Unlearnable CAPTCHA

Che-Yu Wu
National TsingHua University
Department of Computer Science

Jia-Yuan Tien
National TsingHua University
Department of Computer Science

Chih-Wei Huang
National TsingHua University
Department of Computer Science

Tzu-Ling Liu
National TsingHua University
Department of Interdisciplinary
Program of Engineering

Ching Fan-Chiang
National TsingHua University
Department of Interdisciplinary
Program of Engineering

***Abstract*—Captcha are common and essential to differentiate between bot and man. Sometimes people find this a hassle when trying to get access to specific services. They often turn to auto-filler mechanism to skip this step. In this paper, we will discuss some methods in protecting the functionality of the conventional captcha by lowering the accuracy of the auto-filler prediction models. Finally, the accuracy's result will also be shown.**

Keywords—adversarial attack, captcha

I. INTRODUCTION

Adversarial in machine learning is a study of attack on machine learning algorithms, and of the defenses against such attacks. In this paper, we are focusing on some methods to perform an attack on the input data of some machine learning algorithms based on the similar traits of the model to generate a new version of captcha as output. We will do this adversarial attack by trying to hamper the performance of the classifiers by providing the models with false data.

We will build some models that give a good classification accuracy and perform both black-box and white-box adversarial attacks using the models. We would see how well the method we used and implemented perform in decreasing the prediction accuracy.

A. Input Dataset

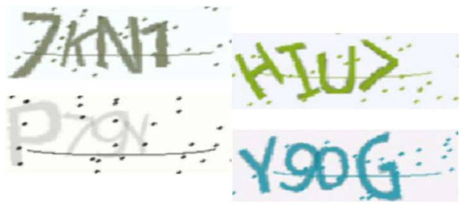


Figure 1. Captcha samples.

PYPI captcha package is used to generate the input data as the training and testing dataset with random key generation.

B. Captcha Generator

We implemented a function that takes a string of 4 characters as input and generates an attacked version of the captcha by the captcha generator module. The objective is to have the captcha's answer wrongly predicted when the attacked version of captcha is given instead.

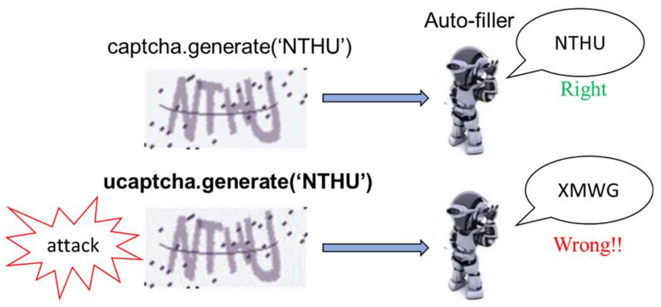


Figure 2. Attacked captcha image generator to confuse bot.

II. METHODS

A. Classification model

We implemented 3 captcha classification models as our proxy models. Two of them have pretty high accuracy but the rather inaccurate one is still useful for our experiment.

Model A(accuracy 96%) : refer to captcha_break from GitHub[3], VGG-like architecture.

Model B(accuracy 93%) : refer to Model A.

Model C(accuracy 70%) : refer to CAPTCHA : 2 solutions to break them [>99%] from Kaggle which is original for black-and-white graph[5].

We calculated model accuracy with the overall accuracy, only every character in one captcha is predicted correct this captcha is considered correct one.

B. Adversarial attack

We have implemented 3 adversarial attack methods in this project.

The first is Fast Gradient Sign method (i.e. "FGSM").The biggest feature of FGSM is that it only updates the parameters once rather than for many time. With only one time update, it can find an image which can attack successfully. As for the parameter 'epsilon' is the disturbance rate for the attack.

