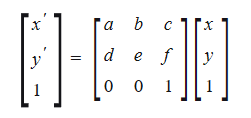
**数据可视化第一次小组作业报告**

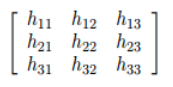
1. **综述**

图像的变换分为多种不同的变换，其中最为基础的便是仿射变换，仿射变换是刚体变换，其不会改变图像的原本存在的平行线，故而得名。仿射变换可以通过一个仿射矩阵来表示，其基本的控制参数有六个，如下图所示：



其中 a、b、d、e 实现对图片的缩放和旋转，c和f则实现对图片像素点的平移。

投影变换和仿射变换十分接近，只不过其将仿射变换矩阵的下面两个零也置换成了控制参数，这样用来实现对图像的不同位置不同形变的操作，如下图所示：



投影变换能够改变图像中原本存在的平行线，因此其不属于刚体变换，属于非刚体变换。

图像配准是一项很有意思的工作，其将原始图片和目标图片的特征分别提取出来，并进行对应解码，然后实现将原始图片转换成目标图片风格的功能。因为不同的特征位于图像中的不同位置，因此初步的图像配准实现可通过图像的局部仿射变换来实现，对于不同的特征（即图像中的不同位置，不同的元素，例如眼睛，鼻子等），我们分别采样若干控制点，然后原始图片到目标图片的控制点，我们可以计算仿射变换的参数矩阵，然后便可实现图像配准的工作。

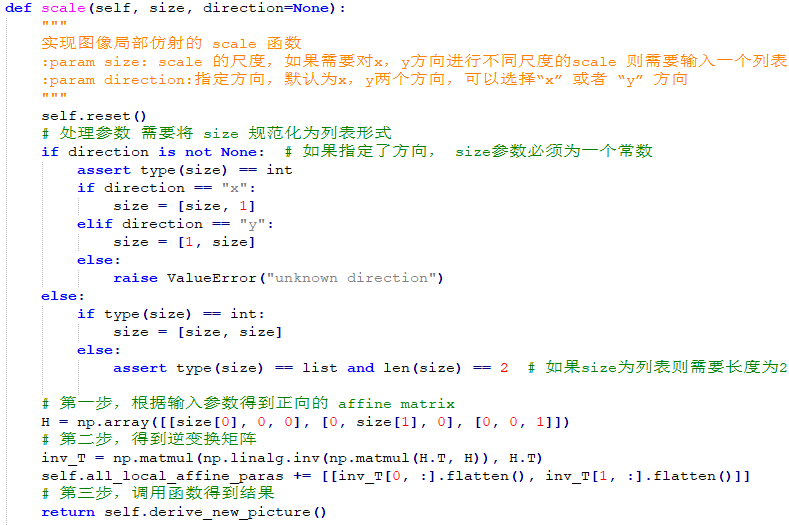
1. **实现**
2. **图像的局部仿射变换**

我们实现了基于反向变换的局部放射变换的4种基本变换（放缩scale、 平移transla、 旋转rotate、 错切shear），该算法将图像的四个顶点固定，并根据选定的局部区域进行仿射变换。算法的基本流程如下：

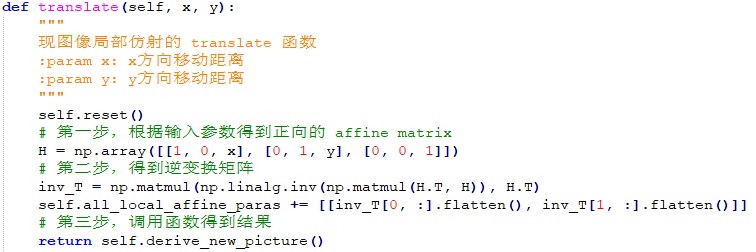
1. 首先确定我们感兴趣的局部区域 ROI，是由一系列控制点构成的。
2. 然后根据我们想要的变换类型以及相应的参数，生成对应的仿射矩阵H。
3. 然后根据逆变换原理 T-1 =(HTH)-1HT 生成逆变换矩阵。
4. 然后再对新图像中每个点进行逆变换得到原图中对应的点， 并插值得到该点的像素值。

