

Received 19 June 2023, accepted 7 July 2023, date of publication 24 July 2023, date of current version 11 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3298442

RESEARCH ARTICLE

Augment CAPTCHA Security Using Adversarial Examples With Neural Style Transfer

NGHIA DINH¹, KIET TRAN-TRUNG², AND VINH TRUONG HOANG²

¹Faculty of Electrical Engineering and Computer Science, VSB—Technical University of Ostrava, 708-33 Ostrava-Poruba, Czech Republic

²Faculty of Computer Science, Ho Chi Minh City Open University, Ho Chi Minh 722000, Vietnam

Corresponding author: Vinh Truong Hoang (vinh.th@ou.edu.vn)

This work was supported in part by the VSB—Technical University of Ostrava, Czech Republic; and in part by Ho Chi Minh City Open University, Vietnam under Grant B2021-MBS-07.

ABSTRACT To counteract rising bots, many CAPTCHAs (Completely Automated Public Turing tests to tell Computers and Humans Apart) have been developed throughout the years. Automated attacks, however, employing powerful deep learning techniques, have had high success rates over common CAPTCHAs, including image-based and text-based CAPTCHAs. Optimistically, introducing imperceptible noise, Adversarial Examples have lately been shown to particularly impact DNN (Deep Neural Network) networks. The authors improved the CAPTCHA security architecture by increasing the resilience of Adversarial Examples when combined with Neural Style Transfer. The findings demonstrated that the proposed approach considerably improves the security of ordinary CAPTCHAs.

INDEX TERMS Machine learning, CNN, DNN, CAPTCHA, security, adversarial examples, cognitive.

I. INTRODUCTION

With a wide variety of applications, CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) or HIP (Human Interactive Proof) is a popular security defense tool to protect websites and other applications. Generally speaking, CAPTCHA is based on a hardness assumption of an AI problem. As a result, as long as a CAPTCHA is not broken, there is a way to differentiate humans from computers. Recently, automated attacks, however, employing powerful deep learning techniques, have had high success rates over common CAPTCHAs. Text-based CAPTCHAs have been found to be less secure against enhanced algorithms, powerful deep-learning techniques, and superior hardware. As a result, many designs prioritize image recognition over text recognition. However, object detection and image classification with deep learning techniques can bypass image-based CAPTCHAs. Some researchers recently discovered that Adversarial Examples [2] can fool state-of-the-art DNNs by adding imperceptible small perturbations in an original image. Furthermore, Neural Style Transfer [3]

performs an image's pixel-level updates by fusing with another style image to increase the difficulty of machine classification.

In this study, we proposed a novel way to strengthen the robustness of Adversarial Examples by combining this technique with Neural Style Transfer in order to improve the security of regular CAPTCHAs. According to our findings, integrating Adversarial Examples with Neural Style Transfer significantly improves the security of typical CAPTCHAs. Extensive security tests are performed against the attacking methods of Gradient Masking, Adversarial Training, and Input Transformation to determine the usefulness of this strategy in improving CAPTCHA security. The findings show that the proposed solution improves the security of common CAPTCHAs substantially. The findings also demonstrate that deep learning can improve CAPTCHA security and provide a promising direction for future CAPTCHA research.

The rest of the paper is structured as follows. Section II summarizes related works. Our proposed method is introduced in Section III. Section IV details and evaluates the security enhancements of this approach. Section V suggests research directions and concludes the work.

The associate editor coordinating the review of this manuscript and approving it for publication was Tyson Brooks¹.

II. RELATED WORKS

Text-based CAPTCHA has been the most extensively used CAPTCHA method. To improve security, this CAPTCHA technique included numerous resistance tactics such as crowding characters together (CCT), noise, backdrops, etc. Yet, as the research [1] has shown, all these resistance mechanisms appear to be ineffectual. Deep learning techniques have lately been used as a new approach to overcoming text-based CAPTCHA. reCAPTCHA was resolved with a high success rate using CNNs. The CNN-based approach was used to successfully defeat Microsoft's CAPTCHA. These successful assaults show that text-based CAPTCHAs are no longer extremely secure and that the age of text-based CAPTCHAs has passed. As a result, image-based CAPTCHAs have appeared since then. In line with this, the most common and effective approach for breaking image-based CAPTCHAs has been training neural networks to classify these images. ASIRRA CAPTCHA, Avatar CAPTCHA, Google CAPTCHA, FR-CAPTCHA, IMAGINATION, ARTiFACIAL CAPTCHA, etc. were bypassed by utilizing CNNs with high success rates. In general, image-based CAPTCHAs still have security flaws that make them vulnerable to automated attacks.

