Meta Perturbation Generation Network for Text-Based CAPTCHA

Zhuoting Wu[1], Zhiwei Guo[1], Jiuxiang You[1], Zhenguo Yang[1(✉)], Qing Li[2], and Wenyin Liu[3]

[1] Guangdong University of Technology, Guangzhou, China
yzg@gdut.edu.cn
[2] The Hong Kong Polytechnic University, Hong Kong, China
qing-prof.li@polyu.edu.hk
[3] Zhongguancun Laboratory, Beijing, People's Republic of China
liuwy@gdut.edu.cn

Abstract. With the development of text-based CAPTCHA, many adversarial example generation methods for text-based CAPTCHA have been proposed. However, the perturbation factors generated by the existing methods are simple and easy to be attacked. In this paper, we present a framework for meta perturbation text-based CAPTCHA generation (denoted as MAPFN), which enhances the security of text-based CAPTCHA and makes the perturbed images friendly for humans. More specifically, we propose a meta perturbation generation network (MPGN) to construct rich and effective perturbation factors. To this end, we devise a perturbation feature fusion module (PFFM) to fuse the perturbation factors generated by MPGN into a new perturbation factor, which can be applied to the CAPTCHA image to make it similar to the origin while being effectively against the attacker models. Extensive experiments on 8 real website CAPTCHA datasets show the excellent performance of the proposed MAPFN. (e.g., attack accuracy falls from 93.99% to 0.98% on the NSFC dataset).

Keywords: Text-based CAPTCHA · Meta-learning · Adversarial perturbation · CAPTCHA images generation

1 Introduction

CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart) [1] is a security mechanism used by most websites or applications to protect the information of websites and users. It mainly distorts texts, pictures, sounds, and other information to tell machines from humans. Among such, text-based CAPTCHA is frequently used because of its convenient operation and simple comprehension.

In recent years, automatic text-based CAPTCHA solutions have gained popularity as a research area [20]. Existing machine learning methods, like shallow models, optical character recognition (OCR) models, and numerous deep

H. Duan et al. (Eds.): SecureComm 2023, LNICST 567, pp. 110–124, 2025.
https://doi.org/10.1007/978-3-031-64948-6_6

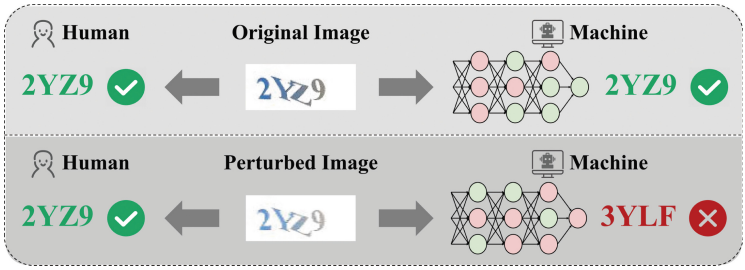

Fig. 1. The purpose of the proposed MAPFN. Humans can successfully recognize images with added perturbations, but machines cannot.

neural network models, have achieved great success in recognizing text-based CAPTCHA. In order to increase the recognition difficulty of the text-based CAPTCHA, researchers have suggested two categories of approaches. Quite a few works make an attempt to add occlusion lines or distort characters. For instance, Ferzli et al. [6] created a novel character that was included in the CAPTCHA based on the masking properties of the HVS. A 3D CAPTCHA scheme built by Kim et al. [13] combined a few scattered spheres into text characters. However, these approaches make text-based CAPTCHAs more complex and challenging for humans to recognize, which is contrary to the original intent of CAPTCHA design. Researchers could increase the perturbation to the CAPTCHA by proposing adversarial factors. For example, Zheng et al. [27] presented a random distribution scheme for generating CAPTCHAs from adversarial samples based on user behavior. Matsuura et al. [18] used spatial smoothing to generate adversarial text-based CAPTCHAs. Regardless, the aforementioned approaches rely on a specific single adversarial perturbation technique, and the generated images can be easily recognized by advanced attacker models. Consequently, how to improve the user-friendliness and security of text-based CAPTCHA without affecting regular human recognition is a challenge.

In the context of meta-learning, MIT-IBM Watson AI Lab et al. [25] proposed a Meta Adversarial Perturbation (MAP) method based on model-agnostic meta-learning that updated natural images with a high likelihood of misclassification through one-step gradient rise. Yuan et al. [26] developed a novel architecture named Meta Gradient Adversarial Attack (MGAA), to generate adversarial examples using any existing gradient-based attack method iteratively.