其实现的具体功能函数如下图所示：

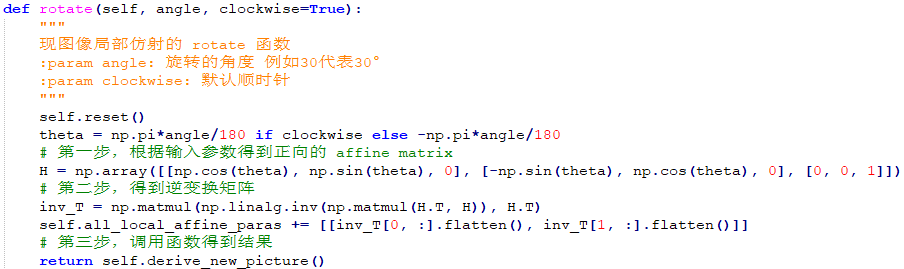
1. 放缩 scale， 实现对局部区域进行放缩，可以从 “x”, “y”两个方向分别进行放缩



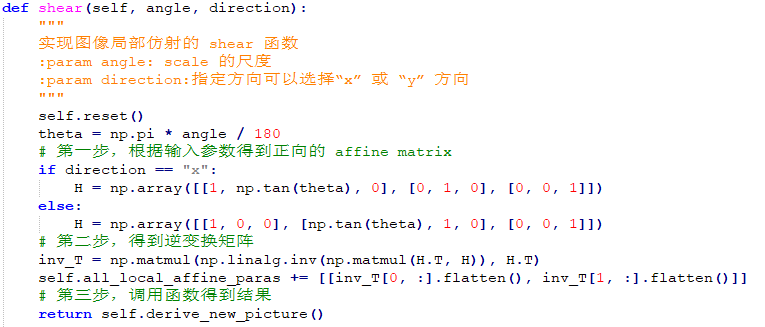
1. 平移 translate， 实现对局部区域进行平移，参数 x, y 分别代表了不同的平移方向。



