**模型创新**

**原有的模型架构分析**

该项目包含主要的四个神经网络模型:  
 生成器模型：  
 1. 初始化：一个DenseMotionwork网络，其内部使用了Hourglass这一网络架构：通过多层卷积的下采样（Encoder）提取特征和上采样（Decoder）恢复分辨率的对称式结构。下采样阶段降低了图像分辨率，通过逐步的下采样块，指数式地增加通道数，获得图片的高级抽象语义信息。上采样阶段恢复特征图大小，将上采样后的特征图与下采样（编码器）的跳跃连接特征图拼接，使得网络结构能包含全局信息。  
 2. 前向传播：构建一个初始化卷积块，使用下采样块压缩分辨率，接着通过一个卷积块限制通道数，再展开为3D特征图并进行处理，通过DenseMotionwork网络估计密集运动和计算变形与遮罩层进行处理，最后通过上采样解码。

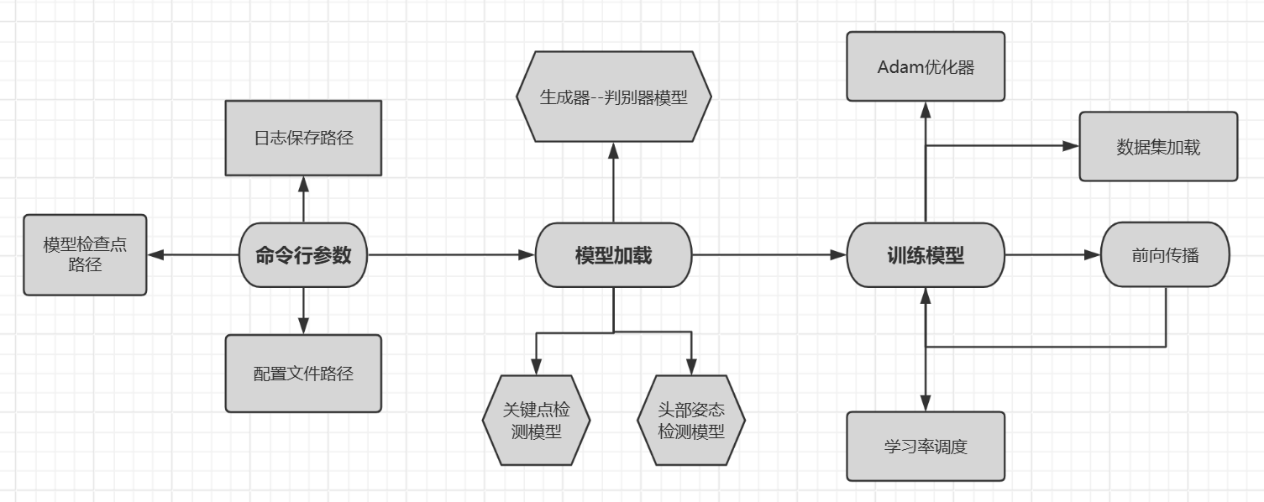
判别器模型（MultiScaleDiscriminator）：  
 主体类似Pix2Pix，通过对输入图像进行下采样提取特征，再进行与生成器类似的压缩分辨率增大通道数，最后通过卷积层输出多尺度特征图。在此基础上将单个判别器更具不同尺度进行组合，获得不同尺度下的特征信息（虽然最后只用到了一个尺度）

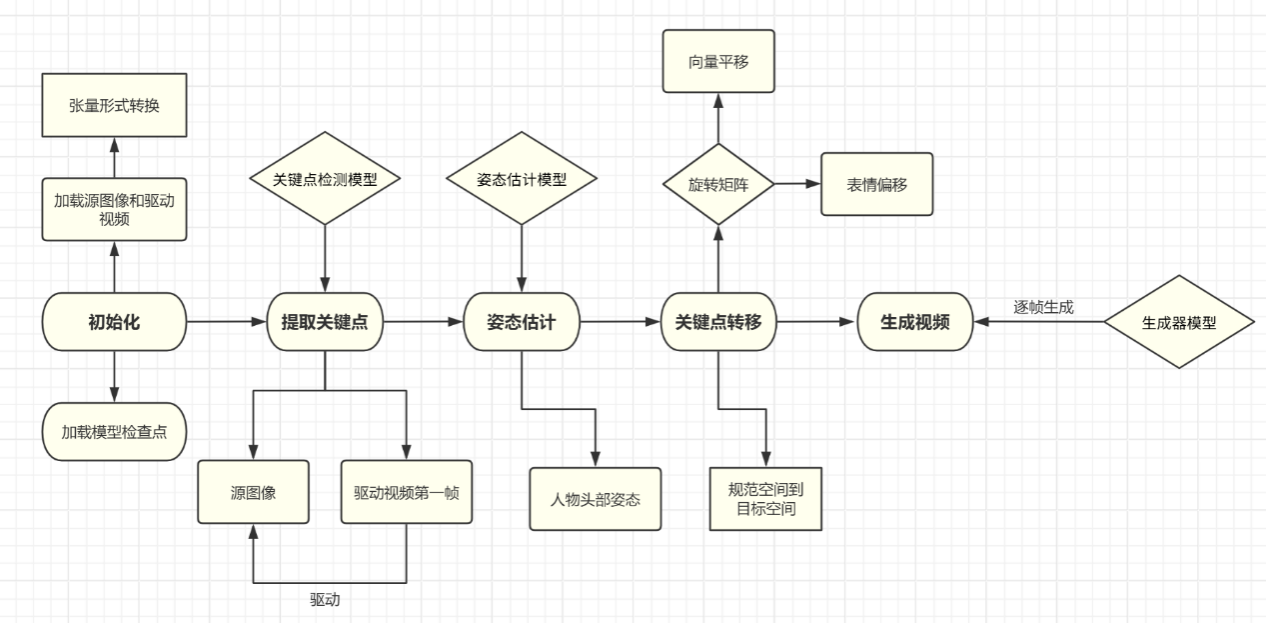
生成器和判断器构成了生成对抗网络（GAN），二者相互竞争，增强生成器生成视频能力的同时也不断提高判别器的分辨能力。

