## 1. 产品调研

Asirra captcha：broken using a classifier trained to recognize image textures



MintEye captcha：broken by a very simple attack based on Sobel operators that only required 23 lines of Python



## 2. 论文阅读与笔记

**《The End is Nigh:Generic Solving of Text-based CAPTCHAs》**

论文基本是宣告了文本验证码可以成为历史了，甚至说需要重新考虑reverse Turing test。比较重要的是论文在future of captchas中提出的可能设计方案，比如从当前计算机视觉还未解决的问题的角度考虑。

引用：Whether there exist situations where the human brain is definitively able to process information more efficiently than machines remains an open question.

今年计算机视觉领域的比赛ILSVRC今年都宣告结束，也说明在图像物体检测识别领域，计算机也不必人差了。下个还未被ai攻破，但已经被盯上的就是图像理解(从明年正式开始的WebVision比赛就是这一领域的)。

引用：Recently game-based captchas have been developed, however implementing this idea as proven to be difficult, as the game captcha schemes for the leading game captcha provider ”Are you a human” have been broken

给我的感觉是：设计困难，攻破简单。

引用：Human solving time is very predictable for random strings.

不禁想到看到的一个例子：一个验证码是解决复杂的高数问题，评论里有人戏称“看到这种验证码才发现原来自己不是人类”。对这类验证码来说，是不是能快速完成验证的才是机器？但这种方案也出现一个问题：如何衡量和预测人类完成验证的时间？如果依靠实验数据，必然导致设计验证码的人员的负担太大；而如果存在衡量的算法，攻击者又可以利用这些算法设计攻击方式。

引用：Clearly solving time is not a good reverse Turing test since it is easy for machines to fake, but other examples might exist, especially for captcha schemes that require a prolonged user engagement.

这里的有关想法记录在另一篇论文(神经科学与密码学交汇)的笔记里。

**《神经科学与密码学交汇:通过密码原语抵御软磨硬泡式攻击》**

这篇文章里提到的一个“内隐式学习”的概念，根据这个，可以训练用户在用户自己也不知道密码的情况下完成用户验证。虽然这个方案是基于密码，但是如果映射过来，把人类社会与单个人的交互映射到“用户训练”这一过程的话，验证码其实可以看做只允许人类通过的密码机制。我们需要做的就是验证对象是否是经过内隐式学习获得过某些知识的对象。其实我们设计的方案里，是有一点这个思想的：因为图片是随机截取的，所以我们其实并不能确定用户分组的依据，但是只要正确率到达标准了，就说明这个依据是确实存在并可以被大多数人所认可的。

**《Asirra: A CAPTCHA that Exploits Interest-Aligned Manual Image Categorization》**

**《Machine Learning Attacks Against the Asirra CAPTCHA》**

Asirra的提出以及被攻破过程，应该说，对如今的计算机视觉技术而言，这样的验证码已经完全无法用于区分人与机器了。

**《Video-Passwords: Advertising While Authenticating》**

提到了视频验证码可以利用的验证方式：Timeline，Click-Based，Tagged。根据这些考虑到我们的方案以随机截图为主，如果利用Timeline，可以选择让用户对乱序的图片排序；Click-Based多半还是需要对图片数据进行标记；Tagged需要考虑在我们方案里，截到的图片不一定会继承视频的Tag。