In this paper, we are inspired to propose a new framework for meta perturbation text-based CAPTCHA generation (denoted as MAPFN), which improves the security of text-based CAPTCHA and is user-friendly. On one hand, we design a new perturbation factor generation network (MPGN) based on meta adversarial perturbation. Rich effective perturbation factors are constructed by it. The perturbation factors are updated through the perturbed image generated by MPGN in each iteration, making it impossible for specific attacker models to properly recognize the resulting generated image. On the other hand, a perturbation feature fusion module (PFFM) is built to fuse the perturbation factors

generated by MPGN into a new perturbation factor, which is then added to the CAPTCHA image. It seeks to make the generated perturbed image as similar to the original image as feasible while assisting it effectively against the attacker models. The purpose of the proposed MAPFN is shown ine shown in Fig. 1.

The main contributions are summarized as follows:

- We propose a novel framework for meta perturbation text-based CPATCHA generation by constructing effective perturbation factors to obtain the perturbed CAPTCHA that is difficult to be attacked by machines while being easily recognized by humans.
- We devise an adversarial perturbation feature fusion module to generate a new perturbation factor by fusing multiple perturbation factors generated from the set of perturbation methods while effectively retaining the features of the multiple generated perturbation factors.
- For the collected 8 CAPTCHA datasets, we conduct qualitative and quantitative experiments to show that the proposed approach effectively improves the security of CAPTCHA. In particular, machine recognition accuracy decreases from 90.0% to 0 for individual datasets.

The rest of this paper is organized as follows. Section 2 summarizes the related works. Section 3 presents the proposed MAPFN. The experimental results are presented and analyzed in Sect. 4. Section 5 concludes the work.

2 Related Work

2.1 Traditional Text-Based CAPTCHA

CAPTCHA schemes are similar to Turing tests, nevertheless, they differ in that a machine is now being judged [16]. As a result, a lot of websites use CAPTCHAs as a network security measure, such as Yahoo, Hotmail, Weibo, etc. [2,12]. Bursztein et al. [5] proposed the text-based CAPTCHA scheme, which is generated by a sequence of random characters or words from lower to upper case and digits. It aims to defend against attacks from machine programs that take advantage of the differences between humans and computers in the recognition of character sequences.

In order to resist attacks, related studies usually add some security features to CAPTCHA, such as multiple fonts, font sizes, fuzzy letters, fluctuations, etc. The goal is to make traditional text-based CAPTCHAs more secure. In this regard, Ahn et al. [3] designed a CAPTCHA generation method by overlapping the text characters in the CAPTCHA so that it had negative cross regions. Kim et al. [13] presented a novel 3D CAPTCHA design scheme that used text characters made up of many scattered spheres. Visual cryptography was used by Yan et al. [23] to design a CAPTCHA scheme for text enhancement. However, the text-based CAPTCHA images generated by the aforementioned methods make it easy for users to have trouble recognizing text or characters. This makes it difficult to successfully recognize the CAPTCHAs.

2.2 Adversarial Text-Based CAPTCHA

At present, most deep neural networks are used by malicious attacks, which poses a great threat to the security of CAPTCHA. In recent years, the research on adversarial attacks on deep learning neural networks has served as inspiration for strengthening the security of CAPTCHAs, which apply the influence characteristics of perturbations on deep neural networks to CAPTCHAs. The adversarial example was suggested by Szegedy et al. [8]. Perturbation generation methods were used to generate new CAPTCHA images in the CAPTCHA datasets. It can reduce the probability that machines successfully recognize CAPTCHA images.

In the field of CAPTCHA, it has been advised to use adversarial perturbation approaches for adversarial text-based CAPTCHA generation. For instance, Shi et al. [22] proposed a generation framework for adversarial text-based and image-based CAPTCHAs using the JSMA method. Kwon et al. [16] utilized adversarial perturbation attack methods, including the Fast Gradient Sign Method (FGSM) and Iterative Fast Gradient Sign Method (I-FGSM), to generate CAPTCHA. The above methods directly apply the commonly used perturbation methods to the CAPTCHA, which makes it easy to be identified by the attacker models. Because they are difficult to generate rich and complex effective perturbation factors. It is unable to effectively improve the security of CAPTCHA.

2.3 Meta Adversarial Learning

Meta-learning is the concept of using knowledge from previous tasks to quickly learn how to solve new tasks. As a new method proposed in recent years, meta-learning has been widely used in various tasks. A model-agnostic algorithm for meta-learning, called MAML, was put forward by Chelsea et al. [7]. It can be applied to any model optimized with gradient descent, including regression, classification, and reinforcement learning. Inspired by MAML, MIT-IBM Watson AI Lab et al. [25] created Meta Adversarial Perturbation (MAP) method. By learning perturbations, this method enabled perturbed gradient-based iterative methods to quickly adapt to new data in one or a few iterations. In the same year, Yuan et al. [26] developed a novel architecture for image recognition dubbed Meta Gradient Adversarial Attack (MGAA), which generated adversarial examples by iteratively using any existing gradient-based attack method. Hu et al. [11] innovatived a novel black-box adversarial attack method to combine meta-learning with substitute model training. Fei et al. [24] proposed a meta-learning framework by training a meta-generator to produce perturbations for natural images.

