在前一课时我们介绍了多种多样的验证码，有图形文字的、有模拟点选的、有拖动滑动的，但其实归根结底都需要人来对某种情形做一些判断，然后把结果返回并提交。如果此时提交的验证码结果是正确的，并且通过了一些验证码的检测，就能成功突破这个验证码了。

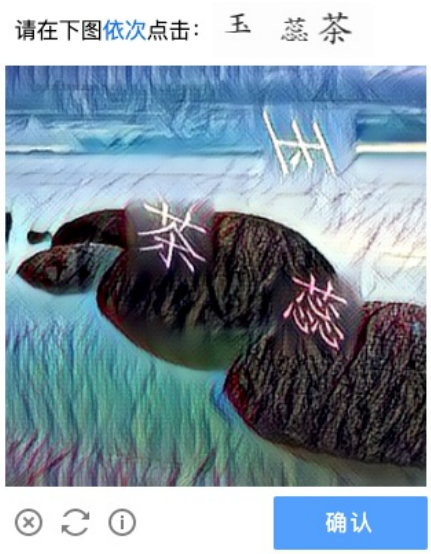
那么，既然验证码就是让人来识别的，那么机器怎么办呢？如果我们也不会什么算法，怎么去解这些验证码呢？此时如果有一个帮助我们来识别验证码的工具或平台就好了，让工具或平台把验证码识别的结果返回给我们，我们拿着结果提交，那不就好了吗？

有这种工具或平台吗？还真有专门的打码平台帮助我们来识别各种各样的验证码，平台内部对算法和人力做了集成，可以 7x24 小时来识别各种验证码，包括识别图形、坐标点、缺口等各种验证码，返回对应的结果或坐标，正好可以解决我们的问题。

本课时我们就来介绍利用打码平台来识别验证码的流程。

学习目标

本课时我们以一种点选验证码为例来讲解打码平台的使用方法，验证码的链接为：https://captcha3.scrape.cuiqingcai.com/，这个网站在每次登录的时候都会弹出一个验证码，其验证码效果图如下所示。



这个验证码上面显示了几个汉字，同时在图中也显示了几个汉字，我们需要按照顺序依次点击汉字在图中的位置，点击完成之后确认提交，即可完成验证。

这种验证码如果我们没有任何图像识别算法基础的话，是很难去识别的，所以这里我们可以借助打码平台来帮助我们识别汉字的位置。

准备工作

我们使用的 Python 库是 Selenium，使用的浏览器为 Chrome。

在本课时开始之前请确保已经正确安装好 Selenium 库、Chrome 浏览器，并配置好 ChromeDriver，相关流程可以参考 Selenium 那一课时的介绍。

另外本课时使用的打码平台是超级鹰，链接为：https://www.chaojiying.com/，在使用之前请你自己注册账号并获取一些题分供测试，另外还可以了解平台可识别的验证码的类别。

打码平台

打码平台能提供的服务种类一般都非常广泛，可识别的验证码类型也非常多，其中就包括点触验证码。

超级鹰平台同样支持简单的图形验证码识别。超级鹰平台提供了如下一些服务。

- 英文数字：提供最多 20 位英文数字的混合识别；
- 中文汉字：提供最多 7 个汉字的识别；
- 纯英文：提供最多 12 位的英文识别；
- 纯数字：提供最多 11 位的数字识别；
- 任意特殊字符：提供不定长汉字英文数字、拼音首字母、计算题、成语混合、集装箱号等字符的识别；
- 坐标选择识别：如复杂计算题、选择题四选一、问答题、点击相同的字、物品、动物等返回多个坐标的识别。

具体如有变动以官网为准：https://www.chaojiying.com/price.html。

这里需要处理的就是坐标多选识别的情况。我们先将验证码图片提交给平台，平台会返回识别结果在图片中的坐标位置，然后我们再解析坐标模拟点击。

下面我们就用程序来实现。

获取 API

在官方网站下载对应的 Python API，链接为：https://www.chaojiying.com/api-14.html。API 是 Python 2 版本的，是用 requests 库来实现的。我们可以简单更改几个地方，即可将其修改为 Python 3 版本。

修改之后的 API 如下所示：

```
import requests
from hashlib import md5
class Chaojiying(object):

    def __init__(self, username, password, soft_id):
        self.username = username
        self.password = md5(password.encode('utf-8')).hexdigest()
        self.soft_id = soft_id
        self.base_params = {
            'user': self.username,
            'pass2': self.password,
            'softid': self.soft_id,
        }
        self.headers = {
            'Connection': 'Keep-Alive',
            'User-Agent': 'Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)',
        }

    def post_pic(self, im, codetype):
        """
        im: 图片字节
        codetype: 题目类型 参考 http://www.chaojiying.com/price.html
        """
        params = {
            'codetype': codetype,
        }
        params.update(self.base_params)
        files = {'userfile': ('ccc.jpg', im)}
        r = requests.post('http://upload.chaojiying.net/Upload/Processing.php', data=params, files=files,
                          headers=self.headers)
        return r.json()

    def report_error(self, im_id):
        """
        im_id:报错题目的图片ID
        """
        params = {
            'id': im_id,
        }
        params.update(self.base_params)
        r = requests.post('http://upload.chaojiying.net/Upload/ReportError.php', data=params, headers=self.headers)
        return r.json()
```

