# 目标

随着数字图像使用的增加，制造数字图像伪造的手段和动机也在增加。因此，对能够检测图像改变和伪造图像的数字图像取证技术有很大的需求，实现对图片是否被PS修改

# 实现方法

主要分两级鉴别：

## 方法一：

通过检查图片文件中的metadata（元数据）获取图片被修改的签名信息，如被Photoshop修改的话会有相关的签名字样，但是这个不能完全杜绝通过其他途径修改图片信息。

代码：

链接：https://www.cnblogs.com/fanghao/p/7629207.html

#!/usr/bin/python

import sys

print open(sys.argv[1]).read().lower().find("photo")!=-1

./isPS.py test.png #在运行之前要给isPS.py添加执行权限chmod u+x isPS.py

f=1img = PIL.Image.open(imgName)

info = 1img.\_getexif()

if info:

for (tag, value) in info.items():

if "Software" == TAGS.get(tag, tag):

lab['text'] = lab['text'] + "非拍照图"

lab['text'] = lab['text'] + "找到修改软件的签名 : " + value

f=0

if f:

lab['text'] = lab['text'] + "没找到修改软件签名"

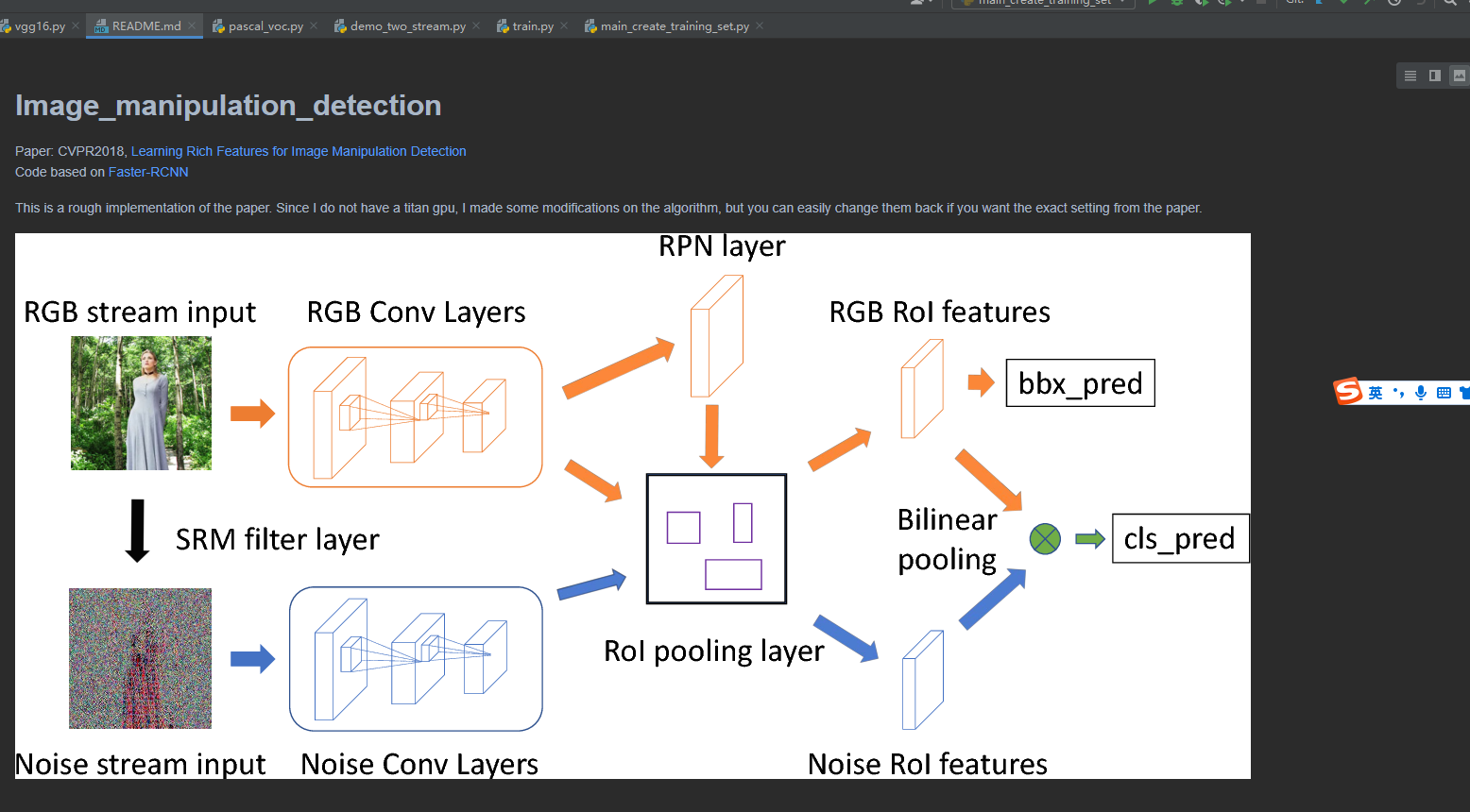
lab['text'] = lab['text'] + "看似是真实图片"

