中国科技核心期刊
ISSN 1009-671X
CN 23-1191/U

基于对抗样本生成的验证码反爬虫机制研究

马军，王效武，朱永川，王海兮

引用本文:
马军, 王效武, 朱永川, 等. 基于对抗样本生成的验证码反爬虫机制研究[J]. 应用科技, 2021, 48(6): 45–50.
MA Jun, WANG Xiaowu, ZHU Yongchuan, et al. Study on the verification code anti–crawler mechanism based on the generation of adversarial samples[J]. *Applied science and technology*, 2021, 48(6): 45–50.

在线阅读 View online: https://dx.doi.org/10.11991/yykj.202103019

DOI: 10.11991/yykj.202103019
网络出版地址: https://kns.cnki.net/kcms/detail/23.1191.U.20210809.1303.002.html

基于对抗样本生成的验证码反爬虫机制研究

马军[1]，王效武[2]，朱永川[1]，王海兮[2]

1. 深圳市网联安瑞网络科技有限公司，广东 深圳 518000
2. 中国电子科技集团公司第三十研究所，四川 成都 610041

摘 要: 为提升网站验证码的安全性，提出基于对抗样本生成的验证码反爬虫机制。本文通过在对抗样本数据集中添加极小的扰动，可导致验证码识别模型输出错误的预测结果，从而无法绕过反爬机制对网络数据进行非法下载。针对常用的验证码识别模型，本文对比了使用图像加扰和未使用图像加扰情况下的文本验证码识别效果。结果表明，采用本文提出的图像加扰算法，可大幅度降低图像识别模型的识别精度，从而有效保护网站验证码反爬机制的可靠性。基于本文提出的图像加扰验证码技术，可作为互联网反爬虫机制的重要手段。

关键词: 验证码识别；字符分割；深度神经网络；对抗样本；图像扰动；图像识别；深度学习；人工智能
中图分类号: TP302.1 **文献标志码:** A **文章编号:** 1009−671X(2021)06−0045−06

Study on the verification code anti-crawler mechanism based on the generation of adversarial samples

MA Jun[1], WANG Xiaowu[2], ZHU Yongchuan[1], WANG Haixi[2]

1. Shenzhen CyberAray Technology Corporation, Shenzhen 518000, China
2. The 30th Institute of China Electronics Technology Corporation, Chengdu 610041, China

Abstract: In order to improve the security of website verification codes, a verification code anti-crawler mechanism based on adversarial sample generation is proposed. In this paper, by adding very small disturbance to the adversarial sample data set, incorrect prediction results can be output by the verification code recognition model, which refrains people from illegally download network data by bypassing the anti-crawl mechanism. Aiming at the commonly used verification code recognition model, this paper compares the recognition effect of text verification code in the cases of using image scrambling and not using image scrambling. The results show that the image scrambling algorithm proposed in this paper can greatly reduce the recognition accuracy of the image recognition model, thereby effectively protecting the reliability of the website verification code anti-climbing mechanism. Image scrambling verification codes based on this paper can be used as an important means of practicing Internet anti-crawler mechanism.

Keywords: verification code recognition; character segmentation; deep neural network; adversarial samples; image disturbance; image recognition; deep learning; artificial intelligence

近年来，互联网信息呈现出指数级增长的趋势，而数据对于商家则至关重要。通过海量互联网数据分析，不仅能降低企业的运营成本，还能提前预估产品的需求，进而降低生产和库存成本。然而，用户对海量互联网数据的采集，使得互联网服务器承受巨大的压力；尤其是大量网络爬虫机器人的存在，造成数据采集的频次和速度超过人类的浏览速度，影响网络服务器的正常运行[1]。因此，大量网站都采用了多种反爬虫的机制，验证码识别则是众多反爬机制中的一种[2]。

对于普通的验证码，部分人工智能的技术能自动识别验证码，突破验证码识别机制，并配合网络爬虫实现全自动化的信息采集。但人工智能技术不是无懈可击，其自身也存在潜在的漏洞[3−5]。为了确保验证码图片不被基于智能算法的机器人识别，利用人工智能技术的弱点，构造出基于图像扰动技术的扰动样本，这些样本能有效降低机器人的识别准确度，使网络机器人自动识别验证码的功能失效，从而提高系统安全性。

收稿日期: 2021−03−16. **网络出版日期:** 2021−08−09.
作者简介: 马军，男，高级工程师.
通信作者: 马军，E-mail: majun_98301@163.com.

