**基于OpenCV、Paddlehub的换发型应用报告**

学号：10205101531、10205101528 姓名：杨政、魏嘉林

摘要：人脸识别和人像分割是指通过对人体轮廓的比较和分析，从而提取人面部特征点或身体各部分分割的方法。本应用受一个基于dlib一键换发型项目的启发，经过分析和反复实验,分别选择了优化模型库face\_landmark\_localization、ace2p，使用OpenCV工具，实现了两种模式的发型交换功能。项目改进并优化了原项目面部融合的算法，对多个实验用例进行测试，比较分析了两种模式的优缺点，并提出相应应用方案。

关键词：目标检测 人脸特征提取 人像分割 颜色校正

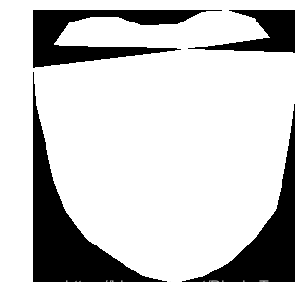
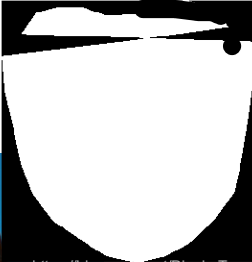
**0.引言**

一般的发型转换实现可分为以下两种：一是基于**人脸识别技术**，提取原图人脸识别关键点，将原始图像的人脸部分剪切，运用仿射变换、颜色矫正等图像融合方法，融合到目标人脸的模式。二是基于**发型分割或人像分割技术**，将目标图像的发型识别并剪切，通过仿射变换、背景修复等图像融合方法，融合到原图像人脸的模式。本报告将就该两种模式的想法由来、优化改进、初步实现等部分逐渐展开。

**1.AI换发型项目的启发**

本项目受到了该AI换发型项目[1]的启发。原项目运用Paddlehub（飞桨）中的face\_landmark\_locallization模型，获得了人脸部68个关键点及其脸部基本轮廓，通过使用OpenCV工具变形待更换人像的脸型，直接嵌入目标人像的脸中，以达到更换发型的目的。

该方法简单、易操作，但存在以下缺点：①面部关键点的检测受人像发型的影响（有无刘海，脸型等），易出现下图中脸型瑕疵的结果；②OpenCV只实现仿射变换和粘贴人脸的效果，当人像肤色、光影效果、饱和度不同时，移植的图像呈现出生硬不和谐的状态。针对该项目的优缺点，本项目学习了它的基本思路，参考其他项目[5]并重写代码，运用了更优良的识别算法和改进移植人脸的过程，弥补其缺陷。

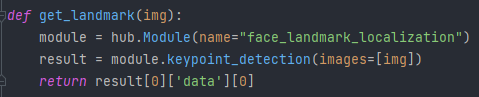


1 识别脸型瑕疵

**2.模式一：基于face\_landmark\_localization的实现**

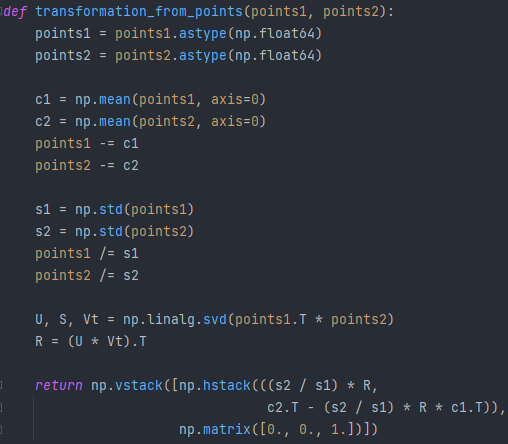
2.1获取面部特征点

在”One Millisecond Face Alignment with an Ensemble of Regression Trees”[2]中，Vahid Kazemi和Josephine Sullivan提出了一个利用回归树集合直接从像素的稀疏子集中估计人脸关键点位置的算法。模型实现了这个算法，模型库的调用很简单，使用命令安装到本地后即可调用。

函数以numpy数组的形式获取图像，并**返回一个68\*2元素矩阵，每一行对应输出图像中特征点的x、y坐标**。该函数是为了获取人像的面部大致轮廓，用于下面的特征融合。

2获取特征点

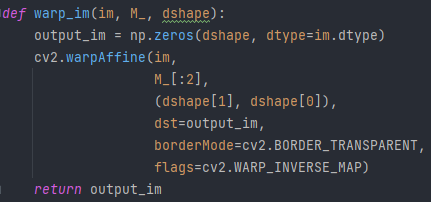
2.2 借助普氏分析（procrustes analysis）完成人脸对齐

在获取两组脸部特征点后，我们需要将脸部进行端到端的对齐，相同的数字标注了脸部相同的特征点。Procrustes analysis是用了分析形状分布的统计方法，即对两个形状进行归一化处理。数学上指出普氏分析就是利用最小二乘法寻找形状A到形状B的仿射变换（平移、放缩和旋转）。