## 3. 设计报告

在前面的调研中，几乎所有验证码都有一个问题，其数据集严重依赖于人类的采集和手动打标签。为了解决这个问题，我们的设计中数据集中所有图片的来源都是从视频中截取。

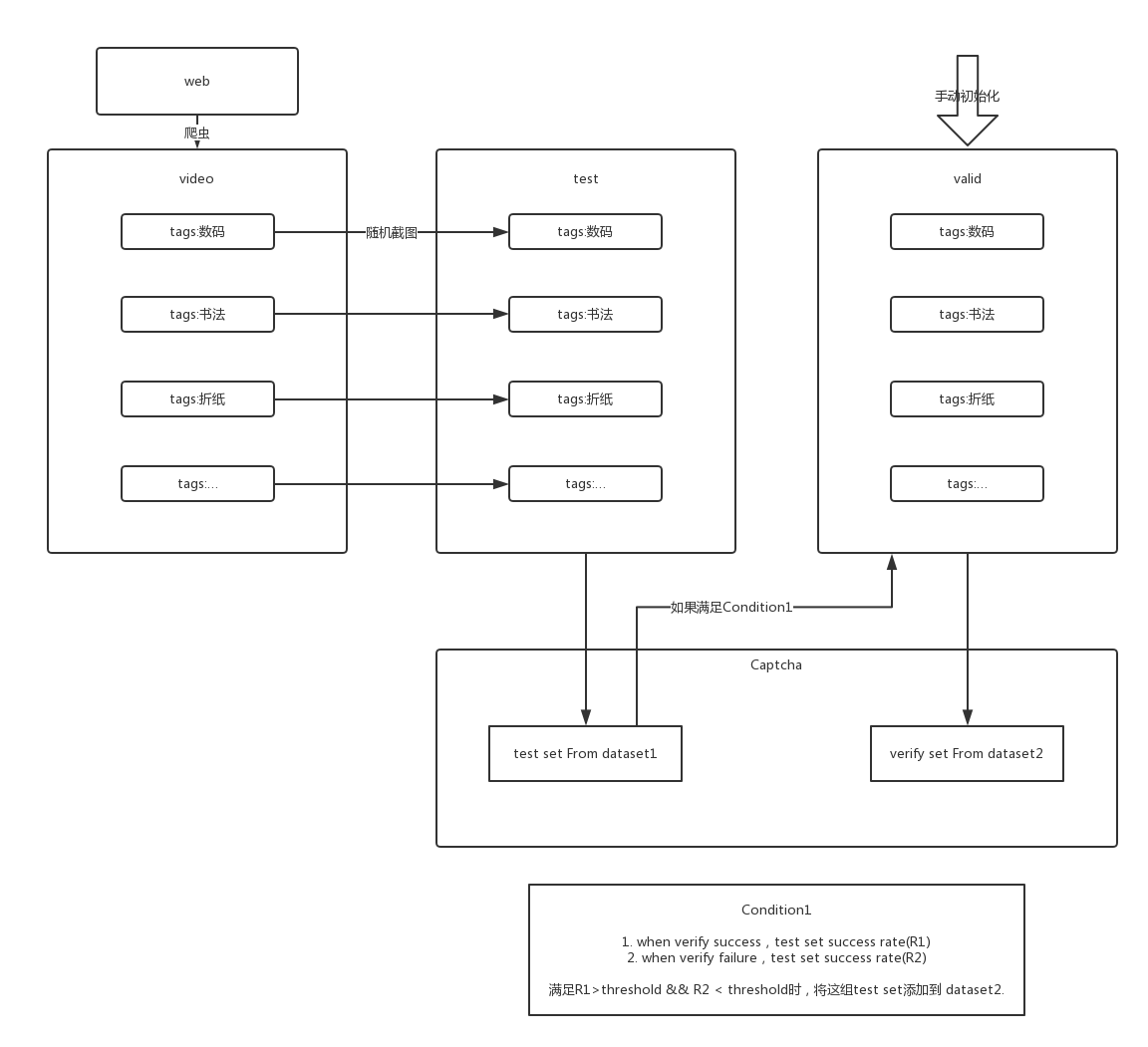
**数据集**

数据集中的图片分为两类，分别放入valid集和test集。关于两个数据集的分类依据及使用在后续会详细说明。

**从视频截图并根据tag分类**

* 视频源：各类视频网站。
* 截取方式：随机截取
* 验证方式一：将随机选择三个tag，然后每个tag内随机选择三张图片，打乱后让用户进行分组。此方案的验证时选择空间大小为1680。
* 验证方式二：随机选择三个tag，每个tag选择三个用来验证的图片和两个用来测试的图片，打乱后让用户分组。其中用来验证的九张分组正确就是验证通过了。此方案的验证时选择空间大小为15498。
* 是否把tag显示给用户：经过简单的实验(没有记录)，如果不显示tag，用户分组更可能依赖于图片之间的关系，将导致某些希望被筛选掉的图片难以被发现。因此后续实验都给用户显示分组tag。

**根据用户验证情况反馈丰富数据集**



该图是我们设计的一个总览，可以清晰的看出valid和test两个数据集的作用。简而言之，test集的数据直接来源时视频，valid集的数据直接来源是valid集。从视频中随机截取的图片只能进入test集。valid集是为了验证用户是否通过了验证，在用户通过验证的基础上，用户在test集上的数据会被收集和记录，最终在某些图片的数据达到我们的预期要求时，可以把这些图片添加到valid集丰富我们的数据集。

