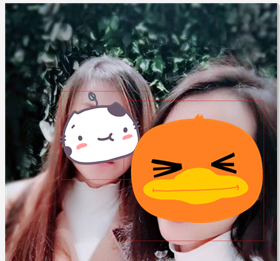
蝈蝈写了两期的爬虫了，说实话蝈蝈自己都厌倦了，不过爬虫我只用了皮毛而已，还有更高性能的包和框架在前面等着。不过让我先喘口气，今天我们带来的是人脸识别。

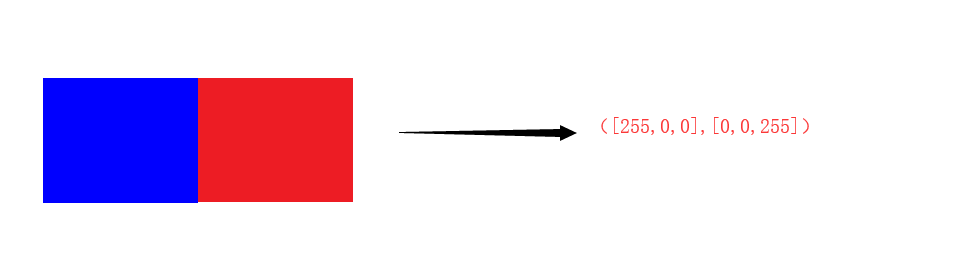
首先要讲解原理，这是我们部门两个最可爱的妹子，经她们同意我就用这张照片来讲解（个人隐私问题已打码处理）。

图像学是一个很复杂的学科，真的研究起来，没有极为强大的知识是不行的，蝈蝈只是个码农，并不会那些东西，好在python为我们封装了一个包python-opencv，只要几行代码就能完成人脸识别了。

OpenCV是一个基于BSD许可（开源）发行的跨平台计算机视觉库，可以运行在Linux、Windows、Android和Mac OS操作系统上。它轻量级而且高效——由一系列 C 函数和少量 C++ 类[构成](https://baike.baidu.com/item/%E6%9E%84%E6%88%90/103686" \t "https://baike.baidu.com/item/opencv/_blank)，同时提供了Python、Ruby、MATLAB等语言的接口，实现了[图像处理](https://baike.baidu.com/item/%E5%9B%BE%E5%83%8F%E5%A4%84%E7%90%86" \t "https://baike.baidu.com/item/opencv/_blank)和计算机视觉方面的很多通用算法。（摘自百度百科）



