation of the the image quality but also generate CAPTCHA images that are robust against known AE removal attacks, such as median filters, with minimal degradation to human vision.

2 Related Works

2.1 Adversarial Example

AE is a method that can make a machine learning classifier misclassify by adding small perturbations to the input. Goodfellow et al. proposed the fast gradient sign method (FGSM) to efficiently generate AE[4]. In Eq. (1), The AE created by FGSM algorithm is defined as $\boldsymbol{X}^{adv}$, with the inputs $\boldsymbol{X}$ and their labels y, the model parameters $\boldsymbol{\theta}$, and the parameters indicating the number of updates ε.

$$\boldsymbol{X}^{adv} = \boldsymbol{X} + \varepsilon \ \mathrm{sign}\left(\nabla_{\boldsymbol{X}} J(\boldsymbol{\theta}, \boldsymbol{X}, y)\right) \tag{1}$$

To reduce the degradation of the image quality owing to perturbations, the basic iterative method (BIM), a method that iteratively applies FGSM by limiting the perturbations of the original image for each pixel, was proposed[9]. BIM[9] uses the Clip function defined as Eq. (2) to limit the perturbations added to the input image to within α of the original image.

$$\mathrm{Clip}_{\boldsymbol{X},\alpha}\left\{\boldsymbol{X}^{adv}\right\} = \min\left\{255, \boldsymbol{X} + \alpha, \max\left\{0, \boldsymbol{X} - \alpha, \boldsymbol{X}^{adv}\right\}\right\} \tag{2}$$

The update equation for AE in BIM is defined as Eq. (3):

$$\begin{aligned} \boldsymbol{X}_0^{adv} &= \boldsymbol{X}, \\ \boldsymbol{X}_{N+1}^{adv} &= \mathrm{Clip}_{\boldsymbol{X},\alpha}\{\boldsymbol{X}_N^{adv} + \varepsilon \ \mathrm{sign}(\nabla_X J(\boldsymbol{\theta}_N, \boldsymbol{X}_N^{adv}, y_{true}))\} \end{aligned} \tag{3}$$

where N is the number of updates, $\boldsymbol{X}_N^{adv}$ is the input $\boldsymbol{X}$ after the Nth update, and y_{true} is the correct label corresponding to the input $\boldsymbol{X}$. Using Eq. (3), we can restrict the range of possible values of the input $\boldsymbol{X}_{N+1}^{adv}$ after $N+1$ updates to within the range shown in Eq. (2).

2.2 Adversarial CAPTCHA

Adversarial CAPTCHA is a method that adds resistance against attacks that use machine learning classifiers to attack a CAPTCHA [14] by applying AE techniques to CAPTCHAs that are considered able to distinguish whether the operator is a human or a computing machine. DeepCAPTCHA, one of the Adversarial CAPTCHA methods proposed by Osadchy et al., uses an AE generation method that is resistant to machine learning classifiers as well ass perturbation removal through preprocessing using image processing filters[10]. Osadchy et al. labeled this method immutable adversarial noise (IAN), and we apply this name in the present paper as well. In this paper, we develop an AE based on the IAN algorithm, which is resistant to perturbation removal methods using median filters and can reduce the degradation in image quality caused by multiple perturbations and apply it to a CAPTCHA.

3 Adversarial CAPTCHA Using Low-frequency Perturbation

Our proposed method consists of the AE generation process using low-frequency perturbation and CAPTCHA challenge generation process. For clarity, we refer to the proposed AE method, the method that generates images for use in CAPTCHAs using AE, and proposed CAPTCHA, the whole system that presents the generated AE images. The AE generation process consists of BIM[9], a method that limits the amount of image modification through perturbations, and a low-frequency perturbation method that limits the perturbations to the low-frequency components of the image. The CAPTCHA challenge generation process presents the user with the challenge of selecting one image from multiple AE images. Here we describe these two processes.

3.1 Basic Iterative Method

To achieve a method with minimal degradation to human vision, we use BIM, shown in Eq. (3), for AE generation. By setting an upper bound on the amount of perturbations, BIM can set a lower bound of image quality. Controlled image quality makes it easier for human to solve CAPTCHAs.