1. 旋转 rotate, 实现对局部区域进行旋转， 参数angle代表了旋转的角度，clockwise代表是否顺时针旋转。

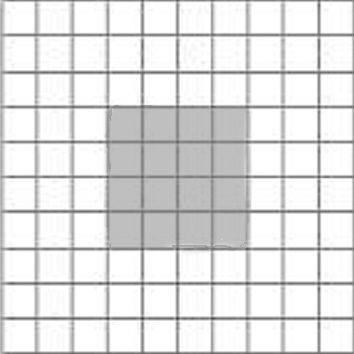


1. 错切 shear， 实现对局部区域进行错切， 参数angle代表了错切的角度， direction代表了错切的方向， 可以是 “x” 或 “y”任意方向。



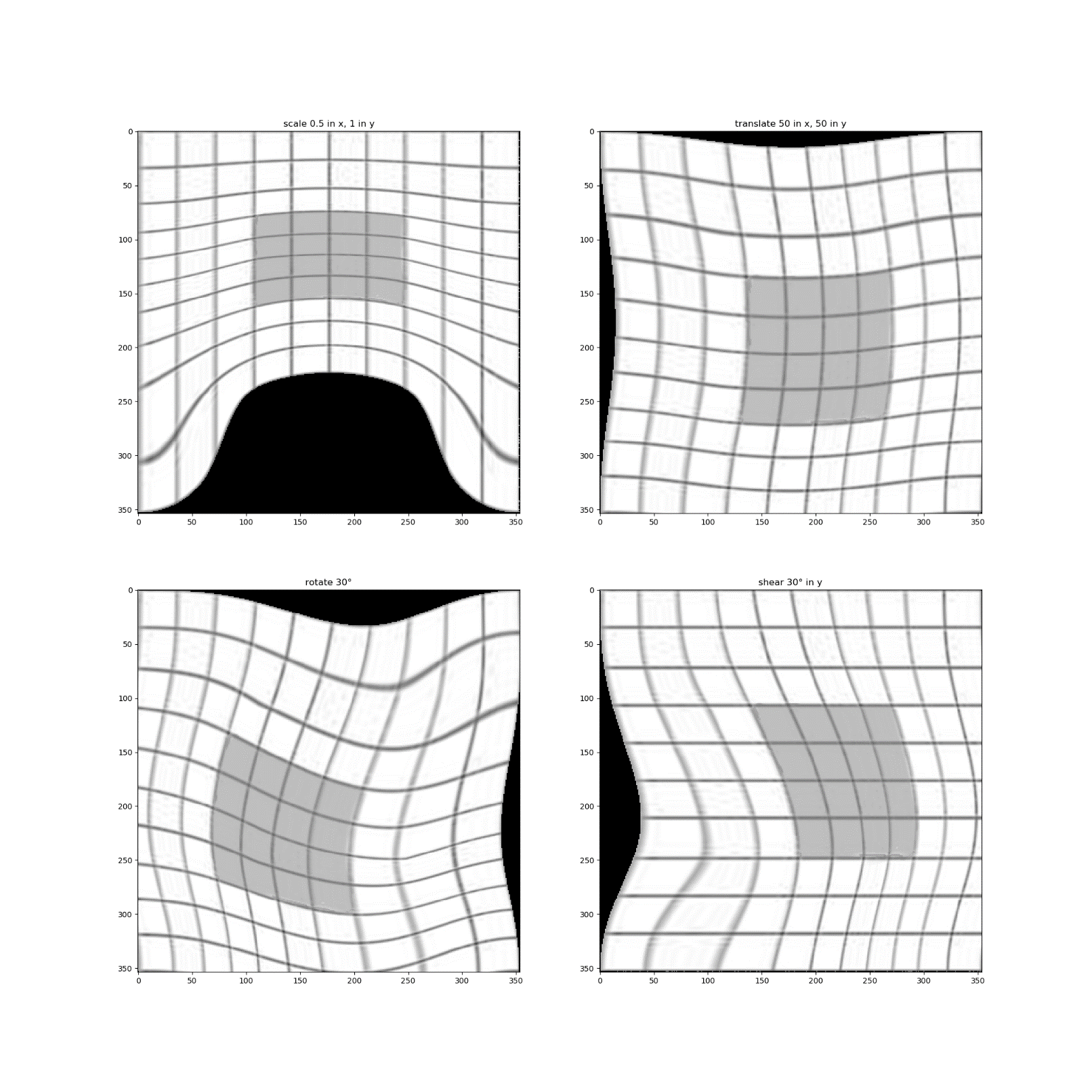
最终得到的效果如下图所示：

原图：



然后对其分别进行4种基本变换：

1. 下图第一行左1为，对原图像“x”方向（即垂直方向）进行缩小0.5倍，“y”方向保持不变。
2. 第一行左2为，对原图像“x”， “y”两个方向的正方形各平移50个像素点。
3. 第二行左1为，对原图像进行顺时针旋转 30°
4. 第二行左2为，对原图像的“y”方向（即水平方向）进行角度为30°的错切。



结论：

从结果可以看出，仅选择控制点进行局部放射变换，并不能达到老师课件中的效果，图像的局部区域仍然会扭曲，边缘部分也会出现一些空洞。主要原因是使用控制点来确定局部区域不太稳定，需要更好的确定局部区域的方法。

*（老师给到的参考文献是：Nonrigid Registration Using Free-Form Deformations: Application to Breast MR Images， IEEE TRANSACTIONS ON MEDICAL IMAGING, VOL. 18, NO. 8, AUGUST 1999）*