关键点检测模型（KPDetector）：  
 1. 生成一个Hourglass架构模型，通过连续下采样压缩空间并限制通道数，再通过卷积层变换通道数，重塑特征图维度后调整形状，最后使用上采样块恢复空间分辨率。  
 2. 建立3d卷积层，根据预测的关键点输出热图，通过在关键点周围生成高斯分布表示关键点的概率（高斯热图回归），将特征图关键点位置转换为概率分布，进而增强平滑性和鲁棒性，并在归一化时调用softmax函数控制热图的平滑度。  
 3. 估计雅可比矩阵，再使用一层卷积层从特征图中生成雅可比矩阵，将矩阵和对应热图点乘计算加权平均，获得关键点周围的雅可比矩阵，表示局部变换。

1. 前向传播：在缩放因子不为1时首先使用高斯卷积核进行下采样，再通过Hourglass架构的关键点捕获多尺度的特征，生成特征图后再通过3d卷积层整形后获得热图，归一化并重塑后再从热图中提取关键点的均值，最后按需计算雅可比矩阵。

姿态估计模型（HEEstimator）：  
 通过一系列的卷积层提取输入图像特征，再通过残差网络提炼特征，最后通过全连接层估计姿态。

训练脚本(train.py):  
 首先定义命令行参数，包括配置文件路径，日志保存路径，模型检查点路径等。接着设置了生成器模型、判别器模型、关键点检测模型和头部姿态检测模型，最后加载数据集进行训练。  


测试脚本(demo.py):  
 首先初始化，需要加载源图像和驱动视频，同时加载模型检查点，将源图像和驱动视频转换为张量形式。接着使用关键点检测模型检测源图像和驱动视频第一帧的关键点，从而实现基于驱动视频的关键点运动驱动源图像的动画。然后使用姿态估计模型获取人物的头部姿态，通过旋转矩阵对关键点旋转后再接平移和表情偏移，将关键点从规范空间姿态转换到目标姿态（可选自由视角），从而模拟人物的三维图像转动而不是仅限于二维。再提取驱动视频每一帧的关键点，同样经过姿态转换到目标空间，最后再进行标准化，通过生成器模型即可生成对应视频。  


**模型创新**

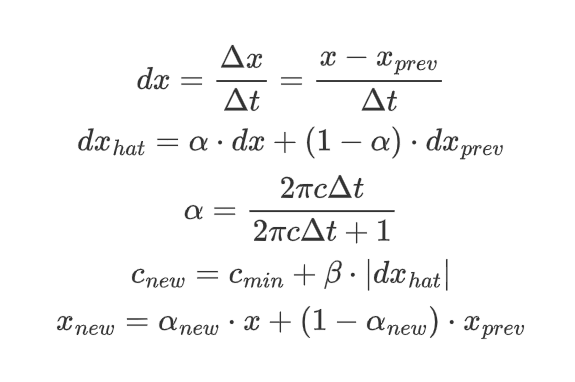
我们尝试去更换原有的关键点检测模型，选择了参照该项目的人脸关键点检测模型（https://github.com/you-old/FaceLandmark1000）。其主体结构包括两部分：人脸检测框与人脸关键点识别，分别实现了对人脸边框的检测和在此基础上的关键点检测，其项目流程如下:

