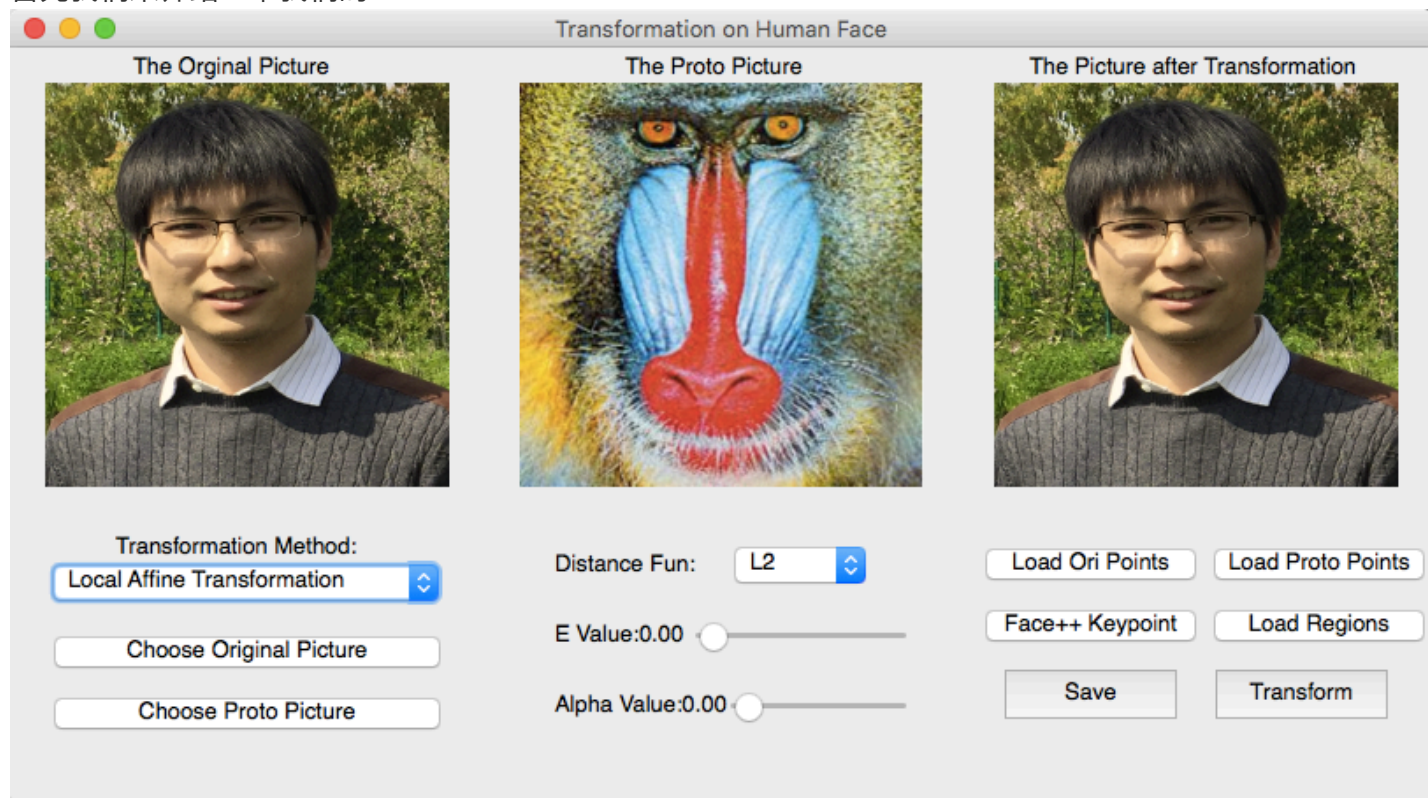


这里我们介绍一下我们本次作业的GUI以及代码的结构还有任务分配

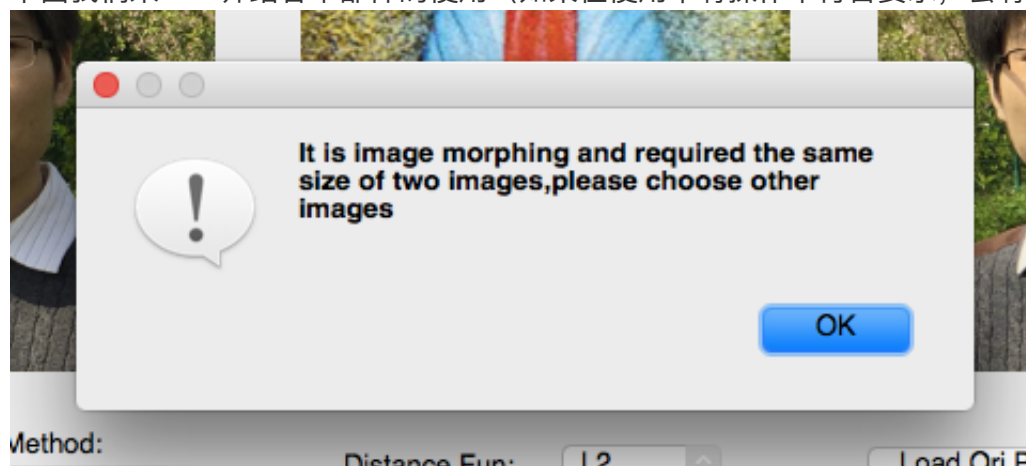
GUI

首先我们来介绍一下我们的GUI

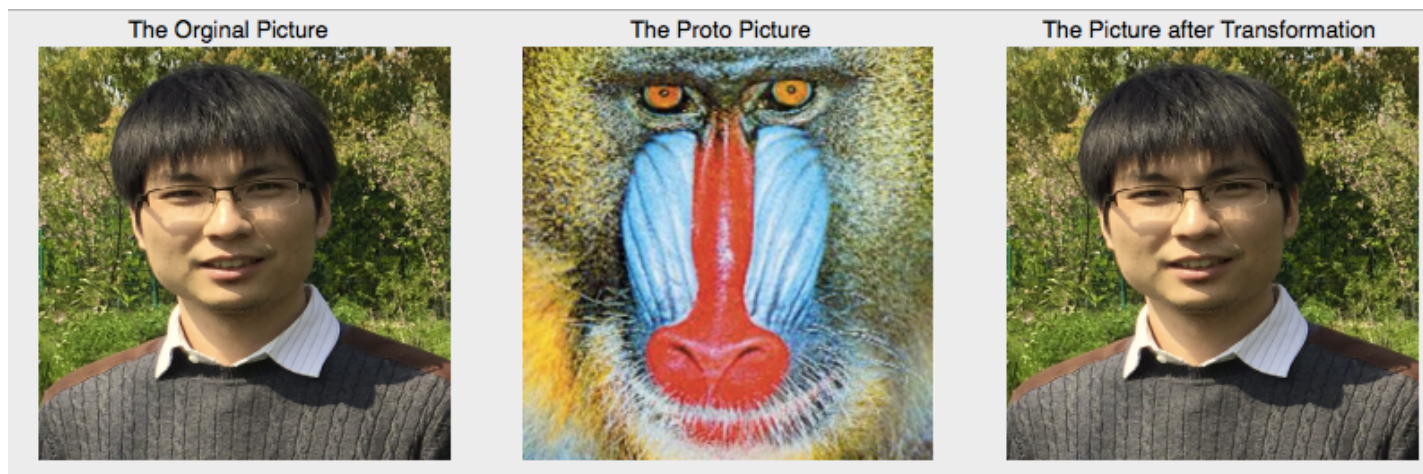


如图，界面如此。

下面我们来一一介绍各个部件的使用（如果在使用中有操作不符合要求，会有提示框）

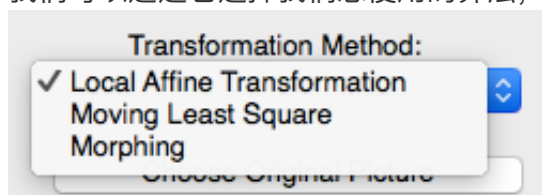


首先上面三幅图分别是原始图，参考图，变换后的图。



左下方

接下来，我们集中在左下方，看到一个选择栏 Transformation Methods 我们可以通过它选择我们想使用的算法，如图



注意，

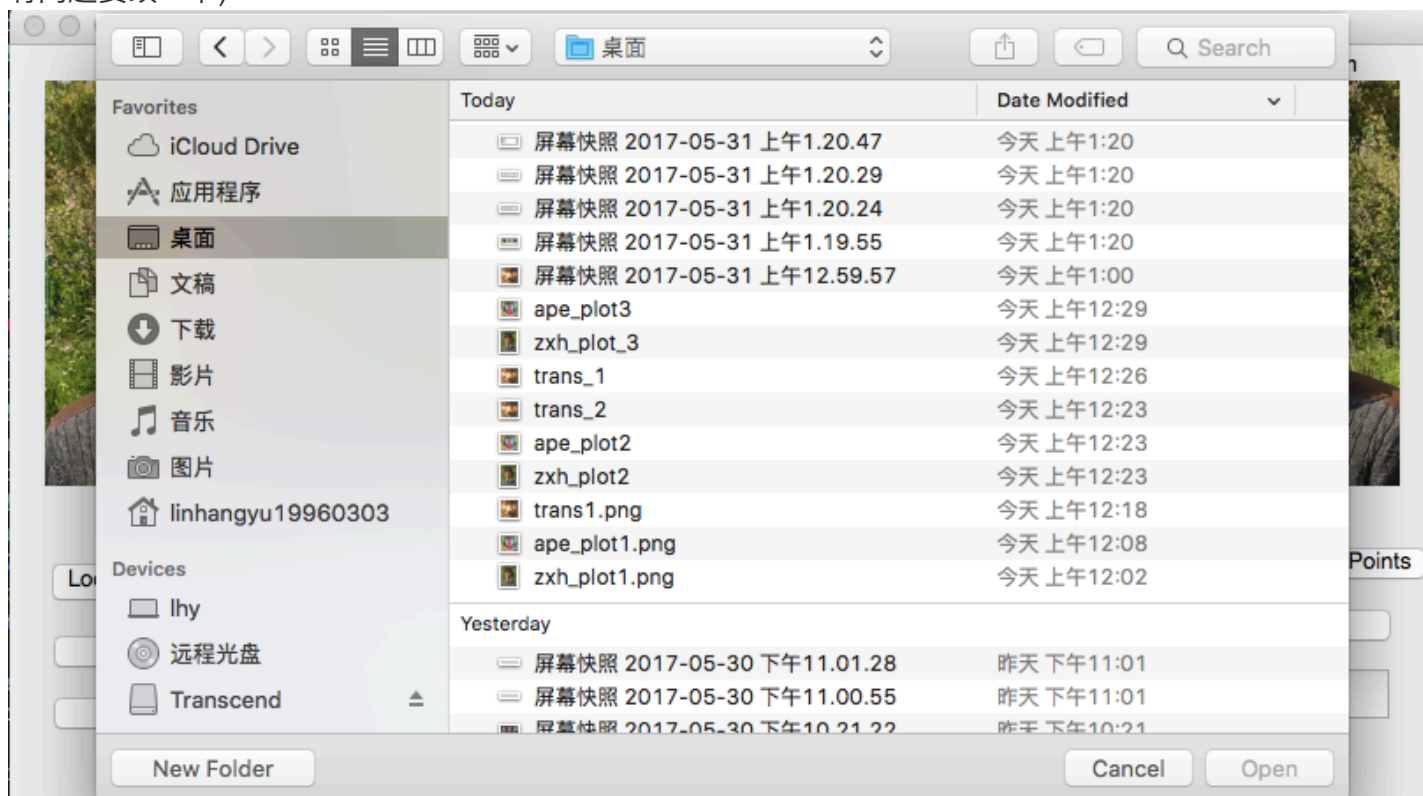
1.morphing方法会自动监测人脸的关键点来进行，所以不需要输入关键点。