3 Methodology

3.1 Overview of the Framework

The overall framework of the proposed MAPFN is shown in Fig. 2, which consists of two components, i.e., the Meta Perturbation Generation Network (MPGN)

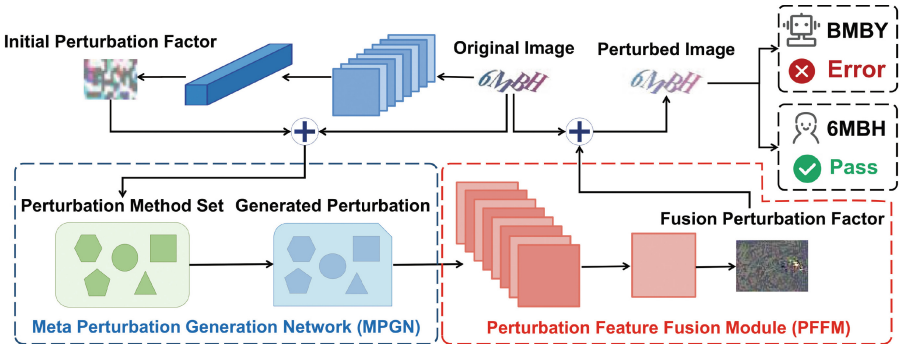

Fig. 2. Overview of **MAPFN**

and the Perturbation Feature Fusion Module (PFFM). Specifically, the MPGN constructs rich and effective perturbation factors through a range of advanced perturbation methods based on the concept of MAP. Furthermore, the PFFM component creates a new perturbation factor by merging the generated perturbation factors. The resulting perturbation factor is added to the original CAPTCHA to generate a perturbed CAPTCHA, which helps the CAPTCHA to effectively against the attacker models.

3.2 Meta Perturbation Generation Network (MPGN)

The existing adversarial CAPTCHA generation methods usually directly apply a specific perturbation method to CAPTCHA images. The resulting images are easily recognized by attacker models. MPGN is designed to generate rich perturbation factors using an adversarial perturbation strategy based on the idea of MAP, and the generated CAPTCHA image is challenging for machines to recognize. The details of MPGN are provided in Fig. 3.

Given the original image Img_{ori} and the corresponding label $Label_{ori}$, the label is decoded to obtain the original label feature $F_{ori} \in \mathbb{R}^{num}$, where num is the number of characters in the label.

$$F_{ori} = decode(Label_{ori}) \tag{1}$$

The original feature C of the original CAPTCHA image Img_{ori} is extracted by using convolutional neural network (CNN). Furthermore, the C is input into the fully connected layer to calculate the target label feature $F_t \in \mathbb{R}^{num}$, where num is the number of characters that may appear in the CAPTCHA result. The operation can be represented as follows:

$$C = \sigma(W_c \cdot CNN(Img_{ori})) \tag{2}$$

$$F_t = \sigma(W_1 \cdot LSTM_{bi}(C)) \tag{3}$$

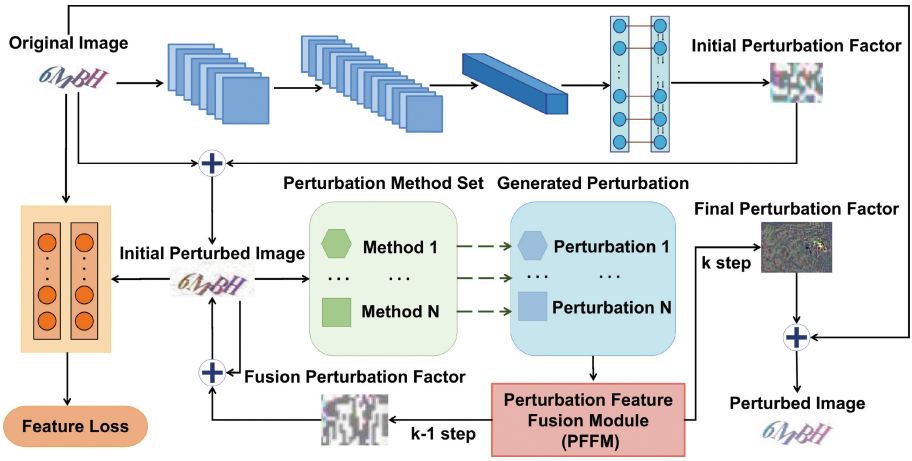

Fig. 3. Structure of Meta Perturbation Generation Network **(MPGN)**

where $CNN(\cdot)$ is the convolutional neural network (CNN) structure, W_c and W_1 are the weight matrix of the fully connected layer, $\sigma(\cdot)$ represents the used activation function and $LSTM_{bi}(\cdot)$ represents the Bi-LSTM model.