The new cognitive CAPTCHA [4] was proposed that combines text-based, image-based, and cognitive CAPTCHA characteristics with deep learning techniques like Adversarial Examples. The authors asserted that their suggested CAPTCHA is more secure than conventional CAPTCHAs while still being user-friendly. Zhang et al. [5] investigated the influence of Adversarial Examples on CAPTCHA security (including image and text-based CAPTCHAs) against DNN attacks. They demonstrated that Adversarial Examples have a considerable impact on the resilience of CAPTCHA security by using the existing generation techniques of Adversarial Examples to construct adversarial CAPTCHAs. Gajani et al. [6] suggested a method of generating secure CAPTCHAs using Neural Style Transfer (NST) and VGG-16. The proposed method employs a 9-cell-grid design with random stylized photos using Neural Style Transfer, and the user is requested to pick images that are aesthetically comparable to the distinct images shown. Osadchy et al. [7] proposed DeepCAPTCHA, an image-based CAPTCHA, that is designed to be resistant to ML (Machine Learning) attacks. In this approach, Immutable Adversarial Noise (IAN) is added to the original images whose generated image can deceive DNN networks and cannot be removed using image filtering. Obviously, DeepCAPTCHA is a new image-based CAPTCHA providing high security. Ye et al. [8] proposed a GAN-based approach to bypass text-based CAPTCHAs. In particular, synthetic CAPTCHAs are generated to build a base solver. The base solver is improved via transfer learning on actual CAPTCHAs. Their approach could successfully recognize some real CAPTCHAs, according to the evaluation. However, the solver is still a CNN network, which is impacted by Adversarial Examples, it cannot successfully overcome

adversarial CAPTCHAs. In other words, approaches of Adversarial Examples have clearly proved their efficacy and resilience when employed against machine learning attacks.

As a result of extensive research on Adversarial Examples, numerous new strategies for generating Adversarial Examples [2] have been proposed, including Fast Gradient Sign Method (FGSM), DeepFool, Basic Iterative Method (BIM), Jacobian-based Saliency Map Algorithm (JSMA), Adversarial Transformation Networks (ATNs), and others. In addition, several researchers [2] were exploring attacking strategies against Adversarial Examples. However, as shown in the research [2], attacking models are still unable to totally eradicate their influence. The approach of random scaling [2] was proved to reduce the impact of Adversarial Examples. Also, Adversarial Examples was claimed that it cannot always guarantee that they would fool the neural network for huge photos acquired from varied distances and perspectives. Nevertheless, OpenAI immediately said on their official blog that they generated photos that, when viewed from different angles and distances, can reliably fool automated bots' categorization. In the near future, research on Adversarial Examples in neural networks will remain a hot area.

In this paper, we present a new method for improving traditional CAPTCHA security by combining Adversarial Examples with Neural Style Transfer on image and text-based CAPTCHAs. To create the output image with high complexity, the technique blends Neural Style Transfer [3] with Adversarial Examples. Neural Style Transfer enhances the complexity by updating an image's pixel level when fusing with any style image. We can build styled images with secure capability against machine learning automated bots depending on the complexity of style images and Neural Style Transfer techniques. Adversarial Examples then will add more extra small unnoticeable noise to the styled image to augment the difficulty of image classification and recognition. This method is more adaptable since it allows us to combine any existing Adversarial Example model with any Neural Style Transfer methodology to generate highly secure traditional CAPTCHAs.

III. PROPOSED METHOD

We suggested a novel method that augments the security of classic CAPTCHA, particularly against machine learning automated assaults, by using existing resources and methodologies.

A. DATASETS

For text, we utilize EMNIST (expanded Modified National Institute of Standards and Technology) dataset [9]. And we utilize ILSVRC-2012 [10] dataset for images, which was employed in the ImageNet visual recognition competition in 2012.

B. APPLIED TECHNIQUES

1) NEURAL STYLE TRANSFER

Neural Style Transfer [3], [11], [12], [14] is a deep learning approach for making aesthetically attractive images. This

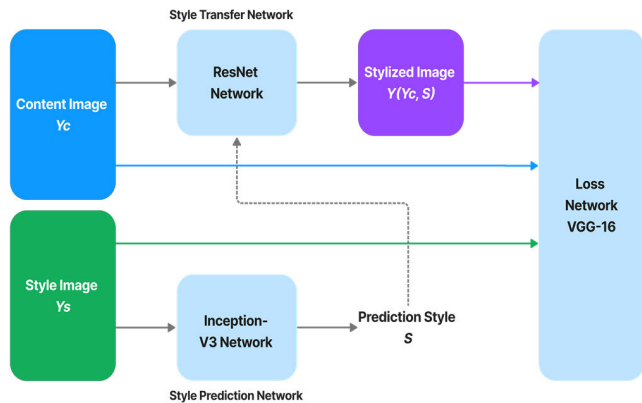


FIGURE 1. Network architecture.

approach applies the style of creative images to other images by using a trained DNN. The fast Style Transfer Network [12] is used in this study to enable for real-time stylization. As depicted in Figure 1, our network model is made up of three principal components: a style prediction network, a style transfer network, and a loss network. The content and style loss network employs a VGG classification network, typically VGG-16 [13]. To anticipate an input style image’s embedding vector S , the style prediction network follows the Inception-v3 design [15] and provides normalization constants to the style transfer network. The style transfer network is primarily as follows [16] in which fractionally strided and strided convolutions are employed for upsampling and downsampling instead of pooling layers.