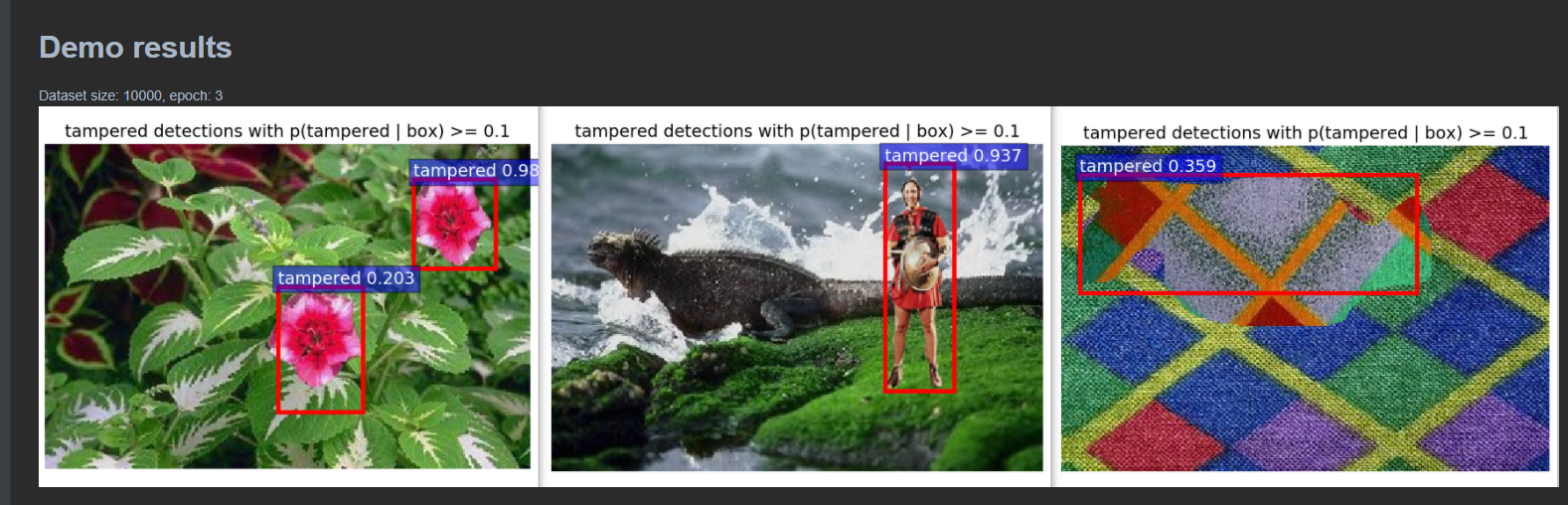
同样可以使用web端上传图片实现对元数据的检查,分析工具地址：<http://exifdata.com/>。

## 方法二：

通过内外网查阅论文，找到《Learning Rich Features for Image Manipulation Detection》论文及其代码，由于算法和环境配置复杂、显卡要求高，暂时搁置。

