2018.09.20 By ΕκάτηⅡ

虽然之前查了各种鬼畜资料变态算法，但是没有任何使用经验，根本没有找到靠谱的思路。今天我的一位硬核宅老师启发了我“似圆度”这个做法。或许我们还是应该独立思考，然后从想得到的做法入手。

1. 战略问题
   1. 我们要不要以神经网络为核心？

不！

神经网络确实是应对各种验证码的核武器，然而人力物力成本其实也很惊人。我们可能需要手动识别一万左右的验证码，然后用最精良的显卡、内存、CPU训练几天几夜。即使训练效果真的很好，神经网络在识别验证码的时候依然要求强大的算力。而且一旦ISML改进了验证码，我们又需要重新训练。

* 1. 我们应该用程序模拟鼠标键盘，还是用程序来操纵浏览器（例如Selenium），还是直接用程序对ISML网站发命令？

本来我当然是想直接对ISML网站发命令，但是这样需要读懂ISML加载过程中大量的js代码。一旦他们做少量修改，则我们又要从头来过。操纵浏览器本身是比较容易的，然而操纵的同时我们还需要伪装Canvas指纹。这又是一项大工程！模拟鼠标键盘的方法虽然很土，很不稳定，而且使你无法用电脑干其他活，但更容易修改适应各种环境，不易被察觉为机器，其代码也更容易被大量的珂学家读懂（emm如果你读不懂的话……那你更不会想去自己伪造Canvas指纹）。

* 1. 我们应该使用很复杂的算法吗？

尽量不要！

验证码错了可以再来。**我们要做的是让更多的珂学家能够理解算法，并参与改进**。而且我们最好能够在90秒内完成一张验证码的识别。

1. 战术想法
   1. 名词解释（并非严格的数学解释！只在这篇文章中成立）
      1. 连通域：**“能连成一块的”色块**。例如下图中有大量连成一起的扭曲青色连通域，以及个数不多但“占地面积”很大的白色网格状连通域。
      2. 单连通域：连成一块，而且中间没有洞的色块。比如字母C,G,N,X。
      3. 复连通域：连成一块，但中间有一个或多个洞的色块。比如字母A,O,R,B。



* + 1. 面积：类似于通常所谓的面积。不过这里的面积指**一个连通域包含多少像素点**。
    2. 似圆度：**一个连通域的面积，除以这个连通域最小的外接圆的面积**。求最小的外接圆可能会太消耗时间。视情况我们可能把“最小的外接圆的面积”改成“以连通域的质心（假设连通域上每个像素点具有相等质量）为圆心的最小外接圆面积”，或者“能包住连通域的最小的水平-竖直矩形的面积”（这个可以称为“似方度”）。
    3. 二值化：把图形变成黑白两色。比如我们要识别的验证码可以把“不够白色的”颜色全部变成黑色，便于后续处理。
    4. 变换到频域：图片是一行一行的不同颜色像素点构成的。我们知道时域上的函数可以经过傅里叶变换/余弦变换之类的东西变换到频域上去分析（假如你不知道，请百度一下傅里叶变换（不是傅里叶级数），实在不明白就算了）。现在假设有一幅验证码图片，我们当然可以求每一个像素到最左边一列像素的水平距离。现在我们把这个水平距离假想成“时间”，或者说“自变量”，把这个点的颜色（黑或白）假想成“函数值”。然后我们对每一行上的这个颜色（函数值）随水平距离（“时间”）变化的过程做傅里叶变换（实际上我可能会去做二维或逐行的离散余弦变换）。现在我们得到了这一行的“频率”信息！我目测认为有字母的行和没有字母的行大概会有明显不同的特征。这样可以把有字母的行分离出来。
    5. 梯度：图像颜色变化的剧烈程度。假设我们对一幅二值化的图像求梯度，我们得到另一幅分辨率完全相同的图像。如果原图像某个点周围8个点不都是同一种颜色，则在新生成的图像中这个点为黑色。否则，新图像中这个点为白色。
  1. 具体做法：
     1. 首先二值化。“不够白”的全部变成黑色
     2. 求所有连通域，并求每个连通域的面积和似圆度/似方度，以及如果它是复连通域，它的“洞”有多大。带字母的连通域，其连通域面积、“洞”面积、外接圆/矩形面积和似圆度/似方度应该会与干扰色块明显不同。
     3. 根据ii求得的结果，把所有“不像是带有字母”的连通域去掉（变成白色）
     4. （作为备用手段）假如我们采取以上三步后用肉眼发现还剩下一些恶心的色块，我们可以把图片每一行/部分行/部分二维区域变换到频域，通过频域特征去除色块。
     5. （作为备用手段）如果试验发现色块式的字母与干扰色块放在一起时，对于现成的图像识别引擎来说干扰太大，则我们对图像求梯度，再试一试。