The main purpose [12] is to minimize the total loss L_{total} for the improvement of the output image’s content and texture:

$$L_{total} = \alpha L_{content} + \beta L_{style} \quad (1)$$

where $L_{content}$ is the loss of content, L_{style} is the loss of style, and α and β are weighting factors that trade-off between style and content. To indicate if two images are similar in content, their extracted high-level features [12] are close in Euclidean distance. As a result, $L_{content}$ is the Euclidean difference between the content and output images:

$$L_{content}(y, y_c) = \sum_{l=1}^C \frac{1}{N_l} \|F^l - P^l\|_2^2 \quad (2)$$

To indicate if two images are similar in style, their extracted low-level features [12] share the same spatial statistics. As a result, L_{style} is the Gram difference between the generated image and style image, calculated as:

$$L_{style}(y, y_s) = \sum_{l=1}^L \frac{1}{N_l} \|G^l - A^l\|_F^2 \quad (3)$$

with G^l and A^l being Gram matrices of generated and style images at the layer l . The feature maps of y , y_c , and y_s are represented by F^l , P^l and S^l at layer l , respectively. C and L are the size of content and style layers, respectively. Besides, N^l is the filter number of the layer l .



FIGURE 2. A demonstration of Adversarial Example generation on ImageNet with $\epsilon = 0.007$.

TABLE 1. Image-based evaluation dataset.

| Dataset | Quantity |
|--|----------|
| Test Dataset - Normal | 1000 |
| Test Dataset - Stylized version (applied Neural Style Transfer) | 1000 |
| Test Dataset - Adversarial version (applied Adversarial Examples) | 1000 |
| Test Dataset - Stylized Adversarial version (applied Adversarial Examples and Neural Style Transfer) | 1000 |
| Training Dataset | 10000 |

2) ADVERSARIAL EXAMPLES

As shown in [2], Biggio et al. and Szegedy et al. first developed the concept of Adversarial Examples. DNN networks have been found to be sensitive to small imperceptible updates, Adversarial Examples, in the original images. Moreover, this technique impacts not just one model but also another with different architectures or training sets, as long as they are trained for the same job. Many new approaches for generating Adversarial Examples [2] have been proposed because of extensive research on Adversarial Examples, including the Fast Gradient Sign Method (FGSM), DeepFool, Basic Iterative Method (BIM), Jacobian-based Saliency Map Algorithm (JSMA), Adversarial Transformation Networks (ATNs), and others. In this work, FGSM (Fast Gradient Sign Method) [17] is employed for misclassification to a non-targeted class with the greatest benefits of speed and ease.

α : GENERATION METHOD

FGSM [17] is the quickest and simplest method for generating Adversarial Examples using neural network gradients. The approach generates a new image called the adversarial image by maximizing the input image’s loss. This can be expressed as follows:

$$x' = x + \epsilon \times \text{sign}(\delta_x J(\theta, x, y)) \quad (4)$$

where y is the original input label, x' is an adversarial image, x is the original input image, ϵ guarantees small perturbations, θ is the set of parameters for a training model, $\text{sign}()$ is to get the sign of the gradient direction, and J is the loss function. Maximizing the function J enables minimizing the difference between the output and the original input image