3.2 Low-frequency Perturbation

Assuming that the attacker knows that the CAPTCHA image is composed of AE, the attacker may remove the perturbations from the CAPTCHA image by using a median filter before inputting the CAPTCHA to the machine learning classifier. Perturbations were added independently of the semantic information of the images. Therefore, the perturbations are distributed within the high-frequency region in the frequency domain and can be easily removed by applying median filters. We add perturbations only in the low-frequency such that the generated perturbations are distributed in the low-frequency region, making it difficult to be removed using a median filter. The low-frequency perturbation $\mathscr{F}(\boldsymbol{X}, c)$ for image X is defined as follows:

$$\mathscr{F}(\boldsymbol{X}, c) = \mathrm{IFFT}(\mathrm{LPF}(\mathrm{FFT}(\boldsymbol{X}), c)) \tag{4}$$

where the function FFT is a 2D Fourier transform, IFFT is a 2D inverse Fourier transform, and $\mathrm{LPF}(\boldsymbol{X}, c)$ is a Low-pass filter with c as the cutoff frequency within the 2D frequency domain. In the proposed AE method, $\mathscr{F}$ is applied to the image $\mathrm{Clip}_{\boldsymbol{X},\alpha}\{\boldsymbol{X}^{adv}\}$ in Eq. (2). In the following, we refer to perturbations that are limited to low frequencies after applying BIM as low-frequency perturbations.

The algorithm used for creating an AE with high visibility and resistance to machine learning attacks using low-frequency perturbations is shown in Algorithm 1. Here, δ_α is a parameter that indicates the amount of update of α. In this paper, we assumed a median filter as a perturbation removal method. We assumed a median filter as a perturbation removal method as it has been validated as the most efficient perturbation removal method among various image processing filters, as described in the Osadchy's work[10].

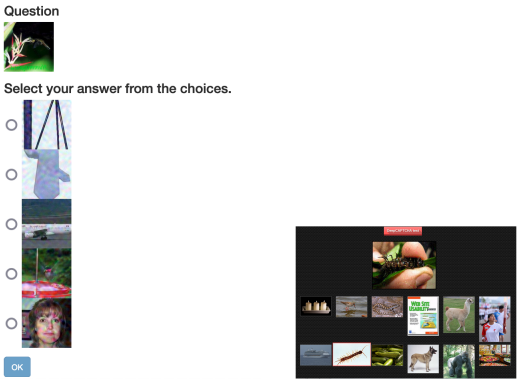
3.3 CAPTCHA Challenge Generation

Fig. 1(a) shows an example of the operation screen of the proposed CAPTCHA. The proposed CAPTCHA is similar to the DeepCAPTCHA format shown in Fig. 1(b). Since the DeepCAPTCHA shows a single AE image as the CAPTCHA challenge,

Algorithm 1 Generation of low-frequency perturbations

Require: Let Net be a deep learning network; I be the original image; C_i be the true class, $C_d$be the predicted class; $confidence$ be the confidence value of the network; p be the threshold for the confidence value; M_f be the median filter; $\mathscr{F}$ be the Fourier transform; $\mathscr{C}$ be the cutoff value; L_i be the upper limit of the number of image updates and L_α be the upper limit of the number of perturbation updates.

$adv(I, C_d, p) \leftarrow I$; ▷ adv denotes an AE generation function
$\eta \leftarrow 0$;
Update the perturbation at most m times until it can change the true label to a different label:
while ($Net(M_f(adv(I, C_d, p)))) = C_i \vee (m < L_\alpha)$ **do**
    Update the perturbation at most n times using a low-frequency perturbation:
    while $(Net(adv(I, C_d, p))) \neq C_d \vee (confidence < p) \vee (n < L_i)$ **do**
        $\eta \leftarrow$ *run BIM with noise magnitude* α;
        $\tilde{\eta} \leftarrow \mathscr{F}^{-1}(\mathscr{C}(\mathscr{F}(\eta)))$;
        $adv(I, C_d, p) \leftarrow adv(I, C_d, p) + \tilde{\eta}$;
    end while
    Update perturbation by adding a small value:
    $\alpha = \alpha + \delta_\alpha$;
end while
Output: η



(a) Proposed CAPTCHA (b) DeepCAPTCHA[10]

Fig. 1 Example of (a) the proposed CAPTCHA and (b) DeepCAPTCHA system.

our system allows the user to choose one image from a set of source images with the same label as the challenge. As shown in Fig. 1, the difference between proposed CAPTCHA and DeepCAPTCHA is the number of A.E. images included in each CAPTCHA challenge. By using multiple images, we aim to improve the machine learning attack resistance of the CAPTCHA system by increasing the number of AEs used in the single CAPTCHA challenge. The images used in the proposed CAPTCHA are Caltech-256[6], which is a publicly available dataset. Here, images used for options are AEs created using Algorithm 1.

