# 关键点检测项目说明

## Stage 1 文件夹

Stage 1文件夹下的模型是采用Resnet网络结构。输入为224\*224\*3。 输出为batchsize \* 42维度的矩阵。这里对数据集进行了相应的处理操作：画出原图上的人脸头像框，然后进行了相关的扩张操作，避免数据集太过密集。然后截取头像框并计算关键点相对头像框的位置，得到相关坐标。并对图像进行了归一化操作，对回归点的坐标做了标准化的处理。

预测的时候得到size \* 42维度的向量，然后我们需要对每张图片得到的42个值进行恢复操作，得到在原图中的相关位置，具体请看Predicts.py中的相关代码。

以下为各文件夹和各文件相关说明：

Datas: 存放数据集、处理后的二进制数据以及待预测图片

DataSets/Image\_covert\_data.py：对数据集处理的工具类

Models: 存放训练好的模型

Detection.py: 训练模型的文件

Network.py: 关键点检测网络

Predicts.py: 预测时执行的文件

## Stage 2 文件夹

Stage 2 训练了五种简单的网络模型，图片输入大小都是39\*39\*3。这五个模型的网络结构文件分别如下：

model\_four\_area.py: 左右眼、鼻子、嘴巴四个区域大概位置预测

model\_left\_eye\_net.py: 左眼六个关键点检测模型

model\_right\_eye\_net.py: 右眼六个关键点检测模型

model\_mouth\_net.py: 嘴巴五个点检测模型

model\_nose\_net.py: 鼻子四点检测模型

predicts.py: 预测文件。需要和Stage 3 结合使用

## Stage 3 文件夹

此文件加入了目标检测的相关模型：MTCNN。首先通过MTCNN检测出图片中人脸所在区域，然后根据五点圈出左眼、右眼、鼻子、嘴巴所在的大概区域，剪裁出相应的图片然后调用Stage 2 中的predicts.py进行预测。结果很好。

以下为各文件夹和各文件相关说明：

Detection: 目标检测代码所在文件夹

Model: 存放模型的文件夹

Output: 预测结果输出文件夹

Picture: 测试图片所在文件夹

Preprocess: 一些数据处理工具类

Train: 模型训练

Video: 视频测试文件

test\_by\_areas.py : 运行此文件此刻得到人脸框和画出21个关键点。

test\_video.py: 能够检测视频中的人脸关键点，但是比较卡吨

一些效果图片：

