**一．工具及代码介绍**

1. **关键点标注工具 labelme**

输入：待标注图片

输出：对应的标注文档（json格式）

1. **注释文档转换代码**

文件：modify\_json.py

输入：labelme标注的json格式标注文档

输出：menpofit框架要求的pts格式标注文档

1. **关键点检测（训练，测试，预测）**

（1）文件：aam.py（不能保存模型版本）

输入：图片和带关键点的标注文档（pts格式）

输出：关键点预测结果（pts文档）

1. 文件：aam\_train\_test.py

输入：图片和带关键点的标注文档（pts格式）

输出：测试结果及训练模型（pkl格式）

1. 文件：predFromSave.py

输入：待预测图片和训练模型（pkl格式）

输出：关键点预测结果（pts文档）

1. **牛脸校正代码**

文件：warpAffine.py

输入：图像和带关键点的标注文档（pts格式）

输出：校正的图像

**二．使用流程**

**1.训练与测试:**

1. 标注：1（第一部分代号，下同）
2. 注释转换：2
3. 训练与测试：3（2）

**2.预测与牛脸校正：**

1. 预测：3（3）
2. 校正：4

**三．工具及代码详细介绍**

**1.关键点标注工具 labelme**

1. 点击‘open dir’打开标注图像文件夹
2. 右击‘create polygons’进行关键点标注，注意每个图像标注点的个数和顺序要保持一致
3. 点击save 进行保存

**2.注释文档转换代码**

1. main函数：传入source\_path（json格式注释文件夹地址），transf\_path（转换后的pts注释文件夹地址）
2. Transfer\_single：将json文件中points内容（图1）取出来，再写入pts文件（图2）

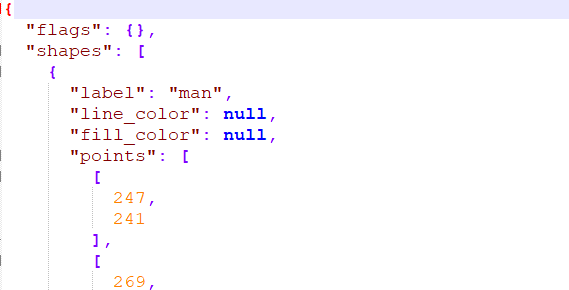


图 1

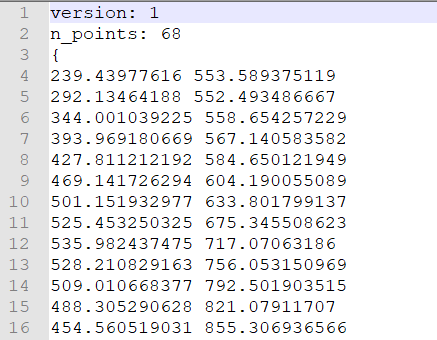


图 2

**3.关键点检测（训练，测试，预测）**

分为训练、测试、预测三部分

1. 训练：

a.将图片预处理(as\_greyscale函数返回图像的灰度版本，crop\_to\_landmarks\_proportion函数将此图像裁剪为围绕一组地标（根据实际情况决定用不用），rescale函数返回此图像的副本，由给定因子重新缩放。 地标被适当地重新调整（根据实际情况决定用不用）;

b.根据训练集建立训练模型aam

C.模型固化，保存文件

（2）测 试 ：

a.将图片进行预处理（上同）；

b.加载ground\_truth,得到变量gt\_s;

c.加载初始化点得到变量s;d.调用fitter的fit\_from\_shape(i, s, gt\_shape=gt\_s)的方法进行测试，打印的结果中包含了正确率

（3）预 测 ：

a.将图片进行预处理（上同）；

b.通过自定义的adjacency\_matrix,points变量，定义PointDirectedGraph对象，将该对象作为预测的初始化参数输入（目前代码功能是整张图片作为初始化）；

c.调用加载的fitter的fit\_from\_bb(i, initial\_bbox, max\_iters=[15, 5],gt\_shape=None)的方法进行预测；

d.将预测结果存成pts文件

**4.牛脸校正代码**

1. main函数：传入img\_dir（图像和注释的文件夹地址），transf\_dir（校正后的图像文件夹地址），注意leftx,lefty,rightx,righty是跟数据真实标注顺序有关系的
2. CropFace函数：通过传入的左右两眼的坐标，计算以左眼为中心需要旋转的角度，并进行剪切工作（代码里注释掉了）
3. ScaleRotateTranslate函数：通过传入的旋转中心和旋转角度，对图像进行旋转