

使用opencv绘制素描图

一、应用场景

我们经常在网页上看到一些有趣的素描图，其实无需通过手工素描，利用Python也可以绘制出一张好看的素描图。

二、项目代码与结果展示

2.1 项目代码

```
1 import cv2
2
3 def dodging(image, mask):  # 亮化操作
4     return cv2.divide(image, 255 - mask, scale=256)
5
6
7 def control_to_sketch(image, to_image):
8     img_rgb = cv2.imread(image)
9     img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_BGR2GRAY)
10
11     img_gray_inv = 255 - img_gray
12     img_blur = cv2.GaussianBlur(img_gray_inv, ksize=(21, 21),
13                                 sigmaX=0, sigmaY=0)
14     img_blend = dodging(img_gray, img_blur)
15     cv2.imshow("sketch", img_blend)
16     cv2.waitKey(100)
17     cv2.destroyAllWindows()
18     cv2.imwrite(to_image, img_blend)
19
20
21 image = '小猫咪.jpg'  # 此处需要传递一个图片的路径
22 to_image = '小猫咪1.jpg'  # 此处传递生成图片的路径
23 control_to_sketch(image, to_image)
```

2.2 结果展示

原始图片（本照片来源于网络）



生成图片





三、项目操作及解析

3.1 安装opencv库

本项目我们需要安装opencv库，在终端输入`pip install opencv-python`（Mac用户输入`pip3 install opencv-python`）即可成功安装。顺便一说，opencv库是目前最流行的开源计算机视觉库，其封装了许多与计算机视觉相关的算法。

3.2 过程讲解

本项目需要使用众多的计算机视觉算法，如果想要彻底理解其原理，需要具备基础的微积分、线性代数以及计算机视觉的知识，因此本项目只讲解大概过程。不过不用担心，你只需要修改代码最后三行的原始图片路径和生成图片路径即可使用此程序。

步骤一：打开图片并将其转化为灰度图片

```
1 img_rgb = cv2.imread(image)
2 img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_BGR2GRAY)
```

步骤二：将灰度图反色

灰度图片的灰度值界限为0到255，将灰度图反色其实就是用255减去每个像素点的灰度值。换句话说就是让图片里黑的部分变白，白的部分变黑。

```
1 img_gray_inv = 255 - img_gray
```

步骤三：对反色图高斯模糊

高斯模糊在数学上其实是一种基于高斯核的卷积操作，在计算机图形学中高斯模糊可以减少图像噪声以及降低细节层次。在opencv库中提供了这个操作，使用cv2.GaussianBlur()函数即可，其中参数ksize为高斯核，参数sigmaX和sigmaY表示在笛卡尔坐标系下高斯核在两个方向的标准差（此内容了解即可）。

```
1 img_blur = cv2.GaussianBlur(img_gray_inv, ksize=(21, 21), sigmaX=0, sigmaY=0)
```

步骤四：亮化

使用过Ps的小伙伴可能会知道亮化与暗化两种技术，它们可以使得图片中的某一块区域变亮或者变暗。本项目仅需使用亮化技术，我们可以写一个函数来实现图片的亮化。

```
1 def dodging(image, mask): # 亮化操作
2     return cv2.divide(image, 255 - mask, scale=256)
```

注意：cv2.divide()是一种图片的融合技术，相对应还有cv2.add()、cv2.subtract()以及cv2.multiply() 三种融合方法，感兴趣可以做进一步了解。