对抗样本可作为机器学习或深度学习模型的输入，最终导致模型得出错误的分类结果[6−7]。其设计的基本思想是攻击者在原始图片中添加较小的扰动，就能使模型结果产生截然不同的错误判断，这些扰动被称为对抗样本。例如，在原版大熊猫图片中加入肉眼难以发现的干扰，生成对抗样本，最终导致神经网络误认为大熊猫图片是长臂猿，其置信度在 99.3%[6]。除此之外，Szegedy 等[4] 还发现，对抗样本具有一定的跨模型和跨数据集泛化性；前者指大部分对抗样本会被具有不同参数 (模型层数、正则化、权重等) 的网络模型误分类，后者指大部分对抗样本会被在不同训练集上被误分类。

因此，本文拟通过向文本验证码添加较小的像素扰动进而生成对抗验证码，从而使得这些对抗验证码不容易被以深度神经网络算法为核心的智能机器人正确识别，进而提高系统的安全性。

1　验证码相关技术研究

目前，验证码可分成 3 种类型：第 1 种是基于文本的验证码，通常由字母和数字组成，其经过复杂的图像扭曲技术处理，不能被自动识别技术所识别，但是肉眼能正常识别；第 2 种是基于图像的验证码，通常要求用户在一组候选图片中选择一张或多张具有特定语义的图片；第 3 种是基于语音的验证码，通常要求用户完成语音识别任务，并常与文本验证码一起使用。

本文主要基于文本的验证码进行研究，原因如下：1) 基于文本的验证码是最广泛使用的一种验证码类型，许多主流网站都在使用，比如 Google、Baidu、Yahoo；2) 文本验证码由来已久且已被广泛使用，相关的攻击手段层出不穷，且攻击技术也相对成熟，因此文本验证码系统的安全性常受到较大威胁；3) 文本验证码作为最基础的形式，其相关研究能较容易应用到其他类型的验证码中。

文本验证码的来源存在争议，2 个研究团队都声称发明了当今互联网广泛使用的验证码。以 M.D.Lillibridge 为首的研究团队人员声称 1997 年在 AltaVista 上使用验证码，防止机器人在其网络搜索引擎中添加统一资源定位器 (uniform resource locators, URLs)[8]。卡内基梅隆大学的一个研究团队在 2003 年发布的一篇论文也提出验证码这一概念，随后被学术界和工业界广泛使用，并成为追捧的研究热点[9]。Chellapilla 等[9] 研究了早期文本验证码的安全性，并且设计出了一种基于机器学习能破解验证码的方法。为了增强文本验证码的安全性，Yan 等[10] 建议将验证码字符挤在一起，由此提出了一种名为字符粘连 (crowding characters together, CCT) 的验证码生成机制。除此之外，还有一些技术用于增强文本验证码的安全性，比如噪声弧、扭曲、旋转、重叠、两层结构等。然而这些安全机制并不绝对安全，较容易被攻破[11−12]。文本验证码的攻防就像矛与盾的问题，针对文本验证码的攻击层出不穷，因此关于提升文本验证码鲁棒性的研究还在不断深入。

2　验证码训练样本采集

验证码数据采集系统针对特定互联网目标网站进行采集，采集数据经过预处理后入库，用以训练验证码。

本系统采用典型的浏览器/服务器模式 (browser/server, B/S) 架构来实现，不同的客户端程序通过 IE/Chrome 浏览器共同访问 Web 服务器的前端页面，访问 Redis 数据库、MongoDB 数据库、Mysql 数据库服务器进行数据存取。系统结构如图 1 所示。