3 获取仿射变换矩阵

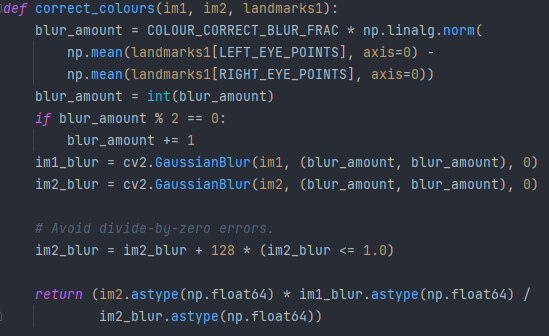
为使第一张人脸对应的向量尽可能对齐第二张人脸对应的向量，算法使用最小二乘法，使变化后所有点与目标点距离和最小。函数代码如上图，分别实现了以下几步：

（1）将输入的68个点的矩阵转换为浮点数，用于后面的运算；（2）每一个点集减去自身矩心cn，一旦为这两个新的点集找到了最佳的缩放和旋转方法，这两个矩心c1和c2就可以找到完整的变换方案；（3）每个点集除以标准偏差sn，这消除了组建的缩放偏差；（4）利用Singular Value Decomposition计算旋转部分(维基百科[3]叙述了解决正交普氏问题的细节）；利用仿射变换矩阵[4]返回完整的转化结果。

得到的结果使用warpAffine函数对目标图像做仿射变换：

4 映射操作

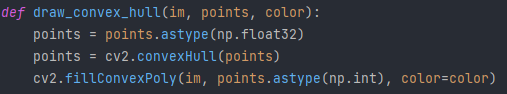
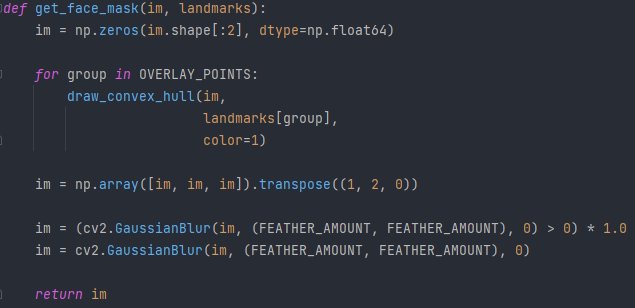
2.3 进行颜色校正

在第二步获取面部特征后，若两人像的肤色和光线不一致，会使重叠区域边缘周围呈现出不和谐效果。因此，采用如下函数做颜色校正。

5颜色更正

该函数通过更改im2的颜色匹配im1的颜色，它将im2除以自身的高斯模糊，再乘以im1的高斯模糊来解决光线差异。**在这个过程中，适当大小的高斯核是关键**，第一张人像的面部特征将出现在第二张人像中，过大的图像或者偏离面部区域的像素将会发生变色，这里函数定义COLOUR\_CORRECT\_BLUR\_FRAC为0.6瞳孔距离的内核。

2.4 融合两张人像的面部特征

面部蒙版用于选择图像融合时，图像哪些部分该在最终图像中展示，值为1的区域对应第二张人像应显示的区域，值为0的区域对应第一张人像应显示的区域，0和1之间的值对应于两张图像的混合

7 获取面部蒙版

6 绘制脸部多边形凸点

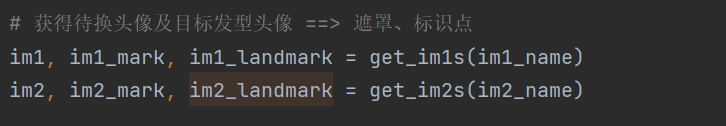
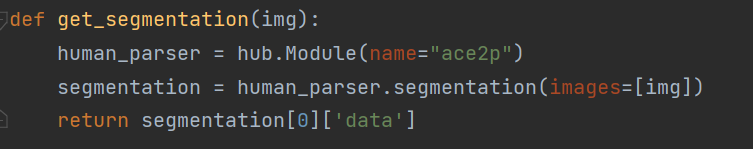
该函数用于生成图像的蒙版和特征点矩阵，它用白色绘制了两个多边形，一个围绕眼部区域，另一个围绕鼻子和嘴巴区域。然后它将遮罩层边缘向外羽化11个像素，用来消除边界的不连续性。两张图像都会产生蒙版，第二个人像的蒙版会被转换到第一个人像的坐标空间矩阵。通过取元素最大值将两个蒙版组合成一个图像，确保第二张人像的特征能显示出来。

**3. 模式二：基于ace2p的实现**

人体解析(Human Parsing)是细粒度的语义分割任务，其旨在识别像素级别的人类图像的组成部分（例如，身体部位和服装）。ACE2P通过融合底层特征，全局上下文信息和边缘细节，执行端到端的人体解析任务。