We calculate the loss value between the F_t and F_{ori}. The preliminary perturbation factor V_{pre} can be obtained through optimization.

$$V_{pre} = \nabla_{Img_{ori}} Loss_{ctc}(F_{ori}, F_t) \tag{4}$$

where $Loss_{ctc}$ is the loss function used to measure the similarity between both label features. The detail of the $Loss_{ctc}$ can be seen in Session 3.4.

We add the preliminary perturbation factor V_{pre} to the original image Img_{ori} to obtain the initial perturbed image Img'_1, and calculate the loss value $Loss_{feature}$ between the target label feature $F_{out}{}'_1$ obtained by Img'_1 and the original label feature F_{ori}. The operation can be seen as follows:

$$\mathcal{L}_i = Loss_{feature}(F_{ori}, F_{out}{}'_i) \tag{5}$$

where $Loss_{leature}$ is the loss function used, and the details about it can be seen in Session 3.4.

Next, we input the initial perturbed image Img'_1 to the set of advanced perturbation methods $MS = [M_1, M_2, \cdots, M_n]$, where n is the number of methods in the set, to generate the perturbation factors set $VS_1 = [v^1{}_1, v^1{}_2, \cdots, v^1{}_n]$. The generated perturbation factors set VS_1 is input into the PFFM (see Session 3.3 for details) to acquire the fused perturbation factor $v_f{}'_1$. Subsequently, the fused perturbation $v_f{}'_1$ is added to Img'_1 to create a new perturbed image Img'_2. In the i^{th} iteration, the perturbed image Img'_i is input into MS to generate perturbation factors set $VS_i = [v^i{}_1, v^i{}_2, \cdots, v^i{}_n]$, and the fusion perturbation $v_f{}'_i$ is obtained by VS_i through the PFFM. The operation can be seen as follows:

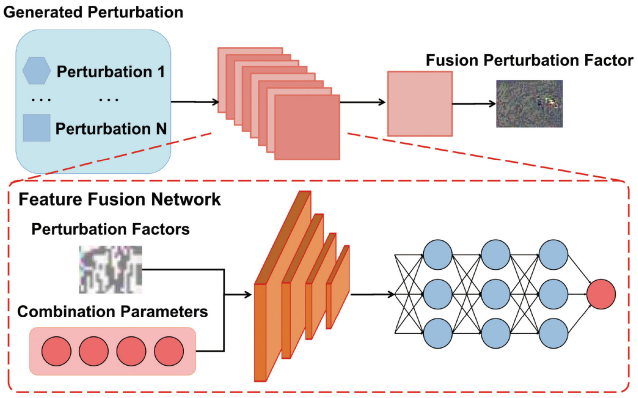

Fig. 4. Details of Perturbation Feature Fusion Module **(PFFM)**

$$v_{f_i}' = Mix(VS_i) = Mix(MS_{M_1,\cdots,M_n}(Img'_i)) \tag{6}$$

$$Img'_{i+1} = Img'_i + v_{f_i}' \tag{7}$$

where $Mix(\cdot)$ is the function of PFFM.

After k iterations, the final fused perturbation v_{f_k}' is output as the final perturbation v_{fin}, where k is the set number of iterations. We adopt FGSM [8] as the optimizer, and add the final perturbation v_{fin} to the original CAPTCHA image Img_{ori} by FGSM to obtain the final perturbed CAPTCHA image Img_{adv}:

$$Img_{adv} = Img_{ori} + \epsilon \cdot sign(v_{fin}) \tag{8}$$

where ϵ is the step size.

3.3 Perturbation Feature Fusion Module (PFFM)

In order to increase the difficulty of machine recognition, PFFM implements the feature fusion of multiple generated perturbation factors to form a new perturbation factor. Figure 4 depicts the PFFM in detail.

The module completes the feature fusion by changing the feature dimension based on 1×1 convolution kernel [17]. N perturbation factors in VS_i generated by MS are combined, and their four dimensions: number of samples, height, width and number of channels are as a matrix. The matrix is input to the meta adversarial convolution fusion network constructed based on 1×1 convolution kernel optimization. During the fusion process, the meta adversarial convolution fusion network maintained the characteristics of 1×1 convolution kernel, it does not slide in the width and height directions of the input and output, and retains the original planar structure of the perturbed feature maps. To achieve the purpose of changing the feature dimension (dimensionality reduction), it

Algorithm 1. The MAPFN algorithm

Input: the original image Img_{ori}, the original label $Lael_ori$, advanced perturbation methods MS.