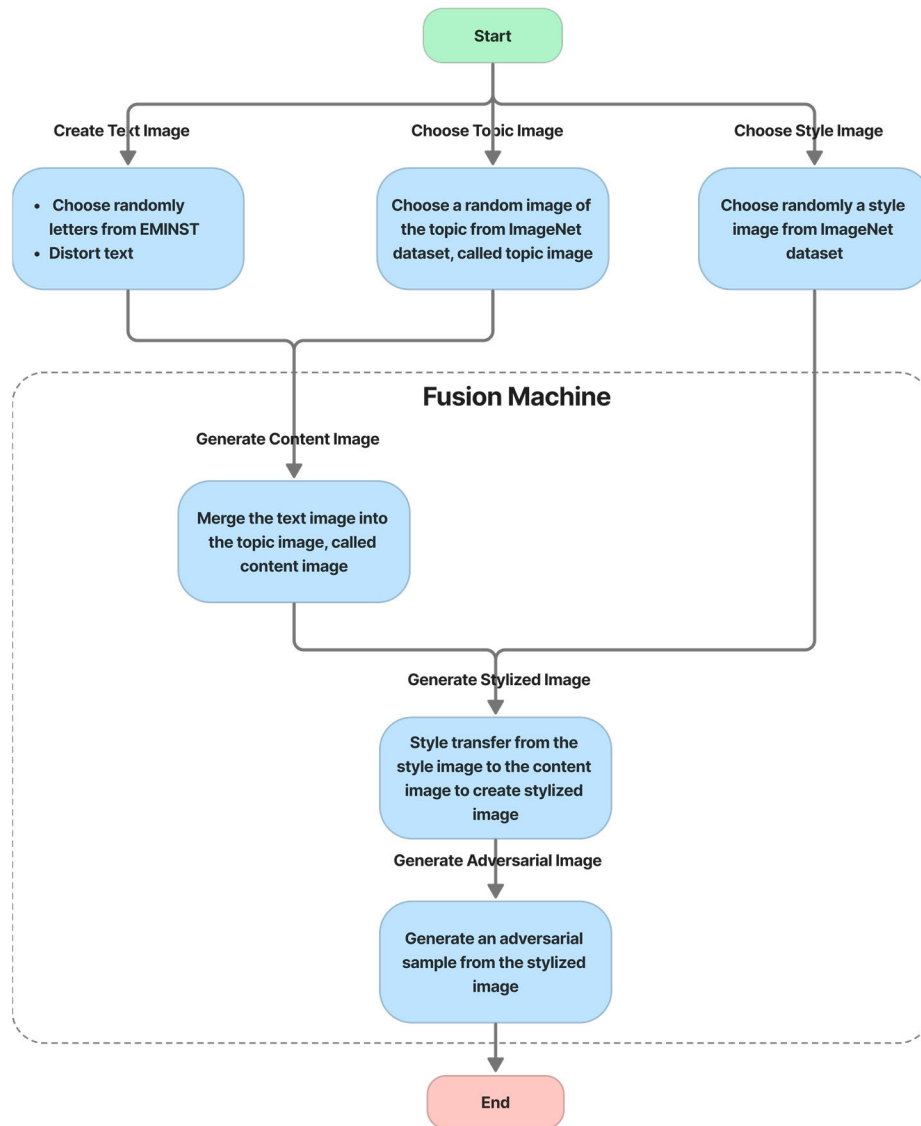


FIGURE 3. Proposed method.

when raising the likelihood of an output image being incorrectly classified.

b: GENERATION NETWORK







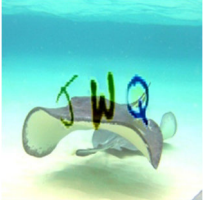
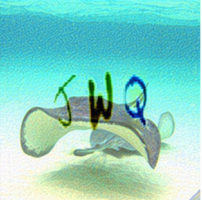
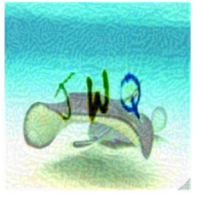
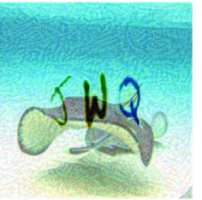



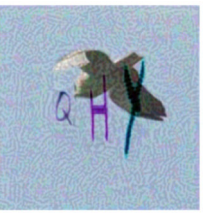
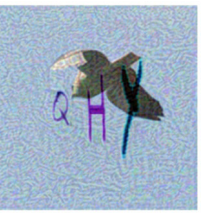
In our work, VGG-16 [13] and ResNet-101 [18] are applied to generate Adversarial Examples. VGG-16 is a CNN (Convolutional Neural Network) with 16 layers that is widely regarded as one of the best computer vision models available today. At the 2014 ILSVRC (ImageNet Large Scale Visual Recognition Challenge) competition, this model finished first and second. In 2015, He Kaiming et al developed ResNet, which stands for Residual Network, as a kind of CNN. ResNet-101 is a 101-layer Convolution Neural Network created from residual blocks and CNNs. At the ILSVRC 2015, this network took first place in the classification task.

C. IMPLEMENTATION

As shown in Figure 3, the CAPTCHA creation is carried out with the following additional details:

- The phase of text image generation selects random characters from EMNIST, distorts and controls the overlap among characters, then merges them with a white background.
- In ImageNet, the images are classified as topic folders, each folder containing all images belonging to its topic. The phase of topic image selection selects a random image of a topic from ImageNet to merge with the text image. The topic can be random or selected by a user.
- The phase of style image selection selects a random style image from ImageNet.
- The fusion phase merges the topic image and the text image to generate the content image using seamless

TABLE 2. Real evaluation image samples.

| Style | Normal | Stylized | Adversarial | Stylized Adversarial |
|--|--|--|---|--|
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

cloning [19], then applies Neural Style Transfer in which the proportion α/β in Formula (1) is 1.5 to ensure details and user experience more than style, and Adversarial Examples where ϵ in Formula (4) is 0.1, to generate the output.

IV. ATTACK EVALUATION

A. EXPERIMENTAL SETUP

We ran the experiments on a workstation with the equipment of an Intel Xeon(R) CPU 2.30GHz, NVIDIA TESLA T4 GPU, and RAM 16 GB. For deep learning frameworks, TensorFlow [20], and Torch [21] were employed, and the dataset we used is from Section III-A.