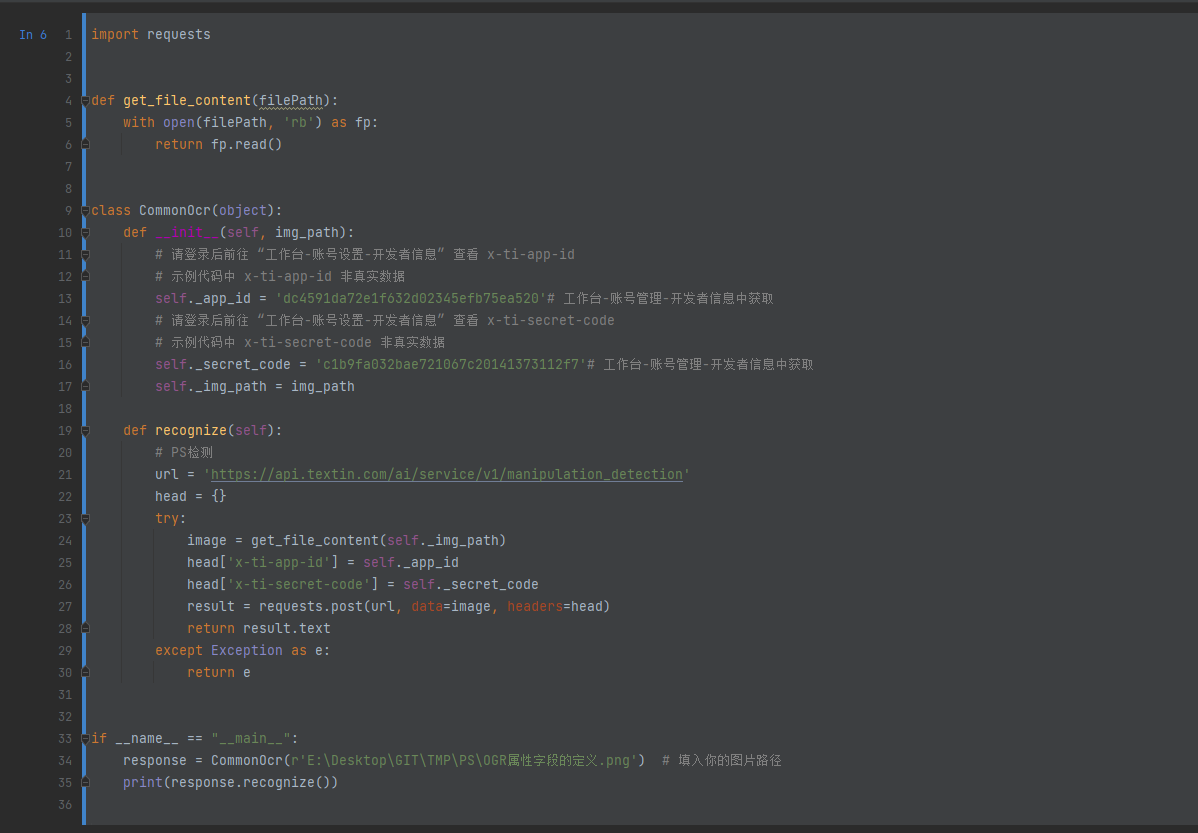
论文地址：https://openaccess.thecvf.com/content\_cvpr\_2018/html/Zhou\_Learning\_Rich\_Features\_CVPR\_2018\_paper.html

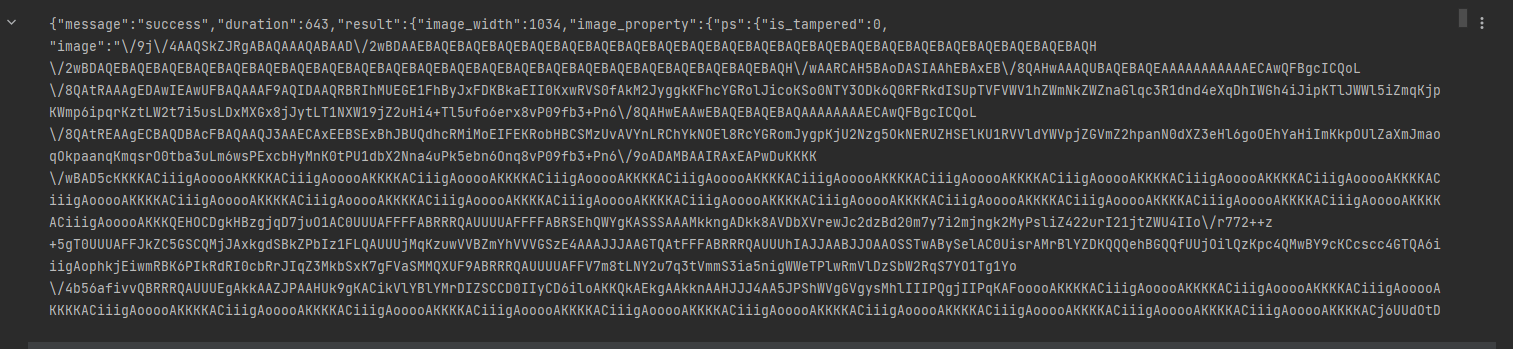




## 方法三：

通过第三方接口https://api.textin.com/ai/service/v1/manipulation\_detection，相对实现简单算法成熟。上传图片的检测。