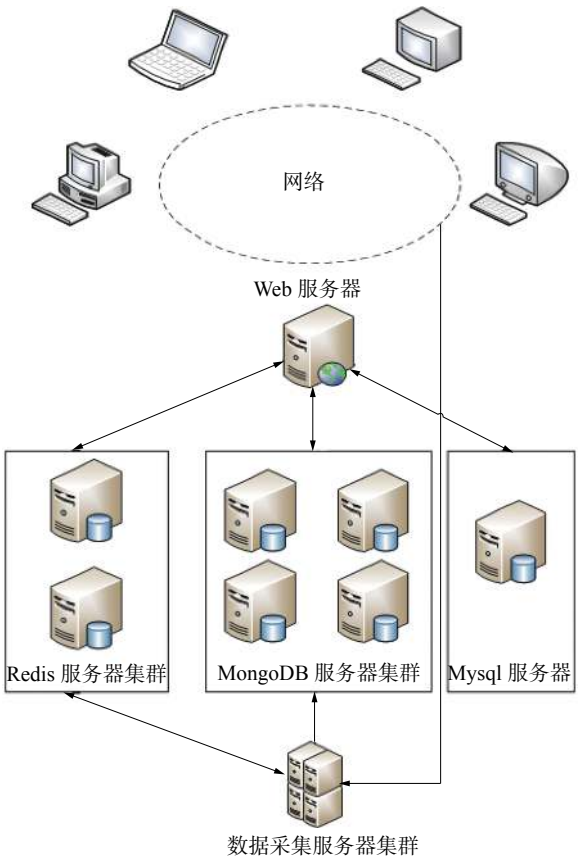

图1　系统结构 (B/S)

系统设计说明如下:

1) 每个客户终端计算机都不需要额外安装客户程序, 但是需要有浏览器支持。

2) Web 服务器访问各个数据库集群, 数据库对外部不可见。

3) 数据采集服务器访问各个目标网站, 获取数据并写入 Redis 和 MongoDB 数据库服务器, 数据库对外不可见。

4) MySQL 服务器用于存储相对繁琐仔细的业务代码部分, 数据库对外不可见。MySQL 服务器在足够硬件环境的支持下能够适用, 并且相对合理。

5) 多用户并发访问和处理、数据锁和事务协调由数据库和后端共同来完成, 不设计单独的事务处理服务器。

应用程序的开发设计采用 Java 进行实现, 前端通过 JavaScript 设计并实现, 数据采集部分采用 Python 实现, 所有的逻辑实现、算法和脚本在服务器端编译开发或解释。系统规划的结果如图 2 所示。

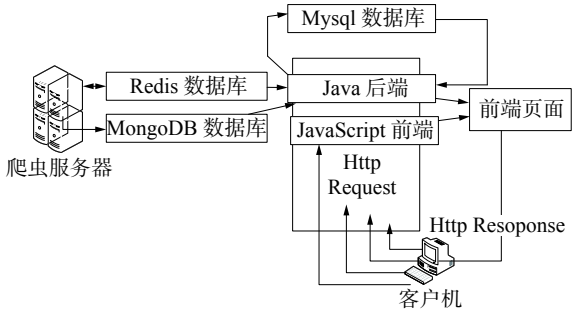

图2　系统逻辑结构

系统运行控制将严格按照各模块间函数调用关系来实现。在各事务中心模块中, 需对运行控制进行正确的判断, 选择正确的运行控制路径。在网络传输方面, 客户机在发送数据后, 将等待服务器的确认收到信号, 收到后, 再次等待服务器发送回答数据, 然后对数据进行确认。服务器在接到数据后发送确认信号, 在对数据处理、访问数据库后, 将返回信息送回客户机, 并等待确认。运行控制方法的具体模块包括:

1) 登录模块: 前端模块接收到账号密码登录请求, 对账号密码进行加密, 通过时间戳以及加盐等方式进行加密, 传输到后端, 后端从 redis 中获取到相应的随机字串, 进行加密验证, 若相同, 则进行允许登录跳转。

2) 采集模块: Java 后端守护进程模块告知 Python 后端守护进程模块开始进行爬虫, Python 守护进程调用 Scrapy 模块根据要求进行爬虫, 并将数据写入 MongoDB 和 Redis 数据库。