B. METHODOLOGY

CAPTCHA security is investigated in two ways:

- Evaluate state-of-the-art neural networks' recognition ability of generated images and texts by accuracy rates, for text recognition using ResNet-50 [18] and LENET-5 [22], and for image classification using ResNet-101 [18] and VGG-16 [13].
- Evaluate CAPTCHA's security against adaptive attack.

C. RESULT ANALYSIS

1) IMAGE-BASED EVALUATION

In Table 1, in training, a dataset of 10,000 hand-labeled photos from 100 categories (each with 100 images) is employed,

while in testing, four test datasets are used, each including 1000 blended photos from 100 categories (each with ten testing images). In Table 2, some illustrated real samples were employed in the evaluation. Character lengths of three to five are also investigated for security evaluation. The results, shown in Table 3, demonstrated that the stylization version has a substantial fooling effect while the adversarial version clearly has a stronger impact on deceiving neural networks. As a result, the stylized adversarial version achieved the highest fooling rate on neural networks. Furthermore, the longer the text, the better the deception rate on neural networks.

2) TEXT-BASED EVALUATION

In Table 4, in training, a dataset of 10,000 hand-labeled photos from 10 categories (from 1 to 10 respectively, each with 1000 images) is employed, while in testing, four test datasets are used, each including 1000 blended photos from 10 categories (each with 100 testing images). In practice, automated bots also meet a lot of difficulties to address the position of each character in complicated image backgrounds, which we don't cover in this section. In Table 5, some illustrated real samples were used in the evaluation. The findings, shown in Table 6, indicated that the stylized adversarial versions can be resilient to attacks effectively. The networks ResNet-50 and LeNet-5 achieved high accuracy rates, greater than 95%, for normal versions, but their accuracy rates are very low for the stylized adversarial versions.

TABLE 3. Image-based accuracy rates.

| Recognition Network | Normal | | | | | | Adversarial | | | | | | Stylized Adversarial | | | | | |
|---------------------|--------|------|------|----------|------|------|--|------|------|--|------|------|--|------|------|--|------|------|
| | Normal | | | Stylized | | | Generated by ResNet-101 ($\epsilon = 0.1$) | | | Generated by VGG-16 ($\epsilon = 0.1$) | | | Generated by ResNet-101 ($\epsilon = 0.1$) | | | Generated by VGG-16 ($\epsilon = 0.1$) | | |
| | l=3 | l=4 | l=5 | l=3 | l=4 | l=5 | l=3 | l=4 | l=5 | l=3 | l=4 | l=5 | l=3 | l=4 | l=5 | l=3 | l=4 | l=5 |
| ResNet-101 | 87.1 | 85.3 | 85.1 | 70.5 | 68.3 | 67.7 | 47.1 | 46.3 | 45.7 | 45.3 | 43.6 | 42.7 | 37.1 | 36.5 | 35.7 | 35.3 | 33.5 | 31.8 |
| VGG-16 | 78.5 | 77.3 | 76.7 | 61.6 | 61.3 | 60.1 | 41.3 | 40.3 | 39.7 | 35.6 | 33.3 | 31.5 | 33.4 | 31.8 | 31.5 | 27.3 | 25.6 | 23.7 |

TABLE 4. Text-based evaluation datasets.

| Dataset | Quantity |
|--|----------|
| Test Dataset - Normal | 1000 |
| Test Dataset - Stylized version (applied Neural Style Transfer) | 1000 |
| Test Dataset - Adversarial version (applied Adversarial Examples) | 1000 |
| Test Dataset - Stylized Adversarial version (applied Adversarial Examples and Neural Style Transfer) | 1000 |
| Training Dataset | 10000 |

3) ADAPTIVE-BASED EVALUATION

Currently, some state-of-the-art techniques are being employed to deal with Adversarial Examples: gradient masking, adversarial training, and input transformation.

a: ENSEMBLE ADVERSARIAL TRAINING

This method [23] incorporates crafted examples from other static pre-trained models into a model’s training data. The original 10,000 training images are used to generate sets of stylized, adversarial, and stylized adversarial. These generated image sets were added to the training dataset. As a result, the training set contains 40,000 training images, shown in Table 7.

b: DISTILLATION

To reduce computing resource requirements, the goal of distillation [24] is to reduce the size of DNN architecture ensembles. The technique is to employ classification probability layers generated by a first DNN network to train a second DNN network without losing accuracy. The method is a gradient masking-based defense that adds a temperature constant T to the softmax function:

$$\text{softmax}(x, T)_i = \frac{e^{\frac{x_i}{T}}}{\sum_{j=0}^{N-1} e^{\frac{x_j}{T}}} \quad (5)$$

where x_i is the probability of class i th, N is the number of classes. In Figure 4, using the SoftMax function at

temperature T , we train the neural network on the training set. The neural network is then used to label each training data with soft labels. Finally, we use the SoftMax function to train the distilled model on the soft labels again at temperature T . The lower temperature in the SoftMax function, the more discrete the probability distribution (i.e., others are close to 0, and only one probability in output $F(X)$ is close to 1), whereas the higher temperature in the function SoftMax, the more ambiguous the probability distribution (i.e., given N the number of all possible probabilities, in output $F(X)$, all probabilities are close to $1/N$). In this experiment, the temperature T was picked up experimentally at the value 20 during the evaluation process.