2.morphing的时候的proto control point要求跟face++的特征点名称相同。

例子：在当前环境下我们默认是使用LAT算法，如要使用morphing算法，我们要先选择original figure zzh_new.png.

之后我们要加载protopoints 选择 ape_morph.txt，然后再点击transform就可以了

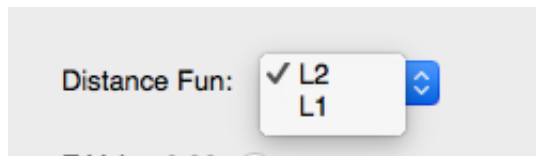
接下来是两个选择图片的按钮，点击这两个按钮会出现文件选择框，如图(注意：windows电脑路径上可能会有问题要改一下)



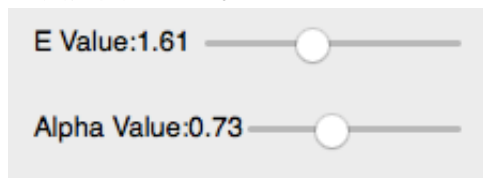
我们限定图片格式，所以只能选指定类型的图片。选择后刷新对应的图片。

中央下方

接下来上方（相对位置）的是距离函数的选择同算法的选择框。



接下来是两个算法参数的slider，在拉动他们会显示他们的数值。要注意，alpha值在不同的算法有不同的范围，所以更换算法时，他们会清零。（当是MLS算法时,alpha从0到2，alpha是控制局部性的参数，当是morphing算法时，alpha表示变换的程度范围是0到1）