这里定义了一个 Chaojiying 类，其构造函数接收三个参数，分别是超级鹰的用户名、密码以及软件 ID，保存以备使用。

最重要的一个方法叫作 post_pic，它需要传入图片对象和验证码类型的代号。该方法会将图片对象和相关信息发给超级鹰的后台进行识别，然后将识别成功的 JSON 返回。

另一个方法叫作 report_error，它是发生错误时的回调。如果验证码识别错误，调用此方法会返回相应的题分。

接下来，我们以 https://captcha3.scrape.cuiqingcai.com/ 为例来演示下识别的过程。

初始化

首先我们引入一些必要的包，然后初始化一些变量，如 WebDriver、Chaojiying 对象等，代码实现如下所示：

```
import time
from io import BytesIO
from PIL import Image
from selenium import webdriver
from selenium.webdriver import ActionChains
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from chaojiying import Chaojiying
USERNAME = 'admin'
PASSWORD = 'admin'
CHAOJIYING_USERNAME = ''
CHAOJIYING_PASSWORD = ''
CHAOJIYING_SOFT_ID = 893590
CHAOJIYING_KIND = 9102
if not CHAOJIYING_USERNAME or not CHAOJIYING_PASSWORD:
    print('请设置用户名和密码')
    exit(0)
class CrackCaptcha():
    def __init__(self):
        self.url = 'https://captcha3.scrape.cuiqingcai.com/'
        self.browser = webdriver.Chrome()
        self.wait = WebDriverWait(self.browser, 20)
        self.username = USERNAME
        self.password = PASSWORD
```

```
        self.chaojiying = Chaojiying(CHAOJIYING_USERNAME, CHAOJIYING_PASSWORD, CHAOJIYING_SOFT_ID)
```

这里的 USERNAME、PASSWORD 是示例网站的用户名和密码，都设置为 admin 即可。另外 CHAOJIYING_USERNAME、CHAOJIYING_PASSWORD 就是超级鹰打码平台的用户名和密码，可以自行设置成自己的。

另外这里定义了一个 CrackCaptcha 类，初始化了浏览器对象和打码平台的操作对象。

接下来我们用 Selenium 模拟呼出验证码开始验证就好啦。

获取验证码

接下来的步骤就是完善相关表单，模拟点击呼出验证码了，代码实现如下所示：

```
def open(self):
    """
    打开网页输入用户名密码
    :return: None
    """
    self.browser.get(self.url)
    # 填入用户名密码
    username = self.wait.until(EC.presence_of_element_located((By.CSS_SELECTOR, 'input[type="text"]')))
    password = self.wait.until(EC.presence_of_element_located((By.CSS_SELECTOR, 'input[type="password"]')))
    username.send_keys(self.username)
    password.send_keys(self.password)
def get_captcha_button(self):
    """
    获取初始验证按钮
    :return:
    """
    button = self.wait.until(EC.presence_of_element_located((By.CSS_SELECTOR, 'button[type="button"]')))
    return button
```