4 Evaluation

4.1 Preliminary

In our experiment, we trained the CAPTCHA generation model using the MNIST and Caltech-256 datasets[1, 6]. For the MNIST dataset, we used a multi-layer perceptron consisting of three fully connected layers with 300, 100 and 10 nodes for the network model. For the Caltech-256 dataset, we used VGG-Net[12]. For the scenario using MNIST with a multi layer perceptron, the number of times the median filter applied t was set to 6.The kernel size of the median filter k was set to 5, and the cutoff size of the low-pass filter c was set to 9. For the scenario using Caltech-256 with VGG-Net, the number of times the median filter applied t was set to 1. The kernel size of the median filter k was set to 7, and the cutoff size of the low-pass filter c was set to 102. We defined cut-off values from preliminary experiment. Note that we applied the low-pass filter only to the proposed AE method for each scenario. We set the threshold p of the confidence value to 0.8. If the confidence value of AE exceeds this threshold, the AE creation process ends. The number of times the median filter applied, the kernel size, and the confidence value were the same as those used by DeepCAPTCHA[10]. In the following part, we will evaluate the proposed CAPTCHA through three evaluations. Among the three evaluations, an attack resistance evaluation using a machine learning classifier is described in Sect. 4.2, an image quality evaluation through a subjective evaluation is detailed in Sect. 4.3.1, and a CAPTCHA workload evaluation is described in Sect. 4.4. The results of the experiments are presented in each section along with the experiment methodology.

4.2 Attack Resistance Evaluation

4.2.1 Procedure

The attack resistance of each CAPTCHA system was evaluated by calculating and comparing the attack resistance retention rate from the results of the classification by the machine learning classifier. We defined the attack resistance retention rate as the ratio of inputs that an attacker cannot solve to the total number of inputs. Here, we assume that the attacker can apply a median filter with parameters k and t to the challenge image before inputting it to the machine learning classifier.

Eq. (5) shows the attack resistance retention rate P_d using the total number of inputs N, the number of creation failures n_{mf}, and the number of countermeasure failures n_{sf}. Here, n_{mf} is defined as the number of inputs images for which the CAPTCHA generator failed to create an Adversarial CAPTCHA. n_{sf} is defined as the number of generated Adversarial CAPTCHAs that adversarial perturbation can be removed.

$$P_d = \frac{N - (n_{mf} + n_{sf})}{N} \tag{5}$$

Method	Dataset	Model	Resistance Retention Rate
IAN	MNIST	MLP	79.72%
Proposed AE Method	MNIST	MLP	81.29%
IAN	Caltech-256	VGGNet	81.62%
Proposed AE Method	Caltech-256	VGGNet	82.10%

Table 1 Comparison of resistance retention rate varying AE generation methods and attack target models. In each experiment, we used all 10,000 images in the dataset.

The resistance against solving by machine learning classifiers is evaluated base on the attack resistance retention rate P_d.

4.2.2 Result

Table 1 shows the attack resistance retention rate of IAN and the proposed AE method. From Table 1, we can see that the proposed AE method has a higher resistance retention rate than IAN.

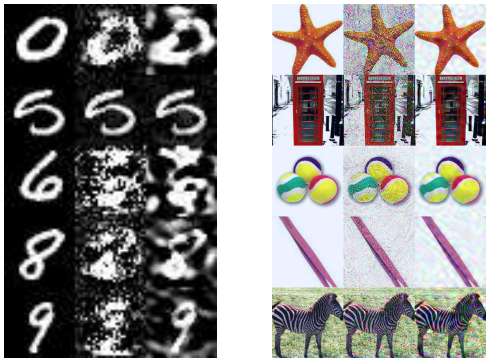
4.3 Image Quality Evaluation

4.3.1 Procedure

An image quality evaluation experiment was conducted using a double stimulus continuous quality scale[8]. The dual stimulus method is a subjective quality evaluation method for images and videos defined in ITU-R BT.500-14. In our experiment, 16 university students majoring in computer science between the ages of 21 and 24 were asked to participate in the experiment.