代码：

import requests  
def get\_file\_content(filePath):  
 with open(filePath, 'rb') as fp:  
 return fp.read()  
class CommonOcr(object):  
 def \_\_init\_\_(self, img\_path):  
 # 请登录后前往 “工作台-账号设置-开发者信息” 查看 x-ti-app-id  
 # 示例代码中 x-ti-app-id 非真实数据  
 self.\_app\_id = 'dc4591da72e1f632d02345efb75ea520'# 工作台-账号管理-开发者信息中获取  
 # 请登录后前往 “工作台-账号设置-开发者信息” 查看 x-ti-secret-code  
 # 示例代码中 x-ti-secret-code 非真实数据  
 self.\_secret\_code = 'c1b9fa032bae721067c20141373112f7'# 工作台-账号管理-开发者信息中获取  
 self.\_img\_path = img\_path  
 def recognize(self):  
 # PS检测  
 url = 'https://api.textin.com/ai/service/v1/manipulation\_detection'  
 head = {}  
 try:  
 image = get\_file\_content(self.\_img\_path)  
 head['x-ti-app-id'] = self.\_app\_id  
 head['x-ti-secret-code'] = self.\_secret\_code  
 result = requests.post(url, data=image, headers=head)  
 return result.text  
 except Exception as e:  
 return e  
if \_\_name\_\_ == "\_\_main\_\_":  
 response = CommonOcr(r'E:\Desktop\GIT\TMP\PS\OGR属性字段的定义.png') # 填入你的图片路径  
 print(response.recognize())

## 方法四：Forensically

文章链接：https://www.zhihu.com/question/377619210

工具链接<https://29a.ch/photo-forensics/#clone-detection>

**Forensically** 是一款基于 Web 的免费图像分析工具，可用于检测克隆、错误级别分析、图像元数据、噪声分析、水平扫描等。错误级别分析或 ELA 用于识别 jpg 图像中不同级别的压缩伪影。

jpg 图像保存得越多，压缩得越多，因此在对其进行分析时，图像的编辑部分变得不那么均匀，因此更加明显。许多人通常认为错误级别分析是主观的，但它可以像放大镜一样帮助您识别图像中被隐藏的细节，帮助揭示真相。

* **克隆检测：**克隆检测器突出显示图像中的复制区域。这些可以很好地指示图片已被操纵。
* **错误级别分析：**此工具将原始图像与重新压缩的版本进行比较。这可以使操纵区域以各种方式脱颖而出。例如，它们可以比未经处理的类似区域更暗或更亮。
* **噪声分析：**这个工具基本上是一种反向去噪算法。它不是去除噪声，而是去除图像的其余部分。它使用超级简单的可分离中值滤波器来隔离噪声。它可用于识别对图像的操作，如喷枪、变形、扭曲和透视校正克隆。它在高质量图像上效果最好。
* **PCA：**此工具对图像执行主成分分析。这提供了一个不同的角度来查看图像数据，这使得发现某些操作和细节更容易。这个工具目前是[单线程](https://www.zhihu.com/search?q=%E5%8D%95%E7%BA%BF%E7%A8%8B&search_source=Entity&hybrid_search_source=Entity&hybrid_search_extra=%7B%22sourceType%22%3A%22answer%22%2C%22sourceId%22%3A2272197707%7D)的，在大图像上运行时速度很慢。

## 方法五：

Adobe 公司的AI鉴图技术主要用了两种鉴定方法。第一种是RGB 识别技术。任何一个电子屏幕上看起来五彩缤纷的图片，其实每个像素点都是由红蓝绿三种颜色叠加形成的。

红绿蓝，是色光的三原色。但是，并非每个格子都储存着红黄蓝三种信号，大多数时候需要通过格子和格子之间的算法来控制颜色，所以每个格子之间其实有一定的数字关系。如果某一篇区域的图像跟周围没有数字关系，或者发生异常，就可以判定图片存在PS。简而言之，根据RGB颜色值的变化关系，AI系统就能迅速来检测出篡改。第二种方法是噪点。我们使用数码相机拍照时，照片上一定会带有很多密密麻麻的小颗粒，夜间拍摄时尤其明显。这些噪点其实也存在一些特定的分布规律，如果图片中的某个部分是复制进来的，噪点的分布以及边缘就会异常，这种异常用肉眼很难识别，用机器却可以轻易找出来。RGB检测和噪点检测两种方法叠加。

# 总结

简单的元数据检测是否被PS数据处理不能确保该图片元数据被修改，因此从算法层面考虑，通过深度算法对图片进行检测，方法二算法配置复杂暂时搁置，方法三通过调用第三方接口，上传待检测图片对图片进行检测，返回Json文件包含图片宽高和是否被修改等信息。