3) 用户信息管理模块: 前端模块接受到用户信息的修改要求, 并把修改后的信息发送给 Java 后端, Java 后端修改 Mysql 数据库中的用户信息。

4) 关键词管理: 前端模块展示关键词库变更过程, Java 后端管理模块通过调用 Mysql 数据库模块从而获取数据传送给前端。

5) 信息添加: 前端模块传输添加信息的内容到 Java 后端管理模块, 后端模块通过 json 拼接将数据写入 MongoDB 数据库模块。

6) 信息删除: 前端模块传输相应信息的关键密钥到 Java 后端管理模块, 控制 MongoDB 数据库模块删除相关信息。

7) 信息修改: 前端模块传输修改后的信息到 Java 后端管理模块, 控制 MongoDB 数据库模块修改相关信息, 并将更改后的信息通过后端返回给前端。

8) 信息排序: 前端模块传输排序请求到 Java 后端管理模块, 控制 MongoDB 数据库模块修改相关信息, 并将更改后的信息通过后端返回给前端。

9) 采集开关: Java 后端守护进程模块作为客户端, Python 后端守护进程模块作为服务端, 通过多线程的方法创建多个客户端实例, 同时向多个 Python 服务端发送请求。

3　对抗验证码设计模型

对抗样本源于计算机视觉领域, 其主要的目的是产生图像扰动。最初由 Kurakin 等[6] 提出, 并就扰动的原理进行了详细的阐述。

根据攻击者所能获取的信息, 攻击可分为白盒攻击和黑盒攻击。在白盒攻击情境中, 攻击者能获取目标神经网络的所有信息, 包括网络的参数和架构、梯度以及数据集等; 攻击者利用这些先验信息构造对抗样本[3,6−7,13]。在黑盒攻击的情境中, 攻击者无法获取神经网络模型的内部信息以及训练集的分布, 只能启用替代模型, 这个替代模型首先随机生成样本输入到判别模型中, 获取对应的输出, 并观察输入和输出的关系, 试图让替代模型找到与判别模型几乎一样的决策边界, 并通过不断地迭代合成对抗样本, 最终成功地让深度神经网络 (deep neural networks, DNN) 分类器错分了 84.24% 的对抗样本, 让 Logistic 回归

分类器错分了 96.19% 的对抗样本[14]。

由于白盒攻击获得的先验信息较多，这意味着攻击者能获取到明确的目标函数，生成模型能更加高效地制造出对抗样本。而黑盒攻击过程中，最初只能随机生成样本，输入到判别模型中，此时的判别模型基本上是个黑匣子，攻击者只能构造一个替代模型，通过观察输入和输出的关系，试图让替代模型的决策边界无限接近判别模型的决策边界，由此合成出对抗样本。因此，综上所述，白盒攻击的成功率自然会比黑盒攻击高。

本文旨在实现针对文本验证码生成对抗样本。由于白盒攻击的成功率较高，因此选取了 2 种经典的白盒攻击算法作为基准测试组，包括快速梯度符号法 (fast gradient sign method, FGSM) 和基础迭代法 (basic iterative method, BIM)，并将以上 2 种经典方法与空间转换 (spatially transformed Adversarial, stAdv) 算法进行对比。

3.1　基准测试算法

Kurakin 等[6] 提出了一种名为 FGSM 的生成对抗样本的方法，这种方法简单易行，只需要攻击一次就可达成无目标攻击的效果。该方法能很自然地迁移到文本对抗验证码生成的任务中，公式为

$$x_{\text{adv}} = x + \varepsilon \cdot \text{sign}(\nabla_x L_{\text{F,label}}(x)) \tag{1}$$

式中：参数ε为扰动的程度，$L_{\text{F,label}}(x)$为用来训练模型的损失函数。除此之外，式 (1) 隐含满足以下条件：x_{adv}与 x 之间的距离ε应该设置比较小的数值，否则对图像的扰动就很容易被识破。

Kurakin 等[6] 提出了基于 FSGM 提出多次迭代的 FSGM 的改良方法（即 BIM），这种方法的思想是将 FGSM 算法以较小的步长运行多次，并且在每步之后对像素值的中间结果进行裁剪，以确保它们在原始图像的ε值域内。其核心公式为

$$x_{\text{adv}} = x + \text{Clip}_{\varepsilon}(\eta \cdot \text{sign}(\nabla_x L_{\text{F,label}}(x))) \tag{2}$$

