

用Python就可以给你的头像戴上圣诞帽，别@微信团队了！

2017-12-25 冰不语 AI科技大本营



本文授权转自微信公众号CVPy（ID：x-cvpy）

作者 | 冰不语

引言

随着圣诞的到来，大家纷纷@官方微信给自己的头像加上一顶圣诞帽。当然这种事情用很多P图软件都可以做到。但是作为一个学习图像处理的技术人，还是觉得我们有必要写一个程序来做这件事情。而且这完全可以作为一个练手的小项目，工作量不大，而且很有意思。

用到的工具

- OpenCV（毕竟我们主要的内容就是OpenCV...）
- dlib（前一篇文章刚说过，dlib的人脸检测比OpenCV更好用，而且dlib有OpenCV没有的关键点检测。）

用到的语言为Python。但是完全可以改成C++版本，时间有限，就不写了。有兴趣的小伙伴可以拿来练手。

流程

一、素材准备

首先我们需要准备一个圣诞帽的素材，格式最好为PNG，因为PNG的话我们可以直接用Alpha通道作为掩膜使用。我们用到的圣诞帽如下图：



我们通过通道分离可以得到圣诞帽图像的alpha通道。代码如下：

```
r,g,b,a = cv2.split(hat_img)
rgb_hat = cv2.merge((r,g,b))

cv2.imwrite("hat_alpha.jpg",a)
```

为了能够与rgb通道的头像图片进行运算，我们把rgb三通道合成一张rgb的彩色帽子图。Alpha通道的图像如下图所示。



二、人脸检测与人脸关键点检测

我们用下面这张图作为我们的测试图片。



下面我们用dlib的正脸检测器进行人脸检测，用dlib提供的模型提取人脸的五个关键点。代码如下：

```
# dlib人脸关键点检测器
predictor_path = "shape_predictor_5_face_landmarks.dat"
predictor = dlib.shape_predictor(predictor_path)

# dlib正脸检测器
```

```
detector = dlib.get_frontal_face_detector()

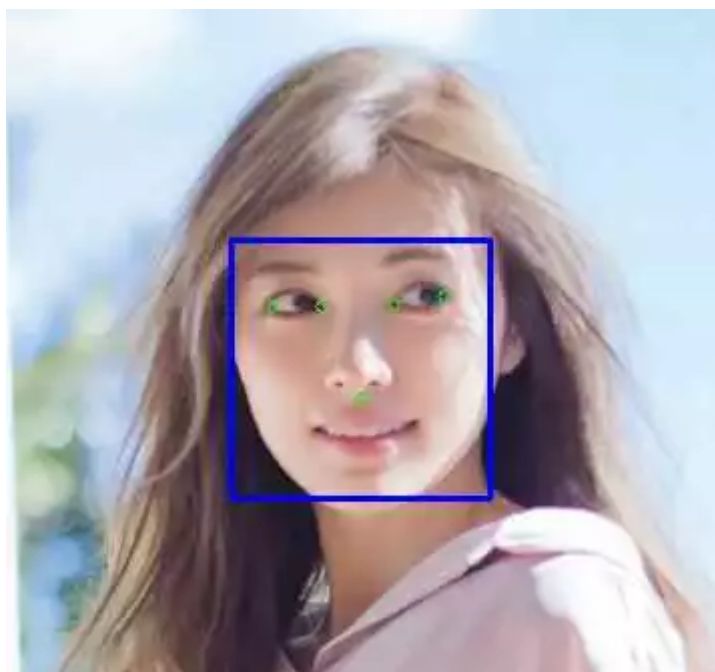
# 正脸检测
dets = detector(img, 1)

# 如果检测到人脸
if len(dets)>0:
    for d in dets:
        x,y,w,h = d.left(),d.top(), d.right()-d.left(), d.bottom()-d.top()
        # x,y,w,h = faceRect
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2,8,0)

        # 关键点检测，5个关键点
        shape = predictor(img, d)
        for point in shape.parts():
            cv2.circle(img,(point.x,point.y),3,color=(0,255,0))

        cv2.imshow("image",img)
        cv2.waitKey()
```

这部分效果如下图：



三、调整帽子大小

我们选取两个眼角的点，求中心作为放置帽子的x方向的参考坐标，y方向的坐标用人脸框上线的y坐标表示。然后我们根据人脸检测得到的人脸的大小调整帽子的大小，使得帽子大小合适。

```
# 选取左右眼眼角的点
point1 = shape.part(0)
point2 = shape.part(2)

# 求两点中心
eyes_center = ((point1.x+point2.x)//2,(point1.y+point2.y)//2)

# cv2.circle(img,eyes_center,3,color=(0,255,0))
# cv2.imshow("image",img)
# cv2.waitKey()

# 根据人脸大小调整帽子大小
factor = 1.5
resized_hat_h = int(round(rgb_hat.shape[0]*w/rgb_hat.shape[1]*factor))
resized_hat_w = int(round(rgb_hat.shape[1]*w/rgb_hat.shape[1]*factor))

if resized_hat_h > y:
    resized_hat_h = y-1

# 根据人脸大小调整帽子大小
resized_hat = cv2.resize(rgb_hat,(resized_hat_w,resized_hat_h))
```