c: THERMOMETER ENCODING

The thermometer encoding approach [25] is to disrupt the linear characteristics of neural networks. With each pixel color $x(i, j, c)$ of an image x , $\tau(x(i, j, c))$ is the l -level thermometer encoding vector:

$$\tau(x(i, j, c))_k = \begin{cases} 1 & \text{if } x(i, j, c) > \frac{k}{l} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where l is the one-hot encoding length for each pixel value, k is the bit index, i and j are coordinates of the pixel in the image, c is the color channel value at the pixel, the pixel value $x(i, j, c)$ is normalized within $[0, 1]$, and $\tau(x(i, j, c))$ is the bit value at the bit index k . For example, the 10-level thermometer encoding of 0.57, $\tau(0.57)$, is 1111100000. The model is then trained using thermometer encoding of training data.

d: EVALUATION

The results, shown in Tables 8 (for image-based recognition) and 9 (for text-based recognition), demonstrated the effectiveness of attacking methods, with ensemble adversarial training having the greatest impact on stylized adversarial versions. Ensemble adversarial training reduces the robustness of stylized and adversarial versions by 10 - 20%, whereas distillation and thermometer encoding reduce by 5 - 10%. Although attackers are well-versed in stylized adversarial example generation and defense strategies, they are

TABLE 5. Real evaluation text samples.






| Style | Normal | Stylized | Adversarial | Stylized Adversarial |
|---|---|---|---|---|
|  |  |  |  |  |

TABLE 6. Text-based accuracy rates.

| Recognition Network | Normal | Stylized | Adversarial | | Stylized Adversarial | |
|---------------------|--------|----------|--|--|--|--|
| | | | Generated by ResNet-101 ($\epsilon = 0.1$) | Generated by VGG-16 ($\epsilon = 0.1$) | Generated by ResNet-101 ($\epsilon = 0.1$) | Generated by VGG-16 ($\epsilon = 0.1$) |
| ResNet-50 | 97.5 | 43.6 | 28.1 | 25.5 | 17.3 | 15.8 |
| LeNet-5 | 95.7 | 37.6 | 19.7 | 17.3 | 9.3 | 7.8 |

TABLE 7. Adaptive evaluation datasets.

| Dataset | Ensemble Adversarial Training | Distillation | Thermometer Encoding |
|--|-------------------------------|--------------|----------------------|
| Test Dataset - Normal | 1000 | 1000 | 1000 |
| Test Dataset - Stylized version (applied for Neural Style Transfer) | 1000 | 1000 | 1000 |
| Test Dataset - Adversarial version (applied Adversarial Examples) | 1000 | 1000 | 1000 |
| Test Dataset - Stylized adversarial version (applied Neural Style Transfer and Adversarial Examples) | 1000 | 1000 | 1000 |
| Training Dataset | 40000 | 10000 | 10000 |

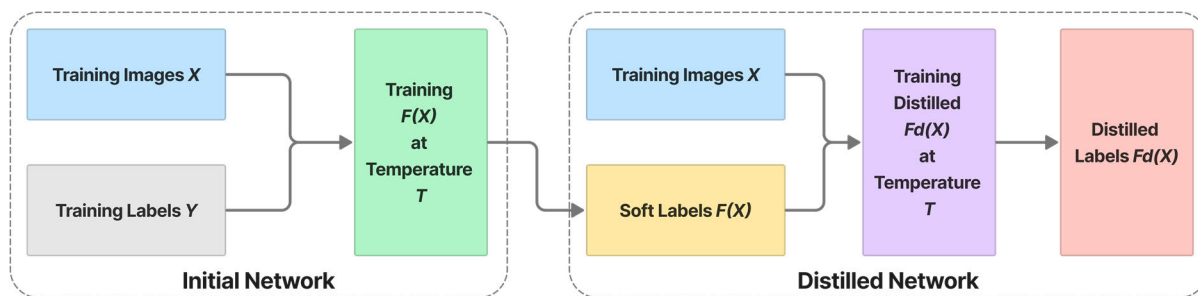


FIGURE 4. Distillation process.

unaware of specific models and methods used to generate stylized adversarial CAPTCHA versions. Furthermore, the attacker has access to all generated CAPTCHAs and has no

knowledge of the specific image or style datasets used to build the adversarial CAPTCHAs. Obviously, the above methods will be less effective in practice.