右边这张图的红框，是根据网上教程写的，opencv自动人脸识别，并作标记机，我们就不画长方形了，我们可以改成把帽子粘帖在他们头上，就像图中遮挡物一样。

Python—opencv的原理就是把每个像素点用一个BGR的列表所表示比如下图。人脸识别的原理应该就是计算机寻找颜色浅黄，白，黑，褐色的一块近似矩形的区域。

一.准备工作 语言：Python3

工具：Pycharm，Windows画图/Photoshop

技术:Opencv2/Opencv3,numpy

1. 流程概述 贴图前期处理

人脸识别

图片缩放

调试戴帽位置

出图

1. 详细实现

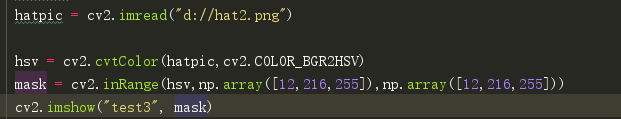
Opencv安装pip install python-opencv

或许会碰到无法编译的问题，可以直接去官方下载包，然后复制cv2.pyd到python site-package目录里就OK了

1.贴图处理

这是需要带的帽子，由于接触opencv时间较短，alpha通道啊，掩膜的位与运算，理解并不透彻，所以只能用最简单的色差抠图。

这是处理以后的帽子，可以看出每一块颜色都不同



这便是贴图处理的代码，读取我们已经颜色处理好的图片

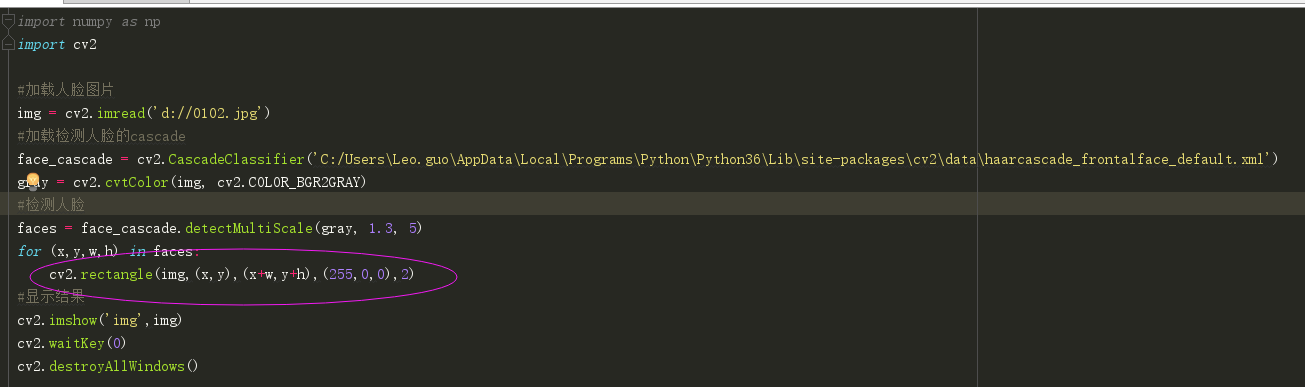
然后使用hsv变换让颜色更鲜明，橙色的背景变成了金色

第三行是从图片中获取不含金色部分的掩膜

第四行显示图片，最后效果如下

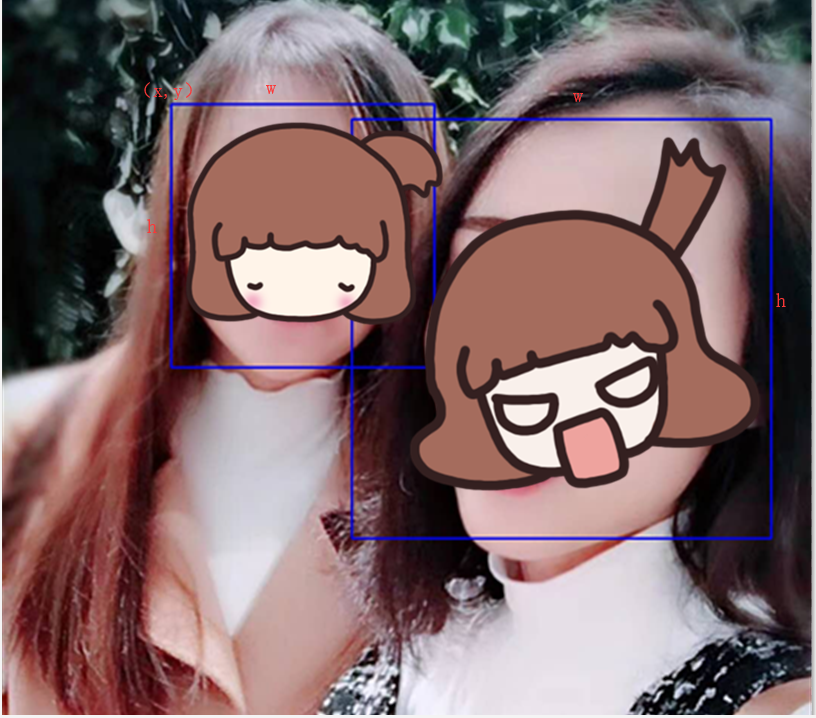


2.人脸识别



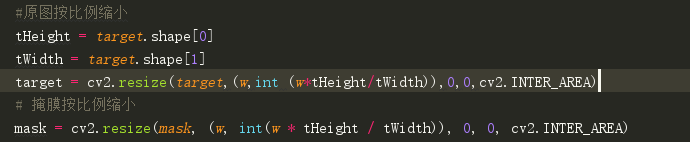
以上便是人脸识别并画框框的代码，我们需要做的是更换打圈的代码，就是画框部分

代码的含义是读取背景图片，然后载入官方人脸识别训练文件，把图片变成灰度显示加快速度和精确性，然后返回人脸的范围（x,y是起始坐标，w,h是该范围矩形的宽度和高度）



1. 图片缩放

背景图片是各种各种的，所以人脸有大有小，但是我们的贴图却是一张我们需要调整贴图大小适应脸形

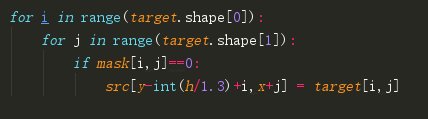


先获取原图片大小，然后指定要缩小的宽度w，然后按比例计算出高度，具体谈实现就是这么简单。我这里帽子是永远和脸部一样宽，大家可以修改w参数，让帽子更小一点。

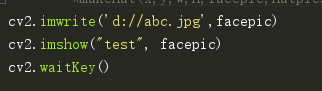
同时别忘记我们的掩膜（那张黑白两色的图片）也同样缩小

1. 调试戴帽子的位置

这一步是整个项目最最核心的位置，也是最可优化的部分，代码如下



按长宽遍历缩放以后的贴图，如果发现掩膜同一部位是黑色的，那就把贴图该部位的像素替换到人脸部位上方某个位置（可以调整）



最后实现效果就如最开始图片所示，最后蝈蝈提醒大家，千万别把贴图主色调设定成绿色的，小心被打啊。