这里我们调用了 open 方法负责填写表单，get_captcha_button 方法获取验证码按钮，之后触发点击，这时候就可以看到页面已经把验证码呈现出来了。

有了验证码的图片，我们下一步要做的就是把验证码的具体内容获取下来，然后发送给打码平台识别。

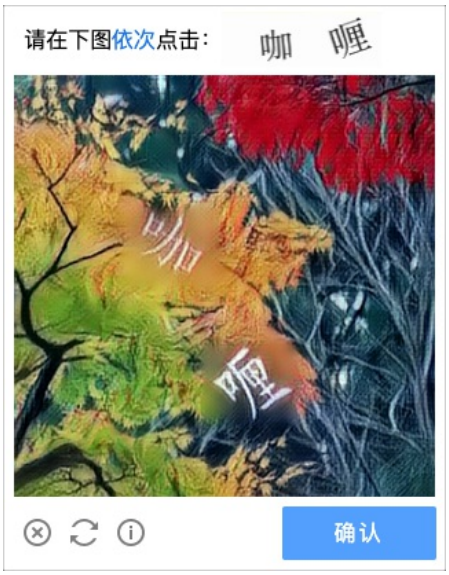
那怎么获取验证码的图片呢？我们可以先获取验证码图片的位置和大小，从网页截图里截取相应的验证码图片即可，代码实现如下所示：

```
def get_captcha_element(self):
    """
    获取验证图片对象
    :return: 图片对象
    """
    # 验证码图片加载出来
    self.wait.until(EC.presence_of_element_located((By.CSS_SELECTOR, 'img.geetest_item_img')))
    # 验证码完整节点
    element = self.wait.until(EC.presence_of_element_located((By.CLASS_NAME, 'geetest_panel_box')))
    print('成功获取验证码节点')
    return element
def get_captcha_position(self):
    """
    获取验证码位置
    :return: 验证码位置元组
    """
    element = self.get_captcha_element()
    time.sleep(2)
    location = element.location
    size = element.size
    top, bottom, left, right = location['y'], location['y'] + size['height'], location['x'], location['x'] + size[
        'width']
    return (top, bottom, left, right)
def get_screenshot(self):
    """
    获取网页截图
    :return: 截图对象
    """
    screenshot = self.browser.get_screenshot_as_png()
    screenshot = Image.open(BytesIO(screenshot))
    screenshot.save('screenshot.png')
    return screenshot
def get_captcha_image(self, name='captcha.png'):
    """
    获取验证码图片
    :return: 图片对象
    """
    top, bottom, left, right = self.get_captcha_position()
    print('验证码位置', top, bottom, left, right)
    screenshot = self.get_screenshot()
    captcha = screenshot.crop((left, top, right, bottom))
    captcha.save(name)
    return captcha
```

这里 get_captcha_image 方法即为从网页截图中截取对应的验证码图片，其中验证码图片的相对位置坐标由 get_captcha_position 方法返回得到。所以就是利用了先截图再裁切的方法获取了验证码。

注意：如果你的屏幕是高清屏如 Mac 的 Retina 屏幕的话，可能需要适当调整下屏幕分辨率或者对获取到的验证码位置做一些倍数偏移计算。

最后我们得到的验证码是 Image 对象，其结果样例如图所示。



识别验证码

现在我们有了验证码图了，下一步就是把图发送给打码平台了。

我们调用 Chaojiying 对象的 post_pic 方法，即可把图片发送给超级鹰后台，这里发送的图像是字节流格式，代码实现如下所示：

```
image = self.get_touclick_image()
bytes_array = BytesIO()
image.save(bytes_array, format='PNG')
# 识别验证码
result = self.chaojiying.post_pic(bytes_array.getvalue(), CHAOJIYING_KIND)
print(result)
```

运行之后，result 变量就是超级鹰后台的识别结果。可能运行需要等待几秒，它会返回一个 JSON 格式的字符串。

如果识别成功，典型的返回结果如下所示：

```
{'err_no': 0, 'err_str': 'OK', 'pic_id': '6002001380949200001', 'pic_str': '132,127|56,77', 'md5': '1f8e1d4bef8b11484cb1f1f34299865b'}
```

其中，pic_str 就是识别的文字的坐标，是以字符串形式返回的，每个坐标都以 | 分隔。接下来我们只需要将其解析，然后模拟点击，代码实现如下所示：

```
def get_points(self, captcha_result):
    """
    解析识别结果
    :param captcha_result: 识别结果
    :return: 转化后的结果
    """
    groups = captcha_result.get('pic_str').split('|')
    locations = [[int(number) for number in group.split(',')] for group in groups]
    return locations
def touch_click_words(self, locations):
    """
    点击验证图片
    :param locations: 点击位置
    :return: None
    """
    for location in locations:
        ActionChains(self.browser).move_to_element_with_offset(self.get_captcha_element(), location[0], location[1]).click().perform()
        time.sleep(1)
```

这里用 get_points 方法将识别结果变成列表的形式。touch_click_words 方法则通过调用 move_to_element_with_offset 方法依次传入解析后的坐标，点击即可。

这样我们就模拟完成坐标的点选了，运行效果如下所示。

最后再模拟点击提交验证的按钮，等待验证通过就会自动登录啦，后续实现在此不再赘述。

如何判断登录是否成功呢？同样可以使用 Selenium 的判定条件，比如判断页面里面出现了某个文字就代表登录成功了，代码如下：

```
# 判定是否成功
success = self.wait.until(EC.text_to_be_present_in_element((By.TAG_NAME, 'h2'), '登录成功'))
```

比如这里我们判定了点击确认按钮，页面会不会跳转到提示成功的页面，成功的页面包含一个 h2 节点，包含“登录成功”四个字，就代表登录成功啦。

这样我们就借助在线验证码平台完成了点触验证码的识别。此方法是一种通用方法，我们也可以用此方法来识别图文、数字、算术等各种各样的验证码。

结语

本课时我们通过在线打码平台辅助完成了验证码的识别。这种识别方法非常强大，几乎任意的验证码都可以识别。如果遇到难题，借助打码平台无疑是一个极佳的选择。