$$X^{adv} = X + \epsilon\,\mathrm{sign}(\nabla_X J(X, y_{true}))$$

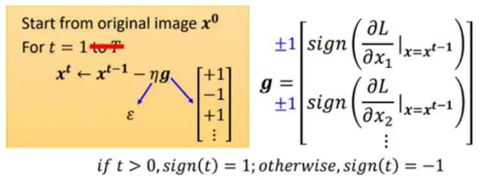

Start from original image $\boldsymbol{x}^0$
For $t = 1$ ~~to T~~
$\boldsymbol{x}^t \leftarrow \boldsymbol{x}^{t-1} - \eta \boldsymbol{g}$
ε
$\begin{bmatrix} +1 \\ -1 \\ +1 \\ \vdots \end{bmatrix}$
$\boldsymbol{g} = \begin{bmatrix} sign\left(\frac{\partial L}{\partial x_1}|_{x=x^{t-1}}\right) \\ sign\left(\frac{\partial L}{\partial x_2}|_{x=x^{t-1}}\right) \\ \vdots \end{bmatrix}$
±1
±1
$if\ t > 0, sign(t) = 1; otherwise, sign(t) = -1$

Figure 3. FGSM Gradient descent function

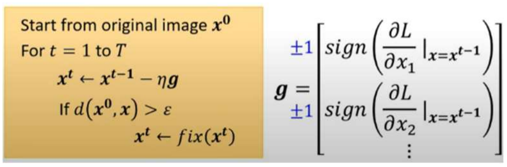
Another attack method is I-FGSM which I stands for iterative. I-FGSM does N times of iterations. And for every iteration, it times alpha which is epsilon divided by numbers of iterations N.

$$X_0^{adv} = X, \quad X_{N+1}^{adv} = Clip_{X,\epsilon}\left\{X_N^{adv} + \alpha\,\mathrm{sign}(\nabla_X J(X_N^{adv}, y_{true}))\right\}$$

e = ε * 256

N = *min*(e+4, 1.25e)

α = ε / N

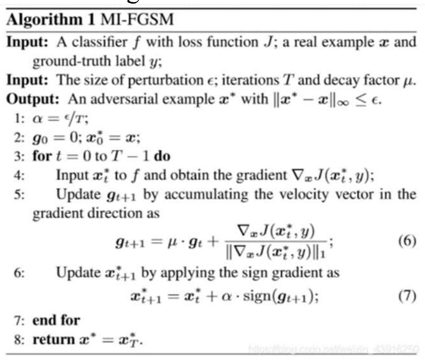


```
Start from original image $x^0$
For $t = 1$ to $T$
    $x^t \leftarrow x^{t-1} - \eta g$
    If $d(x^0, x) > \varepsilon$
        $x^t \leftarrow fix(x^t)$
```

$$g = \begin{bmatrix} \pm 1 \; sign\left(\frac{\partial L}{\partial x_1}|_{x=x^{t-1}}\right) \\ \pm 1 \; sign\left(\frac{\partial L}{\partial x_2}|_{x=x^{t-1}}\right) \\ \vdots \end{bmatrix}$$

Figure 4. I-FGSM Gradient descent function

The last attack method is call MI-FGSM which m and I stands for momentum and iterative. The concept of momentum here upgrades the gradient g by accumulating the velocity vector in the gradient direction.



```
Algorithm 1 MI-FGSM
Input: A classifier $f$ with loss function $J$; a real example $x$ and ground-truth label $y$;
Input: The size of perturbation $\epsilon$; iterations $T$ and decay factor $\mu$.
Output: An adversarial example $x^*$ with $\|x^* - x\|_\infty \le \epsilon$.
1: $\alpha = \epsilon / T$;
2: $g_0 = 0$; $x_0^* = x$;
3: for $t = 0$ to $T - 1$ do
4:     Input $x_t^*$ to $f$ and obtain the gradient $\nabla_x J(x_t^*, y)$;
5:     Update $g_{t+1}$ by accumulating the velocity vector in the gradient direction as
           $g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J(x_t^*, y)}{\|\nabla_x J(x_t^*, y)\|_1}$;     (6)
6:     Update $x_{t+1}^*$ by applying the sign gradient as
           $x_{t+1}^* = x_t^* + \alpha \cdot \mathrm{sign}(g_{t+1})$;     (7)
7: end for
8: return $x^* = x_T^*$.
```

Figure 5. MI-FGSM Gradient descent function

Additionally, we continually attack the same image until every character in CAPTCHA predicts incorrect or the number of iterations exceeds the iterative threshold.

```
Algorithm 1 Iterative Attack Model
  Given proxy model $M_p(X)$, attack method $A(X)$, Iterative threshold $B_T$
  $X' \leftarrow X$
  $B \leftarrow 0$
  while not complete attack and $B \le B_T$ do
      $X' \leftarrow A(X')$
      $B \leftarrow B + 1$
  end while
  return $X'$
```

