

人脸照片的美化

1. 图像分析

待处理的图片为一个脸上有大量斑点的女性，同时还存在肤色偏暗的问题。对于斑点的去除，可以采用双边滤波的方法，因为双边滤波可以在保持图像边缘的情况下对图像的内容进行平滑处理，适合进行磨皮的场景。但是过度磨皮会丢失大量细节，使得任务皮肤看起来不真实，所以在双边滤波后还要叠加原图像以保持原图的纹理细节。对于肤色提亮则可以通过调节 Gamma 曲线来达到效果，将 Gamma 曲线的 Gamma 值调到小于 1 的值，就可以提高亮度的对比度，从而提亮肤色。

2. 实验过程

主要使用 python 以及 opencv 进行图像的处理，代码如下所示：

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

def adjust_gamma(image, gamma=1.0):
    invGamma = 1.0/gamma
    table = []
    for i in range(256):
        table.append(((i / 255.0) ** invGamma) * 255)
    table = np.array(table).astype("uint8")
    return cv2.LUT(image, table)

img = cv2.imread('beauty.jpg')
b, g, r = cv2.split(img)
img = cv2.merge([r, g, b])
g2 = cv2.bilateralFilter(g, 5, 100, 100)
b2 = cv2.bilateralFilter(b, 5, 100, 100)
filtered2 = cv2.merge([r, g2, b2])

filtered = cv2.bilateralFilter(img, 10, 100, 75)
filtered = 0.95*filtered + 0.05*filtered2
filtered = filtered.astype(np.uint8)
filtered = adjust_gamma(filtered, 1.4)

plt.subplot(1,2,1), plt.imshow(img)
plt.subplot(1,2,2), plt.imshow(filtered)
```

3. 结果分析



结果如上图所示，左边为原图，右边为磨皮提亮后的图。可以看到由于调节了 Gamma 曲线，肤色变白了。而且在双边滤波器的作用下皮肤上的斑痕被大幅度减轻，但是还残留少许痕迹，同时眼睛、皮肤以及头发的纹理得以保留。

车牌号识别

1. 图像分析

待处理的图片为阴暗光线下的一辆车，车牌号较为清楚并且与车身其他地方有较大差异。为了提取出车辆的车牌部分，可以使用直方图反投影的方法进行提取，使用直方图反投影法在一张图片中提取区域，首先要提供感兴趣的区域的模板图像。

2. 实验过程

在这里，我们要手动制作模板，考虑到待提取的目标是一个车牌，我们的模板也应该是车牌。



实验主要使用 python 以及 opencv 进行，代码如下：

```

import cv2
import numpy as np
from matplotlib import pyplot as plt

template = cv2.imread('template.jpg')
templateHsv = cv2.cvtColor(template, cv2.COLOR_BGR2HSV)

car = cv2.imread('car.jpg')
carHsv = cv2.cvtColor(car, cv2.COLOR_BGR2HSV)

templateHist = cv2.calcHist([templateHsv], [0, 1], None, [180, 10], [
0, 179, 0, 256])
cv2.normalize(templateHist, templateHist, 0, 255, cv2.NORM_MINMAX)
dst = cv2.calcBackProject([carHsv], [0, 1], templateHist, [0, 179, 0,
256], 1)

disc = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (15, 4))
dst = cv2.filter2D(dst, -1, disc)

ret, thresh = cv2.threshold(dst, 254, 255, 0)

thresh = cv2.merge((thresh, thresh, thresh))

res = cv2.bitwise_and(car, thresh)
res = np.hstack((car, thresh, res))

cv2.imwrite('res.jpg', res)
cv2.imshow('1', res)
cv2.waitKey(0)

```

3. 结果分析

实验结果如下图所示：



左侧为原图像，中间是使用直方图反投影后再进行二值化后得到的蒙版图像，右侧为将蒙版应用于原图以后得到的车牌图像。可以看到提取出来的部分大体是我们想要的区域，但是除此之外还有一部分车体也被提取出来了，这是因为车牌模板的边框以及车牌号中也有银色的部分，与前侧车体的颜色一致。使用直方图反投影无法区分。