1. **图像配准**

我们实现了基于局部仿射变换和反向映射的图像配准算法，用于将人脸图像对准到猩猩的脸部特征上，该算法的基本流程如下：

**得到最终图像**

**对目标图像（图片大小和猩猩的图片大小相等）中的每个点，利用上述求逆后的仿射矩阵计算其在人脸图像中的坐标，然后求取灰度值**

**分别计算从人到猩猩的**

**局部仿射矩阵，然后求逆**

**眼睛、鼻子、嘴巴**

**眼睛、鼻子、嘴巴**

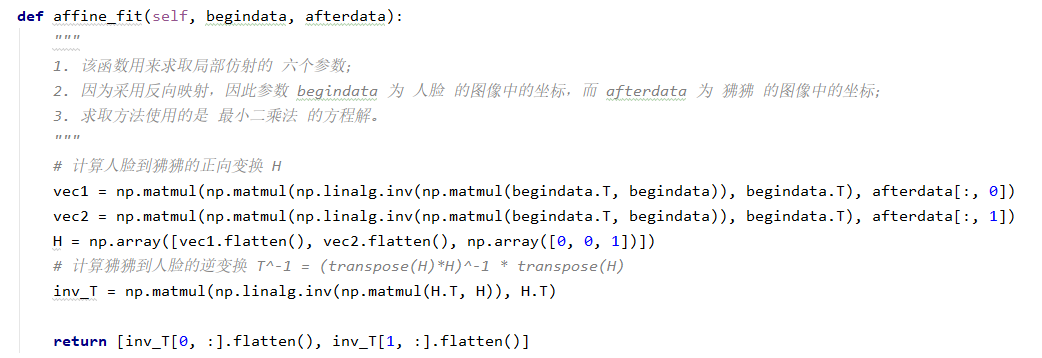
**猩猩的控制点**

**人的控制点**

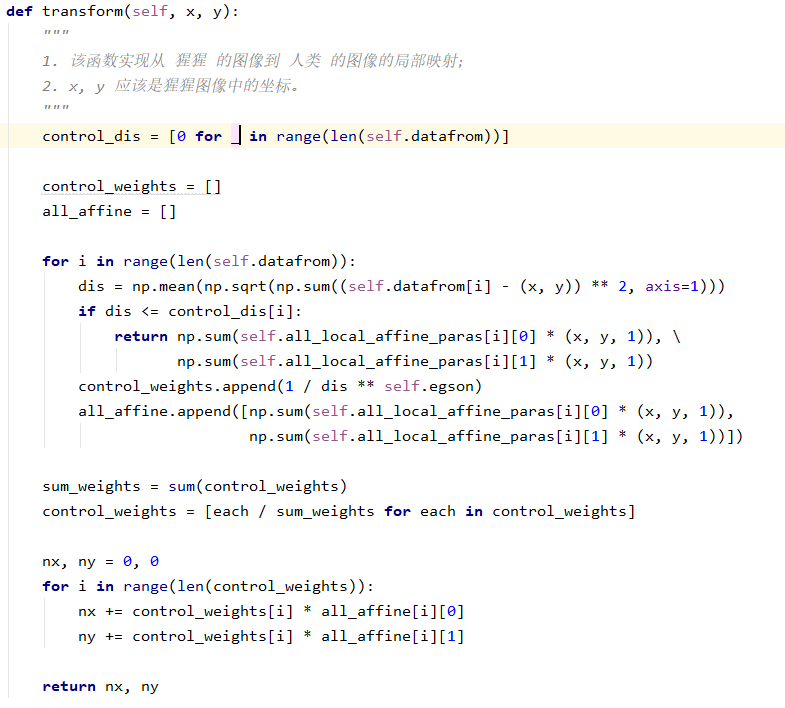
特征选取及对应

其实现具体功能的函数如下所示：

该函数用来求取每一个局部的仿射矩阵，并对它们进行求逆；

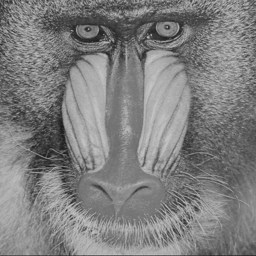


该函数使用上述计算得到的仿射矩阵后，实现从目标图像到原图像像素值的采样；



最终得到的效果如下图所示：

原图片：

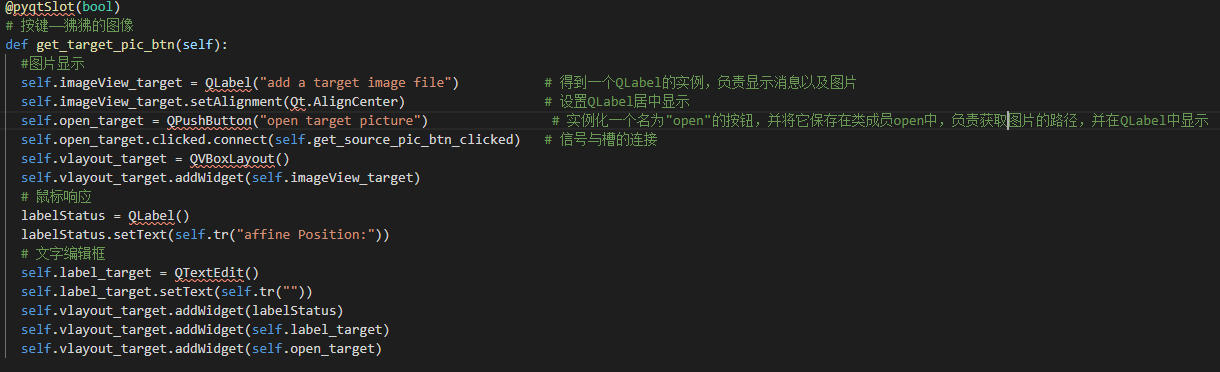
 

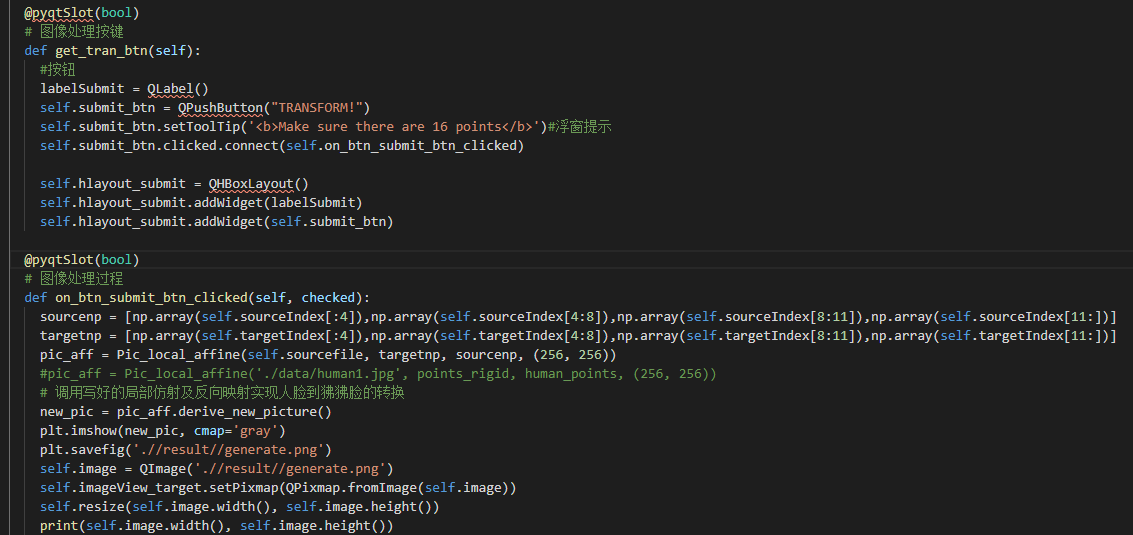
匹配图片（从人脸到猩猩）：

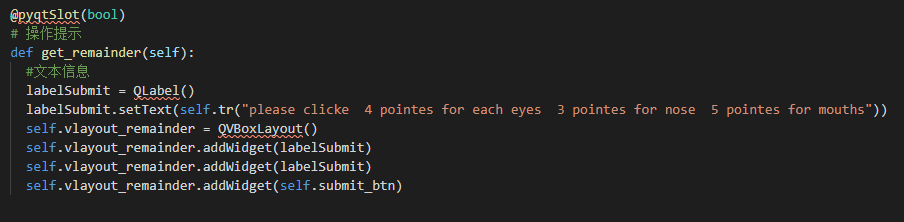


1. **UI界面的可执行文件生成**

