演示过程：

在第一排右一服务器（王宇航现在用的那台）上演示，连接摄像头并把摄像头放置在屏幕上方中央，通过主目录/wyl/show，文件夹即可找到演示的代码，其中gaze.txt文件是演示的具体流程，首先会打开openface工具进行视频帧的处理并不断保存在本地，GazeTR模型会监听文件夹并进行视线估计，转换为屏幕坐标后显示在动画中，同时会删除openface保存的历史图像，只保存最新一帧，从而避免内存占用。如果演示完后不及时关闭openface，openface会在后台持续运行并保存图像，导致内存不足，按“q”即可关闭openface。（我也可能记错了，如果不行随时问我）

视线估计流程：

视频帧->数据预处理->视线估计->转换为坐标

数据预处理：

人脸检测->眼睛特征点存储->头部姿势估计->归一化处理->裁剪眼睛图像

视线估计模型：DVFGaze（Dual-view Far-eye Gaze Estimation）双视远眼注视估计方法

转换为坐标：

EVE数据集提供了转换代码

当前演示的是加入了眼睛图像的GazeTR方法，是一种单视角视线估计方法，需要把摄像头放置在屏幕上方中央。

文件夹中的是相机标定代码，使用待标定相机拍摄标定板各个角度的视频，即可使用该代码进行标定，得到相机的内外参数。

DVFGaze代码是本人在DVGaze基础上增加了远眼图像，修改得到的方法，目前无法进行演示，主要原因在于模型的输出是视线方向，需要转化为屏幕上的注视点坐标，这一步目前还存在很大的误差。具体的代码在林政现在用的服务器上。大致路径：主目录/wyl/eth。我在文件夹中也放了这个代码的文件，但由于数据集都在服务器里，可能需要在服务器上运行。数据集路径：data/wyl。双视方法主要用两个数据集：EVE和ETH-XGaze。里面有很多一样的数据集，有些是我重新处理后的，有些是原始的，具体的问题可以再单独问我，我承认我确实写得太乱了，抱歉。

代码中的文件非常多，包括模型、预处理、转坐标、窗口获取的全部流程，并且注释非常少，我也没有写readme，sorry，我的不良代码习惯给你带来了困扰，事实上后期我本人也非常困扰。

大致介绍一下：

Real\_time\_open.py是原本的文件，后来需要修改转坐标的方法之后，出于备份的目的复制了一份：**Real\_time\_opne\_test.py**，所以目前主要是通过这个文件来修改运行。通过该文件调用数据预处理的代码，以及视线估计模型test.py。而在test.py中调用了坐标转换的代码common.py。

目前有问题的代码主要是common.py及其相应的方法文件，Real\_time\_open\_test.py及其相应的方法文件，主要使用的数据集是EVE数据集，数据预处理以及坐标转换参考的代码也是EVE数据集原方法。

EVE链接如下：[EVE/DATASET.md at master · swook/EVE · GitHub](https://github.com/swook/EVE/blob/master/DATASET.md)

目前演示的单视方法中，采用GazeTR\_eye来估计视线，数据预处理使用开源的openface模型直接得到人脸的2D、3D特征点坐标、头部姿势矩阵，Gaze TR原作者提供的坐标转换方法，该方法以四个眼角、两个嘴角六个特征点为人脸中心。

Gaze TR和DVGaze作者为同一人，作者实验室主页：[GazeHub@Phi-ai Lab.](https://phi-ai.buaa.edu.cn/Gazehub/)。而在双视方法上使用这套逻辑误差极大，因此考虑了EVE数据集原作者提供的另一套方法，该方法以右眼中心为原点。

此外，我还自己采集了一组数据集，并进行了相机标定，数据集在王宇航服务器data/wyl文件夹中。

当前问题：为什么双视方法转坐标后误差巨大？

和牛老师讨论后计划先做两个实验：

1. 在单视方法GazeTR\_eye上使用EVE数据集及方法进行坐标转换，分析误差。
2. 在自己采集的数据集使用DVFGaze模型及EVE的数据预处理和坐标转换方法，测试在自己的数据集上的误差。

牛老师还希望仔细研究一下坐标转换的代码，分析是否存在问题。