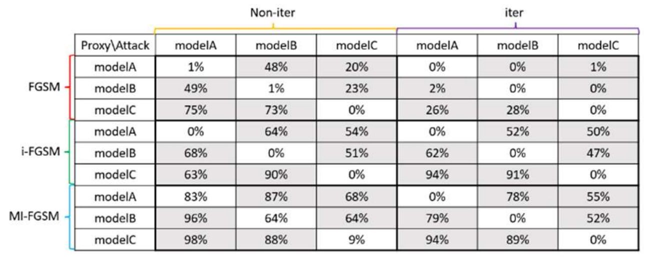
Figure 6. Iterative Attack Model

III. Results

The original accuracy of the three models was 96% for model A, 90% for model B, and 70% for model C, and our experiment tests their accuracy after the black-box or white-box adversarial attack.

In Table 1, we test three different attack methods and the iterative attack model on all of the classification models by setting perturbation constants ε=16/256. For white-box attack, four out of six white-box attack methods (*iter*-FGSM, i-FGSM, *iter*-i-FGSM and *iter*-MI-FGSM) can decrease classification accuracy down to 0% in all of the attacked models, and rest of them also can decrease the accuracy from 0% to 83%. And for black-box attack, *iter*-FGSM attack method performs the best, which can decrease classification accuracy down to 0%~28%.

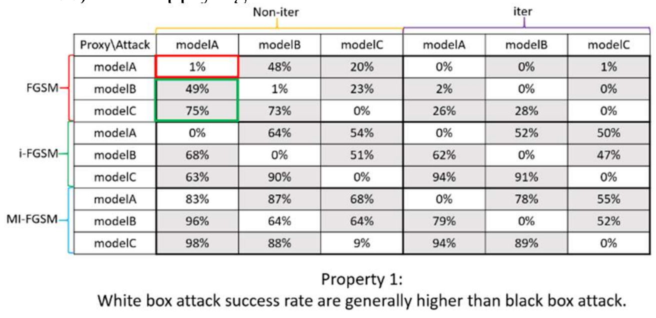
.



		Non-iter			iter		
	Proxy\Attack	modelA	modelB	modelC	modelA	modelB	modelC
FGSM	modelA	1%	48%	20%	0%	0%	1%
	modelB	49%	1%	23%	2%	0%	0%
	modelC	75%	73%	0%	26%	28%	0%
i-FGSM	modelA	0%	64%	54%	0%	52%	50%
	modelB	68%	0%	51%	62%	0%	47%
	modelC	63%	90%	0%	94%	91%	0%
MI-FGSM	modelA	83%	87%	68%	0%	78%	55%
	modelB	96%	64%	64%	79%	0%	52%
	modelC	98%	88%	9%	94%	89%	0%

Table 1. White-box and Black-box attack experiment results by setting ε =16/256.

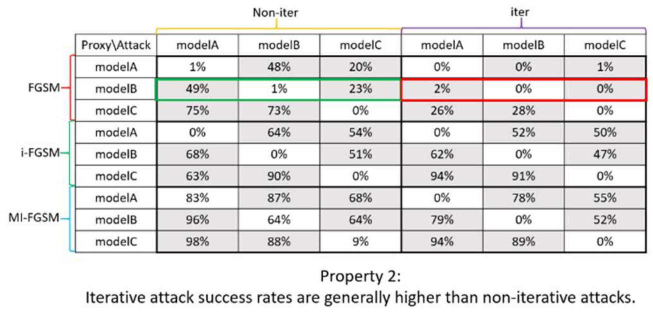
From the above results, we also conclude three properties: (1) White-box attack success rates are generally higher than black-box attacks (Table 2). Take the FGSM attack method as an example, The accuracy of the white-box attack is 1% while the accuracy of black-box attacks is 49% and 75% respectively, (2) Using the iterative attack model generally increases the attack success rate (Table 3). By observing results that using model B as a proxy model, the FGSM attack method has accuracy of 49%, 1%, and 23%, and the iter-FGSM attack method has 2%, 0%, 0%, (3) Using the weaker model as the proxy model decreases the attack success rate (Table 4), the result shows that using model B as the proxy model can attack better (accuracy from 90% down to 2%) than using model C as the proxy model (accuracy from 90% down to 26%) when applying iter-FGSM attack method on model A.