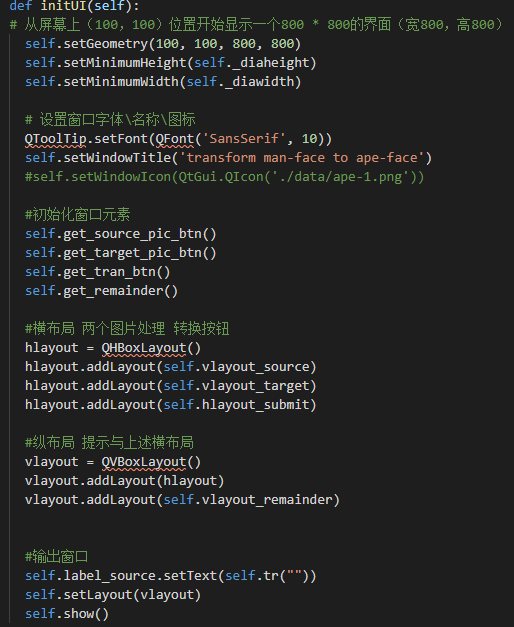
UI界面采用了PyQt5实现，生成exe文件采用了pyinstaller，由于opencv-python与pyinstaller不兼容，所以采用经过改进的opencv-python-headless，同时setuptools 降级到 44.0.0。

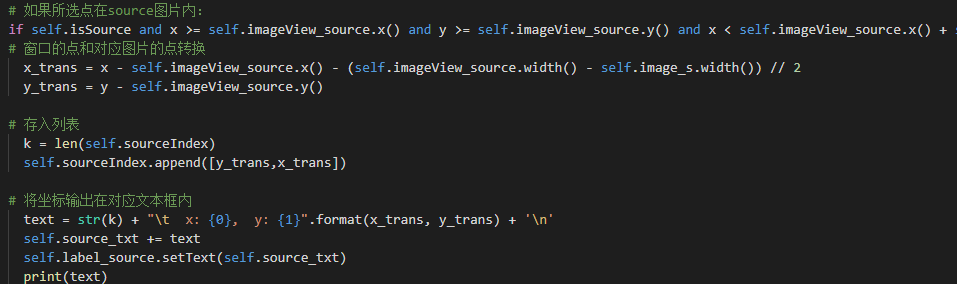
以下是部分图像交互的代码，负责获取图像路径以及显示，同时还有控制点坐标显示

这部分是文件变换按钮的实现以及其具体的调用类操作的方法，其中变换按钮有浮窗提示：

接下是提示信息模块：

整体布局采用的是Layout方法：



鼠标点击事件以人脸图为例，先判断窗口位置，若位于人脸图或狒狒图内，则进行控制点的记录，由于鼠标点击事件的位置是相对于窗口位置，使用我们需要减去偏移得到对应原图像的位置：

1. **执行手册**
2. **打开UI.exe，等待至界面出现**

注意：由于cv2依赖过大，可能需要30秒-2分钟时间不等

1. **打开图片**

按下open xxx pic按钮寻找目录打开对应图片

1. **选择控制点**

依次选择选择仿射点，每只眼睛4个点，鼻子3个点，嘴巴5个点。

注意：请确定选择的数量无误，且控制点一一对应，否则可能程序出错或者出现严重形变等现象，具体可参照演示视频

1. **仿射变换**

