

# 口红色号检测器

## 1.6 问题分析

### 1.安装 Dlib 库的问题：

问题描述：在 Python3.7 版本下安装 Dlib 显示无法安装。

解决方法：Python3.6 版本用 pip 命令可直接安装，但 3.7 版本较为复杂，要先安装 CMake 后，再下载相应版本 Dlib 库的安装包进行安装。

### 2.提取颜色时消除黑色背景影响：

问题描述：在提取颜色时若选择阈值  $r=0$  作为判断标准则获得的主要颜色可能为黑色。

解决方法：若直接跳过  $r=0$  的颜色，得到的颜色仍为黑色，因为没有去掉  $r=1$  或其他值较小的黑色部分，所以直接定一个较大的范围  $r<100$ （在实际操作中可以使用一个小一点的值），可以完全去掉黑色背景。

### 3.获得颜色并没有用 H 值表示：

问题描述：色彩的区分是通过 H 值的改变表示的，但是在寻找嘴唇部分的主要颜色时我们没有使用 H 进行表示，而使用饱和度 S。

解决方法：主色调并不是指颜色最多的，而是指最“醒目”的，所以用 S 值来表征。由于唇色会由于灯光等有高亮色，需要对此进行处理，忽略明度过高的部分。

### 4.在寻找邻近色权值的确定：

问题描述：在寻找距离时我们调整了几次权值，发现 1:1:1 时效果很差，但在减小绿色的权值后，效果变好。

解决方法：先以 1: 1: 1 进行寻找一阶距离最小。并逐步调整系数大小使输出结构更接近视觉效果。本组成员认为在确定相似口红颜色的时候，绿色的增加或减少似乎更少的让人感受到颜色的改变，如图 6-1 所示，所以最终决定减少了其中绿色分量的权值。

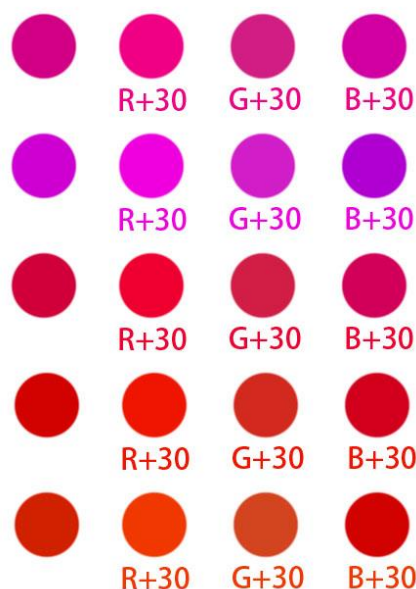


图 6-1 不同颜色分量改变的结果

## 5.HTML 文件中无颜色：

问题描述：在使用正则匹配规则/d{6}匹配 Dior 官网抓取的 HTML 文本时无结果。

解决方法：在观察 Dior 的网页时发现其色号以图片的形式呈现，这导致无法直接输出颜色信息。最终选择用 PS 与 Python 程序结合的方式，获得了 Dior 多个系列口红的颜色。

## 6.自行训练的人脸识别算法性能较差：

问题描述：在项目进行初期，使用 Kaggle 上的数据集训练人脸识别算法时，得到了较差的识别结果。如图 6-2 所示。

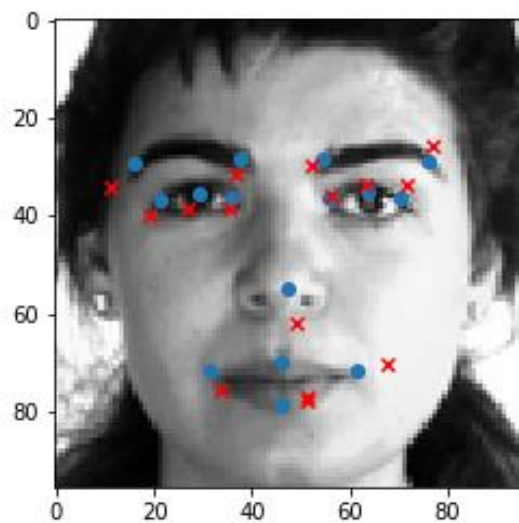


图 6-2 自行训练的模型的识别结果，红色为预测结果

问题分析：首先时标记点数量不足，我们很难找到足量的训练数据，我们找到的 Kaggle 数据集中标记了 15 个点。嘴唇部分只有 4 个点，这本身就限制了我们划分唇部区域。

其次，我们在过程中使用 SAME 方式进行卷积，这样可以保持图的大小在传播过程中不。使用 AdamOptimizer，它是寻找全局最优点的优化算法，我们在算法中只能进行全局的处理，而不能如论文的分块处理局部方式，这可能是我们的模型性能不好的原因之二。

第三，我们的训练速度很慢，处理 50 轮并保存大约需要 6 小时，我们在尝试过程中发现大约 1000 轮才可以获得较好的识别效果，这对于笔记本电脑来说充满挑战。

解决方法：最终我们使用了公认效果较好的 Dlib 库及 face\_recognition 库解决人脸识别标记的问题。