TABLE 8. Image-based adaptive accuracy rates.

| Adaptive Attack | Normal | | | Stylized | | | Adversarial | | | | | | Stylized Adversarial | | | | | | |
|-----------------|-------------------------------|------|------|----------|------|------|--|------|------|--|------|------|--|------|------|--|------|------|------|
| | | | | | | | Generated by VGG-16 ($\epsilon = 0.1$) | | | Generated by ResNet-101 ($\epsilon = 0.1$) | | | Generated by VGG-16 ($\epsilon = 0.1$) | | | Generated by ResNet-101 ($\epsilon = 0.1$) | | | |
| | l=3 | l=4 | l=5 | l=3 | l=4 | l=5 | l=3 | l=4 | l=5 | l=3 | l=4 | l=5 | l=3 | l=4 | l=5 | l=3 | l=4 | l=5 | |
| VGG-16 | Normal | 78.5 | 77.3 | 76.7 | 61.6 | 61.3 | 60.1 | 35.6 | 33.3 | 31.5 | 41.3 | 40.3 | 39.7 | 27.3 | 25.6 | 23.7 | 33.4 | 31.8 | 31.5 |
| | Ensemble Adversarial Training | 80.3 | 79.7 | 78.3 | 70.3 | 68.3 | 66.1 | 57.3 | 55.4 | 54.1 | 63.1 | 61.3 | 59.5 | 35.1 | 34.3 | 34.1 | 43.5 | 42.1 | 40.5 |
| | Distillation ($T=20$) | 77.3 | 76.1 | 76.3 | 65.1 | 64.3 | 61.2 | 42.3 | 41.7 | 40.3 | 49.3 | 47.1 | 46.2 | 29.5 | 28.7 | 27.3 | 37.5 | 36.3 | 35.1 |
| | Thermometer Encoding | 75.3 | 74.6 | 73.1 | 67.6 | 66.3 | 65.1 | 46.7 | 45.3 | 44.6 | 53.6 | 51.1 | 49.3 | 32.6 | 31.3 | 30.1 | 40.3 | 38.3 | 37.1 |
| ResNet-101 | Normal | 87.1 | 85.3 | 85.1 | 70.5 | 68.3 | 67.7 | 45.3 | 43.6 | 42.7 | 47.1 | 46.3 | 45.7 | 35.3 | 33.5 | 31.8 | 37.1 | 36.5 | 35.7 |
| | Ensemble Adversarial Training | 90.5 | 89.3 | 88.1 | 77.3 | 75.3 | 73.1 | 62.3 | 60.2 | 58.3 | 65.1 | 63.2 | 61.4 | 43.3 | 41.3 | 39.7 | 45.8 | 43.3 | 41.2 |
| | Distillation ($T=20$) | 85.3 | 84.8 | 83.2 | 73.1 | 72.3 | 72.5 | 52.4 | 50.4 | 50.1 | 53.3 | 52.8 | 52.1 | 38.1 | 35.2 | 33.7 | 40.2 | 37.5 | 34.1 |
| | Thermometer Encoding | 84.7 | 82.2 | 81.3 | 74.2 | 74.1 | 72.1 | 57.3 | 57.1 | 55.3 | 60.2 | 57.3 | 55.2 | 39.7 | 38.5 | 35.2 | 41.3 | 39.8 | 37.2 |

TABLE 9. Text-based adaptive accuracy rates.

| Adaptive Attack | Normal | | Stylized | | Adversarial | | | |
|-----------------|-------------------------------|------|----------|------|--|------|--|--|
| | | | | | Generated by VGG-16 ($\epsilon = 0.1$) | | Generated by ResNet-101 ($\epsilon = 0.1$) | |
| | | | | | | | | |
| LeNet-5 | Normal | 95.7 | 37.6 | 17.3 | 19.7 | 7.8 | 9.3 | |
| | Ensemble Adversarial Training | 97.3 | 47.3 | 23.1 | 24.3 | 15.3 | 13.8 | |
| | Distillation ($T=20$) | 94.5 | 41.5 | 21.8 | 23.4 | 12.3 | 11.5 | |
| | Thermometer Encoding | 93.2 | 44.7 | 21.2 | 23.1 | 14.6 | 12.3 | |
| ResNet-50 | Normal | 97.5 | 43.6 | 25.5 | 28.1 | 15.8 | 17.3 | |
| | Ensemble Adversarial Training | 98.1 | 48.2 | 34.1 | 37.3 | 23.4 | 26.4 | |
| | Distillation ($T=20$) | 95.8 | 47.4 | 29.2 | 31.1 | 17.1 | 19.8 | |
| | Thermometer Encoding | 94.3 | 45.1 | 31.5 | 34.4 | 20.9 | 23.1 | |

V. CONCLUSION

CAPTCHA will be a long-term battle between humans and computers. Humans leverage AI's hardness and cognitive abilities to compete with computers. Technology is evolving quickly, and computers with the most advanced hardware and software can now solve even the most challenging cognitive and AI challenges. To improve CAPTCHA security in distinguishing humans from automated bots, it must combine both machine learning, artificial intelligent techniques, and cognitive characteristics.