四、提取帽子和需要添加帽子的区域

按照之前所述，去Alpha通道作为mask。并求反。这两个mask一个用于把帽子图中的帽子区域取出来，一个用于把人物图中需要填帽子的区域空出来。后面你将会看到。

```
# 用alpha通道作为mask
mask = cv2.resize(a,(resized_hat_w,resized_hat_h))
mask_inv = cv2.bitwise_not(mask)
```

从原图中取出需要添加帽子的区域，这里我们用的是位运算操作。

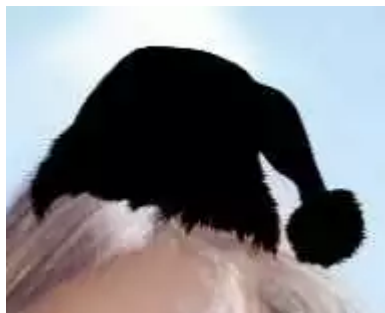
```
# 帽子相对与人脸框上线的偏移量
dh = 0
dw = 0
# 原图ROI
# bg_roi = img[y+dh-resized_hat_h:y+dh, x+dw:x+dw+resized_hat_w]
```

```
bg_roi = img[y+dh-resized_hat_h:y+dh,(eyes_center[0]-resized_hat_w//3):
(eyes_center[0]+resized_hat_w//3*2)]

# 原图ROI中提取放帽子的区域
bg_roi = bg_roi.astype(float)
mask_inv = cv2.merge((mask_inv,mask_inv,mask_inv))
alpha = mask_inv.astype(float)/255

# 相乘之前保证两者大小一致（可能会由于四舍五入原因不一致）
alpha = cv2.resize(alpha,(bg_roi.shape[1],bg_roi.shape[0]))
# print("alpha size: ",alpha.shape)
# print("bg_roi size: ",bg_roi.shape)
bg = cv2.multiply(alpha, bg_roi)
bg = bg.astype('uint8')
```

这是的背景区域（bg）如下图所示。可以看到，刚好是需要填充帽子的区域缺失了。



然后我们提取帽子区域。

```
# 提取帽子区域
hat = cv2.bitwise_and(resized_hat,resized_hat,mask = mask)
```

提取得到的帽子区域如下图。帽子区域正好与上一个背景区域互补。



五、添加圣诞帽

最后我们把两个区域相加。再放回到原图中去，就可以得到我们想要的圣诞帽图了。这里需要注意的就是，相加之前resize一下保证两者大小一致，因为可能会由于四舍五入原因不一致。

```
# 相加之前保证两者大小一致（可能会由于四舍五入原因不一致）
hat = cv2.resize(hat,(bg_roi.shape[1],bg_roi.shape[0]))
# 两个ROI区域相加
add_hat = cv2.add(bg,hat)
# cv2.imshow("add_hat",add_hat)

# 把添加好帽子的区域放回原图
img[y+dh-resized_hat_h:y+dh,(eyes_center[0]-resized_hat_w//3):
(eyes_center[0]+resized_hat_w//3*2)] = add_hat
```

最后我们得到的效果图如下所示。



源码地址：<https://github.com/LiuXiaolong19920720/Add-Christmas-Hat>，或者点击“阅读原文”查看源码

不用数学也能讲清贝叶斯理论的马尔可夫链蒙特卡洛方法？这篇文章做到了
盘点深度学习一年来在文本、语音和视觉等方向的进展，看强化学习如何无往而不利
论文解析 | Google如何用CNN检查乳腺癌？

这三个普通程序员，几个月就成功转型AI，他们的经验是...

干货 | AI 工程师必读，从实践的角度解析一名合格的AI工程师是怎样炼成的

AI校招程序员最高薪酬曝光！腾讯80万年薪领跑，还送北京户口

算法还是算力？周志华微博引爆深度学习的“鸡生蛋，蛋生鸡”问题

详解 | 如何用Python实现机器学习算法

经验 | 如何高效学Python？

如何成为一名全栈语音识别工程师？

Twitter大牛写给你的机器学习进阶手册



内容转载自公众号



[了解更多 >](#)

[阅读原文](#)