Output: the perturbed CAPTCHA image Img_{adv}

Given the original image Img_{ori} and the original label $Lael_ori$, decode and calculate the label features, F_{ori} and F_t
$Loss_{ctc} = \text{CTC}(F_{ori}, F_t)$
$V_{pre} \leftarrow \nabla Loss_{ctc}$
$\mathcal{L} \leftarrow 0$
Generate the initial perturbed image $Img'_{pre} \leftarrow Img_{ori} + V_{pre}$
for all number of training iterations k **do**
  Calculate the feature loss between Img_{ori} and Img'_i: $\mathcal{L}_i$ as Equ. 10
  for all M_j in MS **do**
    $v_j^i = M_j(Img'_i)$
  end for
  $v_{f_i}' \leftarrow Mix_{PFFM}(v^i{}_1, v^i{}_2, \cdots, v^i{}_n)$
  $Img'_{i+1} \leftarrow Img'_i + v_{f_i}'$
end for
$v_{fin} \leftarrow v_{f_k}'$
$Img_{adv} = Img_{ori} + \epsilon \cdot sign(v_{fin})$
return Img_{adv}

performs a convolution operation in the channel direction, which means that each pixel is linearly combined in different channels (feature information integration). N perturbed feature maps are merged into one perturbed feature map.

$$v_{f_i}' = track(clist(VS_i, N)) = track(clist((v^i{}_1, v^i{}_2, \cdots, v^i{}_n), N)) \tag{9}$$

where $track(\cdot)$ is the fusion function applied to the combined perturbation, $clist(\cdot)$ is a function that combines N perturbation factors from generate perturbation factors $VS_i = [v^i{}_1, v^i{}_2, \cdots, v^i{}_n]$.

The meta adversarial convolution fusion network is built using the 1×1 convolution kernel since the design purpose of PFFM is to realize the fusion of several perturbed feature maps. It is possible to employ a number of weight parameters in the fusion process without having an impact on other parameters of the perturbation feature map. At the same time, it is not necessary to fix the output size, which can be better applied to perturbed feature maps generated by CAPTCHA with different types of sizes.

3.4 Loss Function

Connectionist Temporal Classification (CTC). CTC is used to generate the perturbed image and calculate the loss between a source sequence and a target sequence. $Loss_{ctc}$ sums the probability of possible alignments of the input with the target, producing a loss value that is differentiable with respect to each input node [9].

Feature Loss Function. Its purpose is to ensure that attacker models cannot effectively recognize the perturbed image by calculating the feature gap between the original image and the perturbed image. Given the features of the source image and the generated perturbed image, the proposed loss can be as follows:

$$Loss_{feature}(F_{ori}, F_{out_i}') = MSE(\cos_{similar}(F_{ori}, F_{out_i}')) \tag{10}$$

where $MSE(\cdot)$ represents the mean square error, $\cos_{similar}(\cdot)$ on behalf of the cosine-similarity function.

The detailed steps of MAPFN are summarized in Algorithm 1.

4 Experiments

4.1 Datasets

We collected CAPTCHA images from eight websites as the datasets, including CET[1], CNKI[2], Gdgov[3], Gohimall[4], Jxfw[5], NRA[6], NSFC[7], and Weibo[8]. The eight datasets are chosen because of their different types and practical research value. Among eight datasets, the Weibo dataset is the one used by most CAPTCHA security schemes.

4.2 Baselines

We take the CTC (Connectionist Temporal Classification) [9] and ResNet [10] models as the attacker. Because most CAPTCHA recognition models use CTC and Resnet as the basic model structure. We use five advanced perturbation methods in the perturbation method set to measure the effectiveness of MAPFN, including FGM [19], FGSM [8], I-FGSM [14], MI-FGSM [15] and JSMA [21]. Accuracy is adopted for quantitative evaluations, and Amazon Mechanical Turk (AMT) is exploited for qualitative evaluations.

4.3 Quantitative Evaluations of Our Approach

Table 1 summarizes the performance of MAPFN on eight datasets, from which we have following observations. 1) Compared with the original images, the accuracy of the CAPTCHA generated by MAPFN is all reduced. For both attacker models, the accuracy of each dataset is decreased by about half and even by 80–90% in some datasets. 2) The proposed MAPFN achieves great performance, benefiting from the MPGN to generate rich perturbation factors using a new adversarial perturbation strategy and the PFFM to create a new perturbation factor by fusing rich perturbation factors.

[1] https://passport.neea.edu.cn/.
[2] http://my.cnki.net/elibregister/.
[3] https://tyrz.gd.gov.cn/.
[4] http://www.gohimall.cn/.
[5] Anonymous link.
[6] https://login.nra.gov.cn/.
[7] https://grants.nsfc.gov.cn.
[8] https://weibo.com/signup/.