In our research, we proposed a new method to augment Adversarial Examples by combining it with Neural Style Transfer to improve the security of standard CAPTCHAs against deep learning abilities of recognition and classification. Recently, there have been many developed methods against Adversarial Examples such as Gradient Masking, Adversarial Training, or Input Transformation, etc. To validate the proposed method, extensive security evaluations were conducted in this study to determine the enhancement effectiveness of CAPTCHA security. The findings showed that this augmenting technique could considerably increase the security of standard CAPTCHAs by increasing Adversarial Examples' robustness against machine learning bots.

DISCLOSE STATEMENT

No potential conflict of interest was reported by the author(s).

DATA AVAILABILITY STATEMENT

The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

REFERENCES

- [1] N. Dinh and L. Ogiela, "Human-artificial intelligence approaches for secure analysis in CAPTCHA codes," *EURASIP J. Inf. Secur.*, vol. 2022, no. 1, p. 8, Dec. 2022.
- [2] Y. Li, M. Cheng, C.-J. Hsieh, and T. C. M. Lee, "A review of adversarial attack and defense for classification methods," *Amer. Statistician*, vol. 76, no. 4, pp. 329–345, Oct. 2022.
- [3] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," 2015, *arXiv:1508.06576*.
- [4] N. D. Trong, T. H. Huong, and V. T. Hoang, "New cognitive deep-learning CAPTCHA," *Sensors*, vol. 23, no. 4, p. 2338, Feb. 2023.
- [5] Y. Zhang, H. Gao, G. Pei, S. Kang, and X. Zhou, "Effect of adversarial examples on the robustness of CAPTCHA," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery (CyberC)*, Oct. 2018, pp. 1–109.
- [6] Y. K. Gajani, S. Bhardwaj, and M. Thenmozhi, "Guarding against bots with art: NST-based deep learning approach for CAPTCHA verification," in *Proc. Int. Conf. Recent Adv. Elect., Electron., Ubiquitous Commun., Comput. Intell. (RAEEUCCI)*, Chennai, India, 2023, pp. 1–5, doi: 10.1109/RAEEUCCI57140.2023.10134320.
- [7] M. Osadchy, J. Hernandez-Castro, S. Gibson, O. Dunkelman, and D. Pérez-Cabo, "No bot expects the DeepCAPTCHA! Introducing immutable adversarial examples, with applications to CAPTCHA generation," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2640–2653, Nov. 2017.
- [8] G. Ye, Z. Tang, D. Fang, Z. Zhu, Y. Feng, P. Xu, X. Chen, and Z. Wang, "Yet another text captcha solver: A generative adversarial network based approach," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 332–348.
- [9] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: An extension of MNIST to handwritten letters," 2017, *arXiv:1702.05373*.
- [10] (Mar. 11, 2021). *ImageNet*. [Online]. Available: <http://www.image-net.org>
- [11] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, Netherlands, 2016, pp. 694–711.
- [12] G. Ghiasi, H. Lee, M. Kudlur, V. Dumoulin, and J. Shlens, "Exploring the structure of a real-time, arbitrary neural artistic stylization network," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–11.
- [13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
- [14] D. Ulyanov, V. Lebedev, and A. Vedaldi, "Texture networks: Feed-forward synthesis of textures and stylized images," in *Proc. ICML*, vol. 1, no. 2, 2016, pp. 1349–1357.
- [15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
- [16] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation of the artistic style," in *Proc. Int. Conf. Learned Represent. (ICLR)*, 2016, pp. 1–9.
- [17] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.
- [18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*.
- [19] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graphics*, vol. 22, no. 3, pp. 313–318, 2003.
- [20] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, and M. Kudlur, "TensorFlow: A system for large scale machine learning," in *Proc. OSDI*, vol. 16, Nov. 2016, pp. 265–283.
- [21] A. Paszke, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–9.
- [22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, Nov. 1998.
- [23] F. Tramér, A. Kurakin, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *Proc. ICLR*, 2018, pp. 1–20.
- [24] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 582–597.
- [25] J. Buckman, A. Roy, C. Raffel, and I. J. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," *Proc. ICLR*, 2018, pp. 1–22.



NGHIA DINH received the M.Sc. degree in software engineering from the University of Bordeaux, France. He is currently pursuing the Ph.D. degree with the VSB—Technical University of Ostrava, Czech Republic. He is a Software Architecture Enthusiast and Computer Scientist. He has contributed to the success of many open sources and technology companies.



KIET TRAN-TRUNG received the master's degree from Ho Chi Minh City Pedagogical University, Vietnam. He is currently a Lecturer with Ho Chi Minh City Open University. His research interests include machine learning and computer vision.



VINH TRUONG HOANG received the master's degree from the University of Montpellier and the Ph.D. degree in computer science from the University of the Littoral Opal Coast, France. He is currently an Assistant Professor and the Head of the Image Processing and Computer Graphics Department, Ho Chi Minh City Open University, Vietnam. His research interests include image analysis and feature selection.