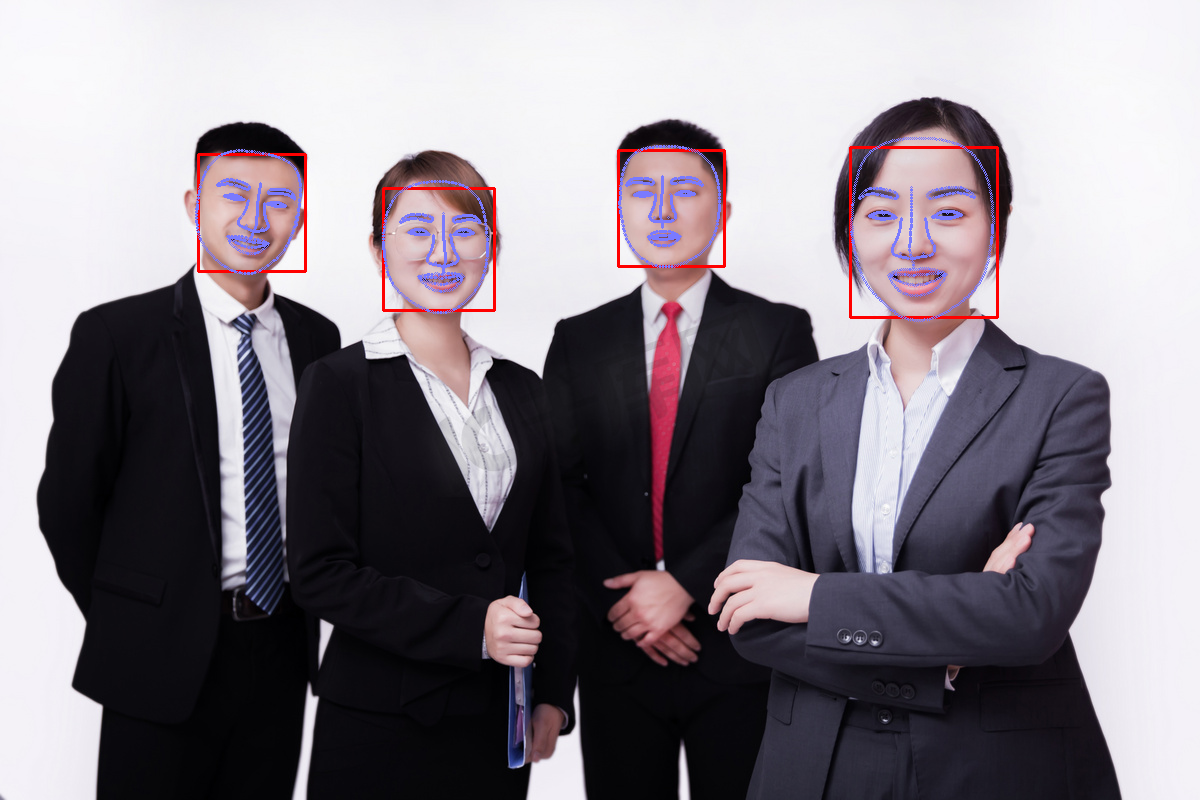
人脸检测器：

1. 预处理：将原图变换为640\*640的图片，计算缩放因子并转换为torch张量，同时标准化减去BGR均值（104,117,123），最后调整通道顺序以符合pytorch的标准输入。
2. 模型推理：使用onnx加载模型，获得预测框相较于先验框的偏移量，置信度和关键点位置。
3. 后处理：创建先验框，在不同特征图下获得所有先验框的位置大小信息，再通过解码预测框和先验框，乘以缩放因子，恢复真实图像的边框和关键点信息，接着筛选出置信度较高的边框和对应关键点，使用非极大值抑制（NMS）去除冗余边框，最终保留靠前的部分结果。
4. 恢复边框的位置，从标准化的图片映射到原图中。

关键点检测器：  
 1. 预处理：将原图片通过之前人脸边框进行裁剪，获得脸部的部分，但也扩展了边界以保证获取完整信息，最后将裁剪后的图片调整为128\*128的格式并转换为灰度图。  
 2. 模型推理：使用 onnxruntime 创建推理会话，获得关键点的位置。  
 3. 后处理：将关键点从归一化转换为原始图像信息。

4. 平滑算法：比较当前帧和上一帧的关键点集合，如果二者的IOU（交并比）超过阈值，则进行平滑处理：如果二者的欧式距离小于阈值，则直接使用上一帧的关键点集合，否则使用OneEuroFilter进行动态平滑。  
 **NMS算法**：  
 从所有检测框中获取可信度最高的一项，依次计算每个剩余框和最高分框的交并比（按照面积），最后返回所有交并比小于阈值的框，从而获得尽可能不重叠的可信度高的检测框。  
 **OneEuroFilter动态滤波器：**

计算信号变化率dx，从而获得当前变化率和上一变化率的加权平滑值dx\_hat（导数）。依据该值按一定学习率beta更新截止频率cutoff,最后按同样的方法加权计算信号的平滑值即可。  


对替换后的模型，我们还进行了额外的错误处理情况，解决了未检测到人脸返回None的情况和多个人脸检测的情况，确保在绝大数情况下能正常运行。运行结果如下：  


最后给出两个模型的比较：