Prior to the experiment, we provided informed consent to all participants and conducted a practice session to confirm the experimental procedure. Note that we used CAPTCHA images that were completely unrelated to the experiment in a practice. In our experiment, we combined two types of attack methods (IAN and the proposed AE method) and two types of datasets (MNIST and Caltech-256) with four scenarios.

We prepared 10 CAPTCHA challenge pairs for each of the four scenarios and presented them to the participants. Each pair consists of an unperturbed original image and a perturbed AE image correspond to the scenario. The participants scored the image quality of the original and AE images on a 100-point scale for each scenario. The participants are free to switch between the original and AE images as a pair during the evaluation. We show some examples of AE produced by IAN and the proposed AE method in Fig. 2. The original image, corresponding output of the IAN, and that of the proposed AE method are arranged from left to right. We pre-



(a) MNIST (b) Caltech-256

Fig. 2 Example of the output result of AE using (a) MNIST and (b) Caltech-256. The original image, corresponding output of the IAN, and that of the proposed AE method are arranged from left to right.It can be seen that the AE creation the proposed AE method suppresses the degradation of the image quality in some images. In the image quality evaluation, a set of the original image and one of the two types of AE was used.

sented the original image and one of the two types of AE to the participants in pairs in the image quality evaluation.

We show the average difference score between the original and AE images with a 95% confidence interval for each scenario. Outliers were detected according to the method applied in ITU-R BT.500-14 Annex.1.

4.3.2 Result

Fig. 3 is the result of the image quality evaluation. Each bar represents the mean of difference in image quality evaluation score between the original image and perturbed image in each experimental condition. A small value of the mean of difference indicates that the quality degradation from the original image is small. Note that the error bars show the 95% confidence interval. As shown in Fig. 3, although the difference is insignificant, the mean of difference of MNIST in the proposed AE method is smaller than that of existing method. In addition, the mean of difference in Caltech-256 is smaller in the proposed AE method than in the existing methods. Additionally, we conducted outlier detection following the method specified in the ITU-R document and confirmed that there were no participants whose answers corresponded to the outlier.

4.4 CAPTCHA Workload Evaluation

4.4.1 Procedure

We compared the convenience of the proposed CAPTCHA to existing CAPTCHAs by evaluating the workload using NASA-TLX[7]. An evaluation of NASA-TLX

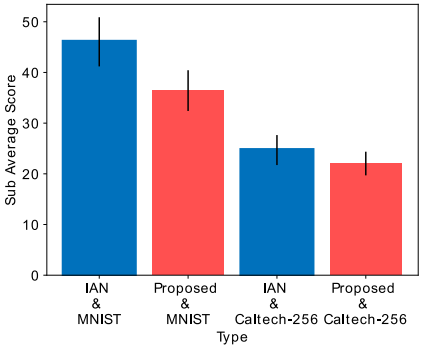


Fig. 3 Results of image quality evaluation. Each error bar is 95 % confidence interval. The scores using MNIST are IAN (46.0) and the proposed AE method (36.4). In addition, the scores using Caltech-256 are IAN(24.7) and the proposed AE method (22.0). A small value of the mean of differences indicates that the quality degradation from the original image is small.

used in this paper was conducted in Japan by evaluating the axes of the Japanese version of NASA-TLX developed by Haga et al.[7, 11]. The CAPTCHAs to be compared were GIMPY and reCAPTCHA, which are used in many different websites[15, 5]. We recruited 250 participants for this survey using Lancers.jp[1]. The participants were asked to go to the web page of the survey from the URL provided in the work request form on Lancers.jp, and solve the three types of challenges, i.e., the proposed CAPTCHA, reCAPTCHA, and GIMPY, and then evaluate the workload using NASA-TLX for each of them.