式 (2) 与式 (1) 最大的区别在于，前者较后者增加了 Clip 函数，该函数是像素值裁剪函数，确保扰动后的像素值在−1~1。

在实际部署时，我们对原有代码做了如下修改，以适应文本验证码的图片。$x(h,w,c)$函数确保扰动后的像素值$x'(h,w,c)$在$[x(h,w,c)-\varepsilon, x(h,w,c)+\varepsilon]$，同时不能超过上界+1 或下界−1。其中，$x(h,w,c)$表示验证码图片$x$在通道$c$位置$(h,w)$处的像素值，$x'(h,w,c)$表示在此处扰动后的像素值；步长$\eta$通常设置为一个较小的值 ($\eta \leqslant \varepsilon$)，通过将最大迭代次数 i 设置为一个合理的值，使得扰动后的像素值能抵达扰动边界$x(h,w,c)-\varepsilon$和$x(h,w,c)+\varepsilon$之间。

3.2　基准测试算法

基准测试中生成对抗文本验证码的算法都采用直接修改原始图片像素值的方式，因此较容易受到图片预处理技术的影响[7]。因此，本文试图找到一种不易受到预处理技术影响的生成对抗样本的算法。通过文献 [7] 发现，spatially transformed 的优化方法较适合我们的应用场景。因此我们基于 spatially transformed 的优化方法 (StAdv) 进行改进，并称其为改进型 StAdv，从而能一定程度上缓解以上提及的预处理的影响问题。

基于传统 StAdv 优化方法做了如下 2 点改进：1) 引入了一个整数型变量 T 作为衡量干扰程度的变量，其取值在 0~n−1，其值越小干扰程度越大。若 T 被设置为 0，则验证码图像经过空间变换后，将被分类器预测为最不可能的结果，图像的前后变化差异也会较大；若 T 被设置为 n−1，则不会对分类器造成任何干扰。2) 为了优化生成算法的速度以适应本文的应用场景，我们并没有使用原文的 L-BFGS 优化器，而使用 Adam 优化器最小化目标函数。改进型 StAdv 生成算法的伪代码如下。

Input x, C, Z, y_{true}, i,
Output x_{adv}
Set $x_{\text{adv}} = x$
Set target = $C(x_{\text{adv}})$
Set $f = 0$
for $i = 1; i <= q; i++$ do
　target$[i]$ = argsort$Z(x_{\text{adv}})^i[T]$
　end for
　if $C(x_{\text{adv}}) \neq y_{\text{true}}$ *and* $C(x_{\text{adv}})$ == target
　return x_{adv}
end if
　based on flow field f and equation (2), generate the base adversarial text-based CAPTCHAs x_{adv}
while $C(x_{\text{adv}}) \neq$ target and i>0 do
　logits = $Z(x_{\text{adv}})$
　$\mathcal{L}_{\text{adv}} = \mathcal{L}_{\text{adv}}(\text{logits}, \text{target}, \kappa)$
　$\mathcal{L}_{\text{flow}} = \mathcal{L}_{\text{flow}}(f)$
　loss = $\mathcal{L}_{\text{adv}} + \tau\mathcal{L}_{\text{flow}}$
　use Adam optimizer with the learning rate l_{r} to minimize the loss;
　based on the updated flow field f and equation (2), generate the updated adversarial text-based CAPTCH
　As x_{adv};
　i − −
end while
return x_{adv}

代码中, x 为原始的文本验证码图像; C 为预先训练的 CNN 分类模型; Z 为模型最后一层的输出; y_{true} 为验证码图像的真实标签; i 为最大迭代次数; τ 为平衡系数, 用以平衡 L_{adv} 和 L_{flow}; k 为验证码图像被误分类为其他类别的置信度; l_{r} 和 T 分别是学习率和干扰程度。

4　实验结果分析

为比较不同图像加扰算法的效果, 使用 LeNet、ResNet、DenseNet、Wide ResNet 等 4 种典型的卷积神经网络模型作为入侵学习模型, 构造用户身份验证图像[15]。此外, 使用 FGSM、BIM、DeepFool、JSMA 等 4 种图像扰动算法作为本文所提改进 StAdv 算法的对比算法, 对上述图像扰动算法的反入侵能力进行比较。仿真实验采用常用目标网站的验证码图像数据集 (详见图 3), 随机选择 100 000 张图像作为训练集, 20 000 张图像作为测试验证集, 实验完成基于该数据集的图像识别任务。