3.1 获得图片人体解析区域

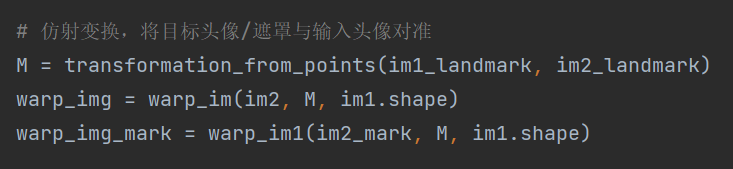
将ace2p模型库用于图像分割，得到人体分割后的区域。调色板给出了不同区域所代表的颜色，本应用中主要针对Hair区域（即返回值为2的区域）进行分割与变换。该步通过get\_segmentation、get\_ims等函数分别得到以下内容：

代换发型（im1）/目标发型（im2）的**图像**，代换发型/目标发型的**图像的遮罩**（将灰度头像的头发部分标记为黑色），代换头像/目标发型的**68个标记点**。

8 Ace2p色调板

9 获取相应内容的函数

3.2 仿射变换，对齐人脸

获取到上述内容后，利用仿射变换将目标头像/遮罩与输入头像对准，仿射变换的过程，上文已给出（**详见普氏分析算法**），不再赘述。

10 对齐人脸

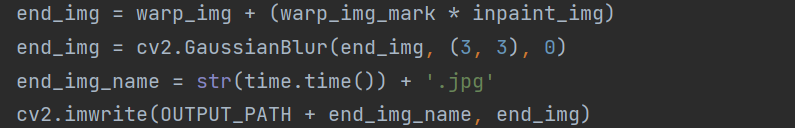
3.3 修复长发变短发后的背景缺失

该步骤利用inpaint函数修复原图像图像边缘，即**流体动力学方法**——沿着已知区域的连续边缘行进到未知区域后，连接具有相同强度的点的线，并在修复区域的边界处匹配渐变矢，填充颜色以减少该区域的最小差异，以使得原图像长发变短发后的背景不显突兀。该函数用到图片待修复区域的遮罩，因此用到了前文得到的im1\_mark结果。

11 补齐背景

3.4 图像融合

该步骤将遮罩（发型部分为(0, 0, 0)，其他部分为(1, 1, 1)的目标头像）与修复后的原图像相乘，得到一个干净的可直接令目标发型头像（im2）于其像素级相加的图像。而后将两者相加即得最终合成结果。

****

12 图像融合

**4.结语**

在实验中，我们弥补了原项目脸型检测和图像融合的不足之处，基于Paddle模型库训练模型，提供了两种换发型的模式：

模式一中，项目运用颜色矫正等算法获得了更好的效果，但在融合**不同种族、肤色、性别及光影效果差别较大**等情况时，结果仍会出现边界感明显、融合差异显著的问题。在该场景下，采用模式二是更为合适的选择；模式二中，针对**长发变短发的情况**，项目仅使用了inpaint算法的方式对背景进行处理，这将导致图像显示效果仍然不佳。因此，在长发变短发的应用场景下，使用模式一不失为明智之举。

以上为本项目报告的全部内容。受限于模型库和边界羽化的不全之处，两种模式虽针对性的解决了某些问题，但仍存在一些欠优化、待完善的实际场景。对系统的模型库及算法实现，项目仍将继续优化更新，以期实现更全面的应用效果。

**附录一 引用文献**

[1]CSDN.PaddleHub创意项目-AI换发型[EB/OL].(2020-8-29)[2022-7-21]. <https://blog.csdn.net/Black_Tguy/article/details/108276972?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522165775828616781683956310%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=165775828616781683956310&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-1-108276972-null-null.142>

[2] Kazemi V, Sullivan J. One millisecond face alignment with an ensemble of regression trees[C]//Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE, 2014: 1867-1874

[3]Wikipedia. Orthogonal Procrustes problem[EB/OL].(2022-4-27)[2022-7-21].<https://en.wikipedia.org/wiki/Orthogonal_Procrustes_problem>

[4]Wikipedia.Transformation matrix[EB/OL].(2022-7-11)[2022-7-21]. <https://en.wikipedia.org/wiki/Transformation_matrix#Affine_transformations>

[5] [Switching Eds: Face swapping with Python, dlib, and OpenCV - Matt's Ramblings (matthewearl.github.io)https://matthewearl.github.io/2015/07/28/switching-eds-with-python/](https://matthewearl.github.io/2015/07/28/switching-eds-with-python/)

**附录二 其他材料**

[1]项目github地址：<https://github.com/yangzheng541/AI_Barbershop>

[2]项目运行手册：

<https://github.com/yangzheng541/AI_Barbershop/blob/master/README.md>