Before conducting the survey, the participants were given an explanation regarding the survey, and were then given a practice session to evaluate the results after they were sufficiently familiar with the operation. Fig. 1(a) showed example of challenge images that were used in the practice session. We selected all challenge images from Caltech-256 with 224×224 image size. The survey was conducted by solving the CAPTCHA for a specified number of times. We evaluated the workload using NASA-TLX for each of the question types GIMPY, reCAPTHCA, and the proposed CAPTCHA.

4.4.2 Result

First, the weighted average of NASA-TLX for each CAPTCHA method is shown in Fig. 4. The scores are GIMPY (54.0), the proposed CAPTCHA (40.7), and reCAPTCHA (39.8) in ascending order.

We also tested The NASA-TLX weighted means for each of these three forms using the Shapiro-Wilk test. On running the Shapiro-Wilk test, we set the null hypothesis as the NASA-TLX weighted mean for each participant in each CAPTHCA method follows a normal distribution, with 188 degrees of freedom and a significance level of 5%. Table 2 shows the results of the Shapiro-Wilk test. Table 2 shows that the null hypothesis cannot be rejected only for GIMPY, whereas the null hypoth-

[1] URL: https://www.lancers.jp/

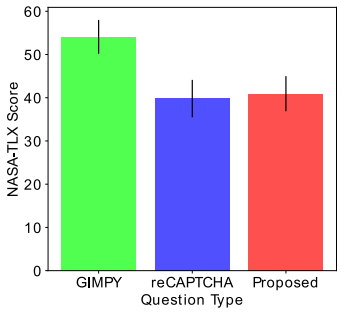


Fig. 4 Results of workload survey. Each error bar is standard deviation. The scores are GIMPY (54.0), the proposed CAPTCHA (40.7), and reCAPTCHA (39.8) in ascending order.

Table 2 Results of the Shapiro-Wilk test for each question type (note, 188 degrees of freedom and a significance level of 5% were applied). The null hypothesis cannot be rejected only for GIMPY.

Question Type	p value
GIMPY	0.480
reCAPTCHA	$<$ **0.001****
Proposed CAPTCHA	**0.030***

*:p$<$ 0.05, **:p$<$ 0.001

Table 3 The results of the Friedman test(note: 188 degrees of freedom and a significance level of 5% were used). It can be confirmed that the null hypothesis is rejected.

Question Type	p value
Proposed CAPTCHA, reCAPTCHA, GIMPY	$<$ **0.001****

**:p$<$ 0.001

esis can be rejected for the other forms. These results indicate that the NASA-TLX weighted mean of GIMPY is not necessarily non-parametric.

Because the NASA-TLX weighted mean values other than GIMPY are non-parametric, we conducted a Friedman test. As the null hypothesis, "There is no difference in the NASA-TLX weighted mean values for each participant in each question format" with 188 degrees of freedom and a significance level of 5%. Table 3 shows the results of the Friedman test. Because the null hypothesis is rejected, we can see that there is a difference in the NASA-TLX weighted mean values among the three problem formats.

The NASA-TLX weighted mean values were then tested using Wilcoxon's signed rank test. As the null hypothesis, "There is no difference between the NASA-TLX weighted averages of the targets", and the following three targets were applied: GIMPY and CAPTCHA of the proposed CAPTCHA, reCAPTCHA and CAPTCHA of the proposed CAPTCHA, and reCAPTCHA and GIMPY. In addition, the significance level was 5%. The results of the Wilcoxon's signed rank test are shown in Table 4.

The effect size d showed in Table 4 was calculated using Cohen's effect size d. It can be seen that at a significance level of 1.6%, corrected for the Bonferroni method, the proposed CAPTCHA is significantly smaller than that of GIMPY, but not significantly smaller than that of reCAPTCHA.

Table 4 Results of Wilcoxon's signed rank test (significance level of 5% and 1.6% significance level when corrected using the Bonferroni method). It can be seen that the null hypothesis of the combination of reCAPTCHA and GIMPY and the combination of CAPTCHA used by the proposed CAPTCHA and GIMPY is rejected.