Table 1. The performance (%) of MAPFN. Small values are better.

Model	Images	Dataset							
		CET	CNKI	Gdgov	Gohimall	Jxfw	NRA	NSFC	Weibo
CTC	Original	93.34	90.03	51.83	100.00	89.17	48.50	93.99	89.17
	Ours	**23.08**	**0.00**	**3.70**	**36.00**	**50.43**	**0.91**	**0.98**	**0.09**
ResNet	Original	100.00	94.31	100.00	100.00	98.97	100.00	100.00	99.28
	Ours	**47.12**	**19.80**	**42.59**	**19.00**	**22.05**	**24.77**	**14.71**	**22.93**

Table 2. Performance (%) of MAPFN on binary images. Small values are better.

Model	Binary Images	Dataset							
		CET	CNKI	Gdgov	Gohimall	Jxfw	NRA	NSFC	Weibo
CTC	Original	92.31	89.67	50.10	96.00	85.30	45.87	93.14	79.22
	Ours	**48.08**	**2.55**	**14.81**	**94.00**	**48.21**	**3.67**	**80.39**	**2.05**
ResNet	Original	100.00	96.47	100.00	96.00	99.49	100.00	100.00	98.87
	Ours	**21.15**	**9.02**	**29.63**	**0.00**	**58.63**	**2.75**	**11.76**	**26.10**

4.4 Performance of MAPFN on Binary Images

In order to evaluate the performance of MAPFN on binary images, we train the attacker models by converting the images into binary images through simple thresholding for CAAPTCHA recognition, and the rest of the experimental settings remain unchanged. The results shown in Table 2 summarize the performance of MAPFN on eight datasets processed by the attacker model, from which we can observe that the accuracy of CAPTCHA images generated by MAPFN is still reduced compared with that of the original images. It proves that the proposed MAPFN still achieves great performance.

4.5 Impact of Hyperparameter

The accuracy of perturbed images in the PFFM of MAPFN depends significantly on the number of perturbation factors N fused into the new perturbation factor. We compare the accuracy of the generated perturbed images against the CTC model with different numbers N in Table 3. The accuracy of the generated perturbed CAPTCHA is lower and more defensive as the number of perturbation factors for fusion increase. The reason is that the complexity of the generated perturbation feature increases, when more perturbation factors are fused. As a result, it is difficult for the attacker models to identify the perturbed CAPTCHA.

Table 3. The performance (%) of MAPFN under different number of perturbation factors to fuse against CTC. Small values are better.

N	Dataset							
	CET	CNKI	Gdgov	Gohimall	Jxfw	NRA	NSFC	Weibo
0	93.34	90.03	51.83	100.00	89.17	48.50	93.99	89.17
1	36.54	26.27	6.29	49.00	63.76	28.44	56.86	13.72
2	36.54	26.27	6.29	49.00	63.76	28.44	56.86	13.72
3	26.92	25.29	3.91	39.00	55.56	22.94	52.96	8.70
4	23.98	16.27	3.78	38.45	53.26	21.10	50.98	6.96
5	**23.08**	**0.00**	**3.70**	**36.00**	**50.43**	**0.91**	**0.98**	**0.09**

4.6 Performance of the Approaches

We compare the performance of our proposed method with that of basic perturbation methods to evaluate the performance of MAPFN. The experimental results of perturbed CAPTCHA generated from 8 datasets under the attack of the CTC model are displayed in Table 4. Table 4 exhibits that the defense success rate of our proposed MAPFN outperforms other perturbation methods (the accuracy is lower, the defense success rate is higher). This indicates that the perturbation generated by MAPFN effectively widens the gap of the CAPTCHA at the feature level and increases the attack difficulty of the recognition model. Yet the accuracy of MAPFN is only slightly lower than GDM and MI-FGSM in JXFW dataset due to background interference in the images. And MAPFN preserves the original design of the CAPTCHA during generation.

Table 4. Performance (%) of the approaches against CTC. Small values are better.

Model	Method	Dataset							
		CET	CNKI	Gdgov	Gohimall	Jxfw	NRA	NSFC	Weibo
CTC	GDM [4]	88.46	3.14	26.85	84.00	73.85	43.12	55.88	52.41
	FGM [19]	90.38	3.73	28.78	79.00	**15.67**	4.59	10.78	5.02
	FGSM [8]	90.38	0.59	6.48	78.00	29.33	21.10	68.63	0.10
	I-FGSM [14]	88.46	0.98	12.04	82.00	42.00	25.69	75.49	0.20
	MI-FGSM [15]	89.42	0.59	14.81	80.00	51.33	33.03	83.33	0.31
	JSMA [21]	27.88	2.94	16.67	89.00	46.00	2.92	1.96	25.36
	Ours	**23.08**	**0.00**	**3.70**	**36.00**	50.43	**0.91**	**0.98**	**0.09**