右下方

上方的两个按钮



也是文件选择器，但是这里选择的是两个图上的特征点，我们规定要求是txt文件，并且格式同zxh.txt,ape.txt。即一行一个特征点，一行是一个等式，左边是特征点名右边是值不要有括号，要求两个文件有相同的特征点及值。

例子：left_eye_left_corner=270,390

注意：这里面前面的值是横坐标，后边是纵坐标（以图的左上角为原点，水平线位x轴，垂直纤为y轴）跟矩阵形式的index相反（很trick）。

下面的face++和regions按钮



face++的作用是对当前的human face进行特征点检测利用face++api（需要联网）。得到结果以我们的format存储在当前运行环境下命名为 face_keypoints_ori.txt

regions按钮作用是选取control regions这里一行可以是一个或多个特征名，中间以逗号隔开。

例子：

left_eye_left_corner

left_eye_left_corner, left_eye_right_corner

最后的两个按钮



transform是执行变换的按钮，因为所有其他的参数改变都不会及时生效，所以在改变其他参数后，我们需要点击这个按钮来得到变换的结果。

save作用是保存当前形变的文件（会以trans_+原始文件名的方式存在当前目录下）

代码结构

本次代码分两个部分，一个是算法的实现部分，另一个是GUI的实现部分

算法实现部分主要在三个file，transformation.py,morphing.py,util.py. transformation实现了 LAT 和 MLS 算

法，morphing.py 实现了morphing算法 util.py 里是一些工具函数。具体细节可以看注释。
GUI的实现在run.py函数中.里面新建了一个类，细节可以见注释。

任务分配

林航宇：代码实现，报告算法和实验部分

夏景晖：标点，资料阅读查找

马一清：报告的介绍部分，GUI的设计

辛苦老师和助教。

祝安康。

林航宇，夏景晖，马一清