Question Type 1	Question Type 2	Effect Size*d*	p value（Question Type 1 <Question Type 2）
reCAPTCHA	GIMPY	0.834	$\mathbf{< 0.001}^{**}$
Proposed Method	GIMPY	0.824	$\mathbf{< 0.001}^{**}$
Proposed Method	reCAPTCHA	0.064	0.766

**:p< 0.001

5 Discussion

In the workload evaluation described in Sect. 4.4, we compared the convenience of the proposed CAPTCHA to other widespread CAPTCHA systems. As we can see from Table 4, Wilcoxon's signed-rank test rejects the null hypothesis in the evaluation using GIMPY and the proposed CAPTCHA but cannot reject the null hypothesis in the evaluation using reCAPTCHA and the proposed CAPTCHA. These results show that the proposed method is more convenient than GIMPY and as convenient as reCAPTCHA while maintaining the resistance to machine learning attacks.

One of the reasons why the participants evaluated reCAPTCHA as so convenient is the inconsistency of the reCAPTCHA procedure. The reCAPTCHA version we used is v2. reCAPTCHA v2 is a method that discriminates between humans and computing machines by selecting objects such as cars and traffic lights scattered in the image only when there is a possibility of access by a bot. In our experiment, we expected that the task of selecting cars or traffic lights would occur at least once in 10 trials, but this did not happen for a small number of participants. Hence, in the open-ended section of the post-survey questionnaire, some participants mentioned that the workload test for the reCAPTCHA was completed only by clicking on the check-boxes. Therefore, it is important to consider that the convenience of reCAPTCHA is highly variable than that of proposed CAPTCHA.

As for the limitations, our proposed CAPTCHA requires users to browse images. In other words, users who cannot see the images, such as the visually impaired users, cannot use the system. As a future challenge, we will extend the CAPTCHA format to include audio and other modalities in addition to images.

6 Conclusion

This paper proposed an Adversarial CAPTCHA with high visibility using low-frequency perturbations. Our proposed method makes it difficult to solve the CAPTCHA by machine learning classifiers and, at the same time, maintains visibility by reducing image degradation.

The proposed CAPTCHA was tested separately from three perspectives: attack resistance against machine learning classifiers, visibility, and usability. We hope that the use and application of the CAPTCHA method proposed in this paper will not only allow the further development of CAPTCHAs as proof of human work, but also contribute to the development of machine learning and various other fields.

References

1. Deng, L.: The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. IEEE Signal Processing Magazine **29**, 141–142 (2012)
2. Elson, J., Douceur, J.J., Howell, J., Saul, J.: Asirra: A CAPTCHA that Exploits Interest-Aligned Manual Image Categorization. In: Proceedings of 14th ACM Conference on Computer and Communications Security, pp. 366—-374 (2007)
3. Golle, P.: Machine Learning Attacks against the Asirra CAPTCHA. In: Proceedings of the 15th ACM Conference on Computer and Communications Security, pp. 535–542 (2008)
4. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples. In: International Conference on Learning Representations (2015)
5. Google: Choosing the type of reCAPTCHA — Google Developers. https://developers.google.com/recaptcha/docs/versions (2021). [Online] (Last accessed : January-27-2022)
6. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report 7694 (2007)
7. Hart, S.G., Staveland, L.E.: Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In: Advances in psychology, vol. 52, pp. 139–183. Elsevier (1988)
8. ITU-R: Recommandation BT.500-14. Methodology for the subjective assessment of the quality of television pictures (2019)
9. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. International Conference on Learning Representations Workshop (2017)
10. Osadchy, M., Hernandez-Castro, J., Gibson, S., Dunkelman, O., Pérez-Cabo, D.: No Bot Expects the DeepCAPTCHA! Introducing Immutable Adversarial Examples, With Applications to CAPTCHA Generation. IEEE Transactions on Information Forensics and Security **12**, 2640–2653 (2017)
11. Shigeru, H., Naoki, M.: Japanese version of NASA Task Load Index: Sensitivity of its workload score to difficulty of three different laboratory tasks. The Japanese Journal of Ergonomics **32**, 71–79 (1996)
12. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. In: International Conference on Learning Representations (2015)
13. 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, pp. 388–403 (2016)
14. 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 (2003)
15. Von Ahn, L., Blum, M., Langford, J.: Telling Humans and Computers Apart Automatically. Communications of the Association for Computing Machinery **47**, 56–60 (2004)