## 4. 系统设计

爬虫模块：

* 使用python编写，根据tag在视频网站上爬取查找视频
* 爬取结果按照时间排序，返回爬取的视频的链接
* 对链接使用脚本下载视频，主要依赖是一个名为you-get的开源视频下载工具

截图模块：

* 采用shell脚本编写
* 对每个视频截取五张图片

网页界面：

* 实现基本的用户验证过程
* 使用nodejs编写前后端



## 5. 实验报告

**实验参与者**

关于参与实验的成员，都是我们小组的成员，更广泛的说，参与实验者是学生，虽然未做过实验，但根据我们的分析，我们的实验数据很可能会比真正应用到实际中的用户验证通过率高。

不过我们的实验通过控制变量，保证了在不同测试间依然有可比性，因此该实验的数据仍具有参考价值。

针对我们的方案设计了三组实验。

实验一：

* 用户验证方式：随机对3×3的图片分组
* 测试次数：50次左右
* 数据集：100(test)

实验二：

* 用户验证方式：随机对3×5的图片分组，来自valid集的九张分组成功即可完成验证
* 测试次数：50次左右
* 数据集：100(test)+100(valid)

实验三：

* 用户验证方式：同二，不过valid集来自测试二中满足了条件(因为这部分数据集数量略显不足，有额外手段添加部分)的图片
* 测试次数：50次左右
* threshold：100%
* 数据集：100(test)+100(valid)
* Note：测试二与三的用户必须更换

**实验记录(Accept/Total/Rate)**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 验证 | Test集 | Valid集 |
| 实验一 | 32/45/0.71 | 370/405/0.91 | - |
| 实验二 | 13/40/0.32 | 512/600/0.86 | - |
| 实验三 | 42/50/0.82 | 255/300/0.85 | 425/450/0.94 |

其中验证是验证码的成功次数的记录，Test集和Valid集分别是对具体图片集的分组正确率的统计。

实验数据分析：

**实验一与二的对比**

先简单分析一下实验一，首先根据实验者反馈，有大量难以判断如何分类的图片，但是有时通过排除，是能正确完成分组的。因为每组三个是固定的，这样最后不确定如何分组的图片最后放，是有很大概率分对的。  
 对比实验二，虽然仍然是使用九张进行验证，但是加入了六张其实无实际作用的图片，每组的上限变为五个，导致的成功率下降极其明显。  
 对那些用户未能成功分组的图片进行分析，发现几乎都是即使告知其正确分组，也无法理解的那种。这种图片，由于是随机截图，所以其产生难以控制，但我们希望有机制能把它们剔除掉。

但一二的对比可以明显看到，在分类空间比较小的时候，排除法能使用户的分组成功率获得很大提升，这意味着一的方案不利于我们筛选不合格的图片。

**实验二与三的对比**

这两组的对比是为了测试我们方案的可行性。其中实验二中的总分组正确率为512/600=0.86。其中实验三的分析如下，对test集的分 类正确率255/300=0.85，对valid集的分类正确率425/450=0.94。  
 这些数据已经很能说明问题了，在valid集仅比test集正确率高0.1左右的情况下，验证成功率从32%左右提升到了82%左右。  
 这次实验中的threshold选择了100%，因为数据集的不足和测试人员太少导致限制太多，最合适的threshold值实际并没有测试出来，这个值应该从希望的valid集正确率考虑，从我们实验推断，valid集上的正确率会与你选择的threshold值密切相关并比threshold值稍微低一点。

**对机器的抵御能力**

因为很多限制，这方面的实验没有实现，我们对这部分只有一些理论分析。首先因为图片是随机截取，我们是难以确定用户真正的分组依据的，比如有的地方是根据偶然截取的字幕完成的分组，有的则是根据对截取物体的联想(警车->警匪，砚台->书法)。对机器而言，这些未知的挑战是远比简单的物体识别任务难度要大的。

另外当tag是物体时，当前的计算机视觉技术应该不比人类的识别能力差的，但tag是一些诸如"极限运动"的tag时，起码很难直接找到合适的模型对图像分类。但tag的选择也需要小心，某些tag截取的图片很可能导致正确率飞速下降。

## 6. 总结

最大成果就是证明了在配合我们方案的情况下，随机截图用来做验证码是有可能的。这样就解决了验证码数据集高度依赖人类的问题。但是仍然有很多限制，比如一个正常视频中截取的图片如果缩放的太小，很多信息就看不清了。以及目前方案里，用户的一次验证其实是有点长的。