图3　验证码图像集

如图 4 所示, 在无图像干扰时, 非法入侵者通过使用 LeNet、ResNet、DenseNet、Wide ResNet 等典型模型构造验证图像时, 系统认证的通过率分别可达到 87.3%、90.2%、91.1% 和 90.1%。

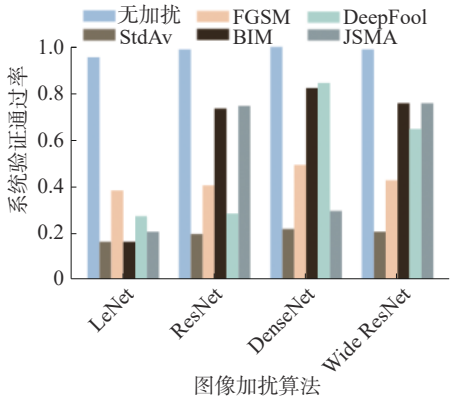

图4　不同图像加扰算法对应的干扰效果

作为本文所提改进 StAdv 算法的对比算法, 实验中考虑使用 FGSM、BIM、DeepFool、JSMA 等图像扰动算法。对比结果显示, FGSM 算法可以将入侵模型所伪造图像的认证通过率降低至 30%~40%; BIM、DeepFool、JSMA 等算法在防御入侵模型时不太稳定, 构造图像的认证通过率在 20%~70% 浮动; 本文所提 StAdv 算法可以将入侵模型构造图像的认证通过率下降至 20% 以下。因此, 基于 StAdv 图像扰动算法的入侵防御模型, 其干扰效果和健壮性明显优于其他 4 种扰动算法。

5　结论

本文研究基于文本验证码图片的对抗扰动方法, 以提升反爬模型的健壮性和可用性。为比较不同图像加扰算法的效果, 使用 LeNet、ResNet、DenseNet、Wide ResNet 等 4 种典型的卷积神经网络模型作为入侵学习模型, 构造用户身份验证图像。此外, 使用 FGSM、BIM、DeepFool、JSMA 等 4 种图像扰动算法作为所提改进 StAdv 算法的对比算法, 对上述图像扰动算法的反入侵能力进行比较。结果表明, 基于 StAdv 图像扰动算法的入侵防御模型, 其干扰效果和健壮性明显优于其他 4 种扰动算法。

参考文献:

[1] YAN J, EL AHMAD A S. Usability of CAPTCHAs or usability issues in CAPTCHA design[C]//Proceedings of the 4th Symposium on Usable Privacy and Security. Pittsburgh, USA, 2008: 44–52.

[2] YIN Yingjie, XU De, WANG Xingang, et al. Adversarial feature sampling learning for efficient visual tracking[J]. IEEE transactions on automation science and engineering, 2020, 17(2): 847–857.

[3] ROY D, MUKHERJEE T, CHATTERJEE M, et al. RFAL: adversarial learning for RF transmitter identification and classification[J]. IEEE transactions on cognitive communications and networking, 2020, 6(2): 783–801.

[4] SZEGEDY C, ZAREMBA W, SUTSKEVER I, et al. Intriguing properties of neural networks[EB/OL]. arXiv preprint arXiv: 1312.6199, 2014.

[5] GOODFELLOW I J, SHLENS J, SZEGEDY C. Explaining and harnessing adversarial examples[EB/OL]. arXiv preprint arXiv: 1412.6572, 2015.

[6] KURAKIN A, GOODFELLOW I, BENGIO S. Adversarial examples in the physical world[EB/OL]. arXiv preprint arXiv: 1607.02533, 2017.

[7] WPA C, RLB C, RWA C, et al. EnsembleFool: A method to generate adversarial examples based on model fusion strategy[J]. Computers & security, 2021, 107: 102317.

[8] VON AHN L, BLUM M, HOPPER N J, et al. CAPTCHA: using hard AI problems for security[C]//EUROCRYPT: International Conference on the Theory and Applications of Cryptographic Techniques. Warsaw, Poland, 2003: 294–311.