		Non-iter			iter		
	Proxy\Attack	modelA	modelB	modelC	modelA	modelB	modelC
FGSM	modelA	1%	48%	20%	0%	0%	1%
	modelB	49%	1%	23%	2%	0%	0%
	modelC	75%	73%	0%	26%	28%	0%
i-FGSM	modelA	0%	64%	54%	0%	52%	50%
	modelB	68%	0%	51%	62%	0%	47%
	modelC	63%	90%	0%	94%	91%	0%
MI-FGSM	modelA	83%	87%	68%	0%	78%	55%
	modelB	96%	64%	64%	79%	0%	52%
	modelC	98%	88%	9%	94%	89%	0%

Property 1:
White box attack success rate are generally higher than black box attack.

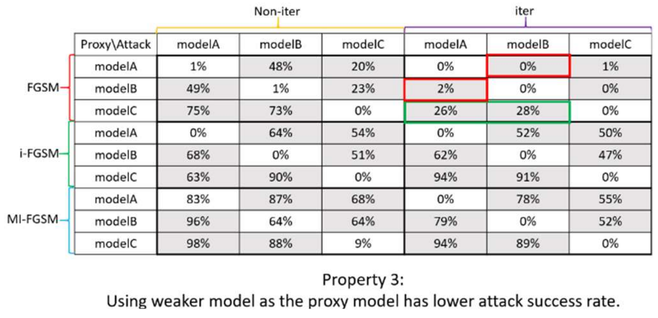
Table 2. Property (1): White-box attack success rates are generally higher than black-box attacks.



		Non-iter			iter		
	Proxy\Attack	modelA	modelB	modelC	modelA	modelB	modelC
FGSM	modelA	1%	48%	20%	0%	0%	1%
	modelB	49%	1%	23%	2%	0%	0%
	modelC	75%	73%	0%	26%	28%	0%
i-FGSM	modelA	0%	64%	54%	0%	52%	50%
	modelB	68%	0%	51%	62%	0%	47%
	modelC	63%	90%	0%	94%	91%	0%
MI-FGSM	modelA	83%	87%	68%	0%	78%	55%
	modelB	96%	64%	64%	79%	0%	52%
	modelC	98%	88%	9%	94%	89%	0%

Property 2:
Iterative attack success rates are generally higher than non-iterative attacks.

Table 3. Property (2): Iterative attack success rates are generally higher than non-iterative attacks.



		Non-iter			iter		
	Proxy\Attack	modelA	modelB	modelC	modelA	modelB	modelC
FGSM	modelA	1%	48%	20%	0%	0%	1%
	modelB	49%	1%	23%	2%	0%	0%
	modelC	75%	73%	0%	26%	28%	0%
i-FGSM	modelA	0%	64%	54%	0%	52%	50%
	modelB	68%	0%	51%	62%	0%	47%
	modelC	63%	90%	0%	94%	91%	0%
MI-FGSM	modelA	83%	87%	68%	0%	78%	55%
	modelB	96%	64%	64%	79%	0%	52%
	modelC	98%	88%	9%	94%	89%	0%

Property 3:
Using weaker model as the proxy model has lower attack success rate.

Table 4. Property (3): Using weaker model as the proxy model has lower attack success rate.

We also test attack performance by shrinking perturbation constants ε to 4/256, and the results are shown at Table 5.

		Non-iter			iter		
	Proxy\Attack	modelA	modelB	modelC	modelA	modelB	modelC
FGSM	modelA	28%	82%	59%	0%	37%	28%
	modelB	87%	21%	57%	31%	0%	31%
	modelC	100%	91%	0%	88%	83%	0%
i-FGSM	modelA	2%	85%	57%	0%	77%	51%
	modelB	83%	0%	61%	61%	0%	44%
	modelC	95%	89%	0%	95%	83%	0%
MI-FGSM	modelA	86%	92%	64%	0%	79%	52%
	modelB	93%	67%	65%	76%	0%	60%
	modelC	97%	85%	13%	94%	89%	0%

Table 5. White-box and Black-box attack experiment results by setting ε =4/256.