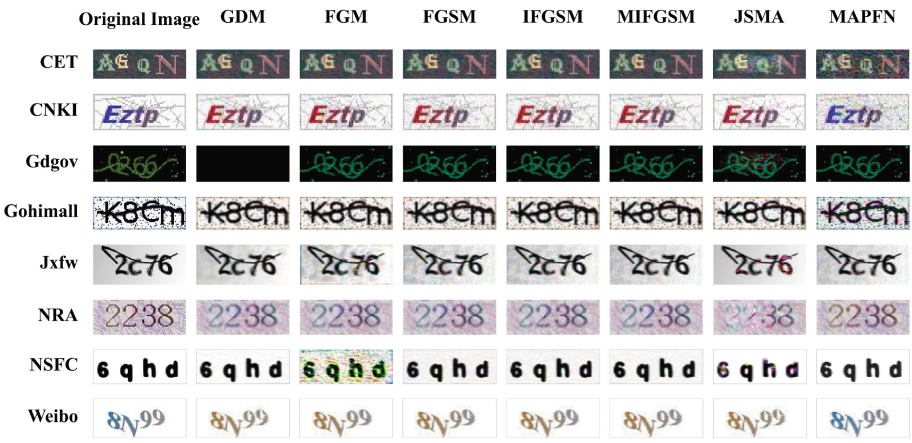

Fig. 5. The CAPTCHA examples of all methods.

4.7 Qualitative Evaluations of the Approaches

Figure 5 shows the performance of basic perturbation methods and MAPFN on all datasets, from which we have some observations. 1) The images generated by basic perturbation methods are different from the original images and feature glaring noise. For instance, the color of the images generated by FGSM in CNKI, NRA, and Weibo datasets has altered, which has a significant impact on human recognition. 2) The CAPTCHA images generated by MAPFN not only maintain user-friendliness like the original images but also increase appropriate perturbation to affect the attacker models. The results indicate that the proposed MAPFN has no apparent impact on the original CAPTCHA design in human vision.

4.8 Qualitative Evaluations with AMT

We randomly selected 280 group CAPTCHA images from 8 real-world datasets and recruited 560 volunteers to assess the degree of similarity between each set of images and recognize the results. The 560 volunteers for the experiment are selected at random. Table 5 and Fig. 6, respectively summarize the accuracy of human recognition and the perception of image similarity, from which we have some observations. 1) The recognition accuracy of the original images and the generated perturbed images differ just slightly. 2) 92% of the volunteers believe that the perturbed images generated by MAPFN are highly similar to the original images. Some of the volunteers chose little similarity because the background color of the perturbed images has changed in their opinion. 3) According to the experimental results, the CAPTCHA generated by MAPFN is highly similar to the original CAPTCHA and highly user-friendly for humans, making it possible for humans to recognize it successfully.

Table 5. Human Recognition (%) of AMT. Large values are better

Images	Dataset							
	CET	CNKI	Gdgov	Gohimall	Jxfw	NRA	NSFC	Weibo
Original	87.93	**98.48**	91.53	88.14	88.89	**100**	90.24	**96.49**
Ours	**89.66**	96.97	**91.53**	**88.14**	**90.48**	98.46	**90.24**	94.74

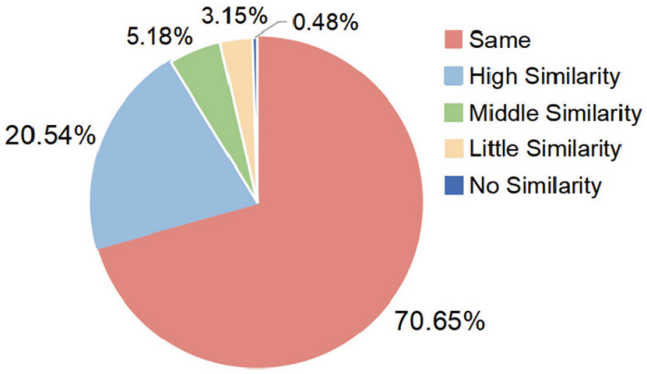

Fig. 6. The similarity comparison results of AMT between ours perturbed images and the original images.

5 Conclusion

In this paper, we propose a novel framework for meta perturbation text-based CPATCHA generation (called MAPFN), balancing user-friendliness and security. To generate rich and effective perturbation factors that might deceive the attacker models, we specifically exploit a perturbation generation network (MPGN) with a new adversarial perturbation strategy based on the concept of MAP. In particular, the proposed network introduces a perturbation feature fusion module to fuse a new perturbation factor, which can be added to the CAPTCHA image to make it imperceptible to humans. Additionally, our proposed MAPFN demonstrates much lower recognition accuracy by the CAPTCHA solvers compared to prior works, making the resulting CAPTCHA easy for humans but difficult for machines.