按下‘TRANSFORM！’的按键，等待数秒得到放射后的人脸图，结果保存为文件中的generate.png。

1. **代码结构和开发环境**
2. 我们的全部工作都由 python 完成，只有一个源代码文件 img\_transform.py ；
3. 该源文件共分为两部分，第一部分是实现上述局部仿射变换和图像配准等相关功能的函数，一共有三个类，分别是 class ‘base\_LocalAffine’, class ‘Pic\_local\_affine’ 和 class ‘LocalAffine’，其对应的具体功能在该代码文件的注释中有详细讲解； 该源文件的第二部分是测试部分，实现对上述功能的具体测试，并生成相应的结果图片；
4. 所有图片数据均位于 文件夹data 中，该报告设计的所有结果图片均可见于文件夹result中。
5. UI文件调用tran中的类实现，pyinstaler采用了‘opencv-python-headless’和‘setuptools 44.0.0’。img\_transform.py由于有运行部分打包后过大，所以采取了将仿射类单独使用文件tran.py储存。
6. **小组成员及分工**

杨小珪 17307130136 ：参与实现图像配准算法，参与局部仿射算法的改进。

邓文 17307130171 ： 实现局部仿射算法，参与图像配准算法的改进。

董笃翀 17307130154 ：参与局部仿射算法的改进，实现UI界面。