[9] YAN J, EL AHMAD A S. A low-cost attack on a Microsoft captcha[C]//Proceedings of the 15th ACM Conference on Computer and Communications Security. New York, USA, 2008: 543–554.
[10] GAO Haichang, WANG Xuqin, CAO Fang, et al. Robustness of text-based completely automated public Turing test to tell computers and humans apart[J]. IET information security, 2016, 10(1): 45–52.
[11] BURSZTEIN E, AIGRAIN J, MOSCICKI A, et al. The end is nigh: generic solving of text-based CAPTCHAs[C]// Proceedings of the 8th USENIX Conference on Offensive Technologies. Berkeley, USA, 2014.
[12] MOOSAVI-DEZFOOLI S M, FAWZI A, FROSSARD P. DeepFool: a simple and accurate method to fool deep neural networks[C]//Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, USA, 2016: 2574–2582.
[13] PAPERNOT N, MCDANIEL P, GOODFELLOW I, et al. Practical black-box attacks against machine learning[C]//Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. New York, USA, 2017: 506–519.
[14] XIAO Chaowei, ZHU Junyan, LI Bo, et al. Spatially transformed adversarial examples[EB/OL]. arXiv preprint arXiv: 1801.02612, 2018.
[15] CHELLAPILLA K, SIMARD P Y. Using machine learning to break visual human interaction proofs (HIPs)[C]//Proceedings of the 17th International Conference on Neural Information Processing Systems. Vancouver, Canada, 2004: 265–272.

本文引用格式：

马军, 王效武, 朱永川, 等. 基于对抗样本生成的验证码反爬虫机制研究 [J]. 应用科技, 2021, 48(6): 45–50.
MA Jun, WANG Xiaowu, ZHU Yongchuan, et al. Study on the verification code anti-crawler mechanism based on the generation of adversarial samples[J]. Applied science and technology, 2021, 48(6): 45–50.

(上接第 44 页)

[5] BENGIO Y. Deep learning of representations: looking forward[C]// Proceedings of the First international conference on Statistical Language and Speech Processing. Heidelberg: Springer Berlin Heidelberg, 2013: 1–37.
[6] REN Shaoqing, HE Kaiming, GIRSHICK R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[C]//Proceedings of Annual Conference on Neural Information Processing Systems. Montreal, Canada, 2015: 91–99.
[7] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection[C]// Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, USA, 2016: 779–788.
[8] LIU Wei, ANGUELOV D, ERHAN D, et al. SSD: single shot multibox detector[C]//Proceedings of the 14th European Conference on Computer Vision. Amsterdam, The Netherlands, 2016: 21–37.
[9] 韩家明, 杨忠, 陈聪, 等. 无人机视觉导航着陆标识检测与分割方法 [J]. 应用科技, 2020, 47(4): 1–7, 13.
[10] REDMON J, FARHADI A. YOLO9000: better, faster, stronger[C]//Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, USA, 2017: 6517–6525.
[11] REDMON J, FARHADI A. YOLOv3: an incremental improvementt[J/OL]. arXiv. org, (2018 –04 –08) [2010–11–23]. https://arxiv.org/abs/1804.02767.
[12] 王翠萍. Android Studio 应用开发实战详解 [M]. 北京: 人民邮电出版社, 2017: 114–179.
[13] 郑逸凡. Java 多线程机制及其在 socket 编程中的应用 [J]. 赤峰学院学报(自然科学版), 2018, 34(9): 62–63.
[14] 王朝硕, 李伟性, 郑武略, 等. 一种改进 SSD 的输电线路电力部件识别方法 [J]. 应用科技, 2020, 47(4): 75–81.
[15] 邢浩强, 杜志岐, 苏波. 基于改进 SSD 的行人检测方法 [J]. 计算机工程, 2018, 44(11): 228–233, 238.
[16] 柯飞雄. 基于 RK3399 多主机系统实现方法 [J]. 通讯世界, 2020, 27(4): 51–52.

本文引用格式：

樊雪倩, 陈春雨. 基于深度学习的智能广告牌的设计与实现 [J]. 应用科技, 2021, 48(6): 39–44, 50.
FAN Xueqian, CHEN Chunyu. Design and implementation of intelligent billboard based on deep learning[J]. Applied science and technology, 2021, 48(6): 39–44, 50.