References

1. von Ahn, L., Blum, M., Hopper, N.J., Langford, J.: CAPTCHA: using hard AI problems for security. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 294–311. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_18
2. von Ahn, L., Blum, M., Langford, J.: Telling humans and computers apart automatically. Commun. ACM **47**(2), 56–60 (2004). https://doi.org/10.1145/966389.966390

3. von Ahn, L., Maurer, B., McMillen, C., Abraham, D., Blum, M.: reCAPTCHA: human-based character recognition via web security measures. Science **321**(5895), 1465–1468 (2008). https://doi.org/10.1126/science.1160379
4. Akhtar, N., Mian, A.: Threat of adversarial attacks on deep learning in computer vision: a survey. IEEE Access **6**, 14410–14430 (2018). https://doi.org/10.1109/ACCESS.2018.2807385
5. Bursztein, E., Martin, M., Mitchell, J.: Text-based captcha strengths and weaknesses. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS 2011, pp. 125–138. Association for Computing Machinery, New York (2011). https://doi.org/10.1145/2046707.2046724
6. Ferzli, R., Bazzi, R., Karam, L.J.: A captcha based on the human visual systems masking characteristics. In: 2006 IEEE International Conference on Multimedia and Expo, pp. 517–520 (2006). https://doi.org/10.1109/ICME.2006.262439
7. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 1126–1135. PMLR (2017)
8. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples (2015)
9. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning, ICML 2006, pp. 369–376. Association for Computing Machinery, New York (2006). https://doi.org/10.1145/1143844.1143891
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
11. Hu, C., Xu, H.Q., Wu, X.J.: Substitute meta-learning for black-box adversarial attack. IEEE Signal Process. Lett. **29**, 2472–2476 (2022). https://doi.org/10.1109/LSP.2022.3226118
12. Kim, D., Sample, L.: Search prevention with captcha against web indexing: a proof of concept. In: 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), pp. 219–224 (2019). https://doi.org/10.1109/CSE/EUC.2019.00049
13. Kim, S., Choi, S.: DotCHA: a 3D text-based scatter-type CAPTCHA. In: Bakaev, M., Frasincar, F., Ko, I.-Y. (eds.) ICWE 2019. LNCS, vol. 11496, pp. 238–252. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19274-7_18
14. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. arXiv abs/1607.02533 (2016)
15. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial machine learning at scale. arXiv abs/1611.01236 (2016)
16. Kwon, H., Yoon, H., Park, K.W.: Robust captcha image generation enhanced with adversarial example methods. IEICE Trans. Inf. Syst. **E103.D**(4), 879–882 (2020). https://doi.org/10.1587/transinf.2019EDL8194
17. Lin, M., Chen, Q., Yan, S.: Network in network. CoRR abs/1312.4400 (2013)
18. Matsuura, Y., Kato, H., Sasase, I.: Adversarial text-based captcha generation method utilizing spatial smoothing. In: 2021 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2021). https://doi.org/10.1109/GLOBECOM46510.2021.9685046

19. Miyato, T., Dai, A.M., Goodfellow, I.: Adversarial training methods for semi-supervised text classification. In: International Conference on Learning Representations (2017)
20. Mohamed, M., et al.: A three-way investigation of a game-captcha: automated attacks, relay attacks and usability. In: Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security, ASIA CCS 2014, pp. 195–206. Association for Computing Machinery, New York (2014). https://doi.org/10.1145/2590296.2590298
21. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: 2016 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 372–387 (2016). https://doi.org/10.1109/EuroSP.2016.36
22. Shi, C., et al.: Adversarial captchas. IEEE Trans. Cybern. **52**(7), 6095–6108 (2022). https://doi.org/10.1109/TCYB.2021.3071395
23. Yan, X., Liu, F., Yan, W., Lu, Y.: Applying visual cryptography to enhance text captchas. Mathematics **8**, 332 (2020). https://doi.org/10.3390/math8030332
24. Yin, F., et al.: Generalizable black-box adversarial attack with meta learning (2023)
25. Yuan, C.H., Chen, P.Y., Yu, C.M.: Meta adversarial perturbations. arXiv abs/2111.10291 (2021)
26. Yuan, Z., Zhang, J., Jia, Y., Tan, C., Xue, T., Shan, S.: Meta gradient adversarial attack. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 7728–7737 (2021). https://doi.org/10.1109/ICCV48922.2021.00765
27. Zheng, W., Wang, W., Ren, W., Feng, S., Liu, S., Ren, Y.: A user behavior-based random distribution scheme for adversarial example generated captcha. In: 2021 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), pp. 1215–1221 (2021). https://doi.org/10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00167