We also compare the performance of each attack model in a different situation, and the results are shown in Table 6. In the situation of not applying the iterative attack model, i-FGSM performs better when the perturbation constant is smaller, while FGSM decreases the accuracy the most when ε is set to be bigger. While the iterative attack method is applied, all of the methods can decrease classification accuracy down to 0% using the white-box attack for each classification model, and *iter*-FGSM decreases the accuracy the most when applying the black-box attack.

ε = 4/256

	Non-iter		iter	
Method\Target	White Box	Black Box	White Box	Black Box
FGSM			✓	✓
iFGSM	✓	✓	✓	
MI-FGSM			✓	

ε = 16/256

	Non-iter		iter	
Method\Target	White Box	Black Box	White Box	Black Box
FGSM		✓	✓	✓
iFGSM	✓		✓	
MI-FGSM			✓	

Table 6. Compare the performance of each attack model in different situations.

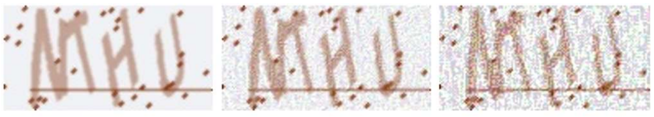
IV. Discussion

As for summary, this paper has made the following observations:

Method	Pros	Cons
FGSM	• Higher transferability	• More perturbations
iFGSM	• Less perturbations	• Lower transferability
Iterative attack	• Better attack effect	• Slower speed

Table 7. Summary of the pros and cons between FGSM, iFGSM and Iterative attack.

The above table shows that FGSM has a higher attack success rate on black box attack but also applies more intense disturbance on the image. iFGSM generates cleaner images but it lacks transferability compared to FGSM. The Iterative attack model has a better effect but costs more time.



(a) ε =4/256 (b) ε =16/256 (c) ε =64/256

Figure 7. The example attacked images generated by FGSM: (a) ε =4/256; (b) ε =16/256; (c) ε =64/256.

V. Conclusion

The best black box attack performance was achieved using the FGSM method. The prediction result on the attacked version of the captcha can go as low as around 30% from the initial prediction accuracy in range 70%-96% if the resulting image has less perturbation after the attack. If more perturbation were to be allowed on the image, we can make the prediction accuracy to be as low as 2% or less.

The current result is not the end of it. Further improvement is possible by using other attack methods such as BIM. It may also be interesting to perform the attack on ensemble model, and possibly perform some sort of defense mechanism to defend against the existing attack method including the one discussed in this paper.

Author Contribution Statements

C.Y.Wu (25%): conceive and design the study, system architecture development, model A DNN architecture implement, iterative attack model implements, data interpretation, a co-worker of figure design, a co-worker of the final presentation.

J.Y.Tien (21%): system architecture development, FGSM, iFGSM, MI-FGSM attack model implement, data collection and summary, a co-worker of figure design, a co-worker of the final presentation.

T.L.Ling (18%): model C DNN implementation, captcha generator implementation, a co-worker of the final presentation.

C.Fan-Chiang (18%): model C DNN implementation, customized captcha generator implementation, a co-worker of the final presentation.

C.W.Huang (18%): model B DNN implementation, attacked captcha generator implementation, co-worker of final presentation.

References

[1] Hung-yi Lee, 來自人類的惡意攻擊 (Adversarial Attack) (上) - 基本概念, https://www.youtube.com/watch?v=xGQKhbjrFRk, 2021.

[2] Hung-yi Lee, 來自人類的惡意攻擊 (Adversarial Attack) (下) - 基本概念, https://www.youtube.com/watch?v=z-Q9ia5H2Ig, 2021.

[3] Yang Peiwen, captcha_break, github.com/ypwhs/captcha_break, 2019.

[4] maomao,captcha-adversarial-attack, github.com/zhangbincheng1997/captcha-adversarial-attack, 2019.

[5] CAPTCHA : 2 solutions to break them [>99%], https://www.kaggle.com/code/arnrob/captcha-2-solutions-to-break-them-99