1. 完整算法结构框图、思路步骤详述、代码组织结构介绍

思路：

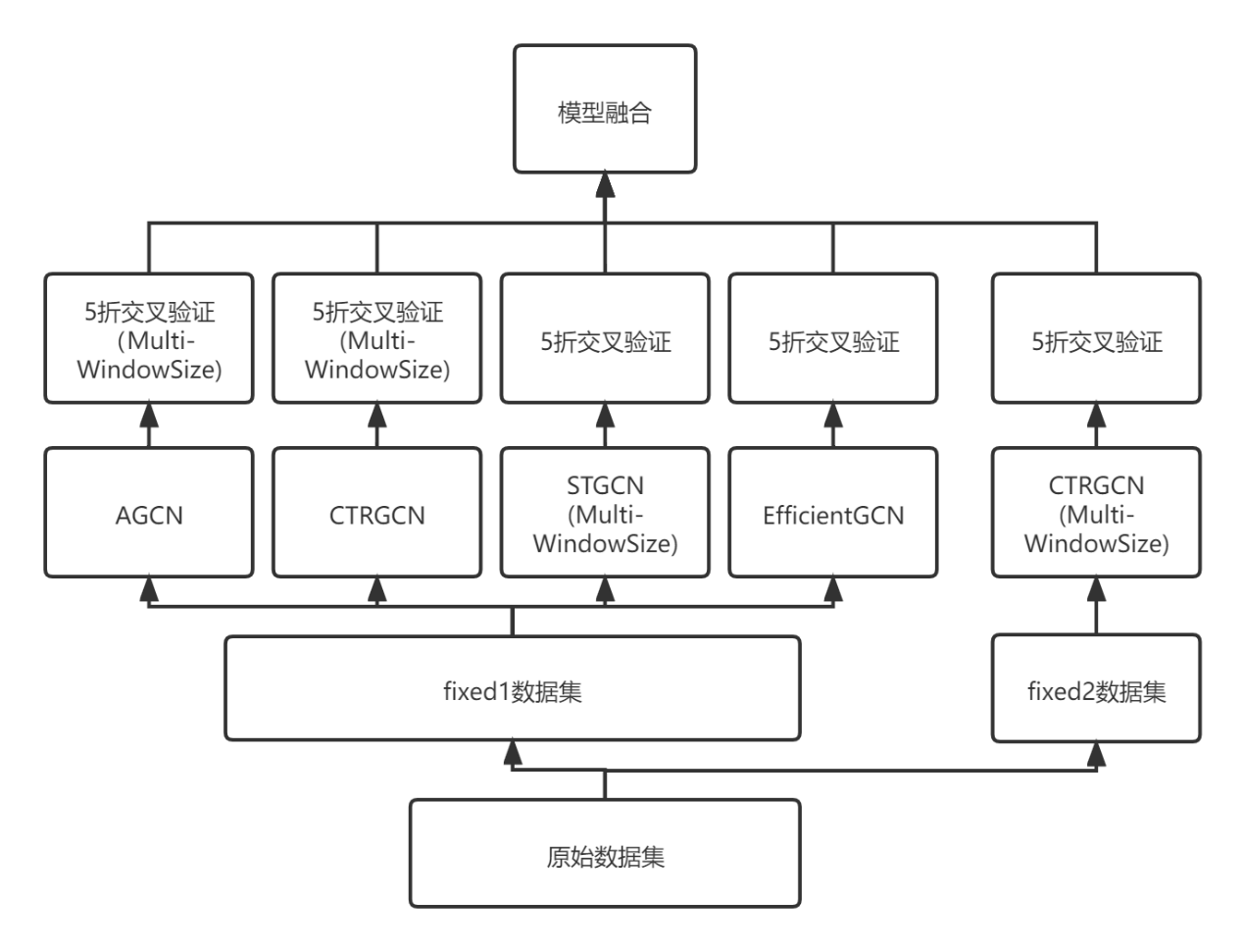
在数据集上，我们可视化并认真分析了数据集的特点，认为数据集可能存在中心化失效的情况，并对此进行处理。此外，我们尝试了多种可能适用的数据增强策略，如AutoPadding，CutOut，Mixup等，通过优化，丰富了数据多样性，提高了模型泛化能力，减小了过拟合。

在模型上，我们成功在PaddleVideo上复现了AGCN，STGCN，CTRGCN，EfficientGCN等模型。

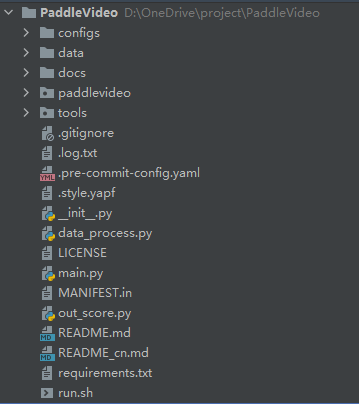
在训练上，我们采用5折交叉验证训练模型，在上述复现的模型上进行了多次实验并分别选取最优的模型，同时我们使用了Multi-WindowSize策略，提升了模型鲁棒性。

在测试上，我们使用了TTA策略进行推理，并融合所有模型的结果。

结构框图：



代码组织结构：

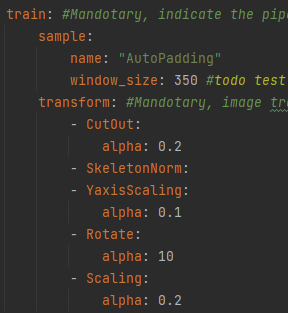


data\_process.py中实现了数据预处理。

main.py中实现了训练和测试的接口。

out\_score.py中实现了模型测试结果的融合。

1. 数据增强/清洗策略；



在训练过程中，我们使用了如下的数据增强策略：

CutOut：对随机选中的骨骼点数据做置0处理

YaxisScaling：在Y轴上对数据做缩放

Rotate：随机对数据进行旋转

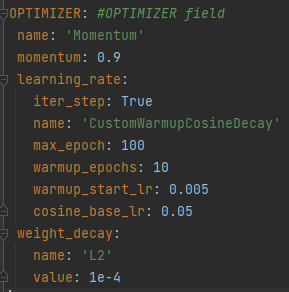
Scaling：随机对数据进行缩放

1. 调参优化策略

AGCN：

epoch：100

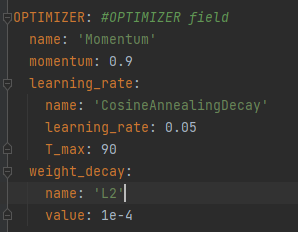
batch\_size: 64



STGCN:

epoch: 90

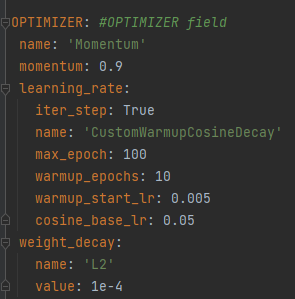
batch\_size: 64



CTRGCN:

epoch:100

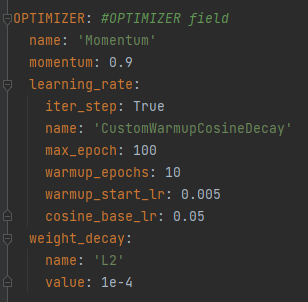
batch\_size:64



EfficientGCN:

epoch: 100

batch\_size: 16



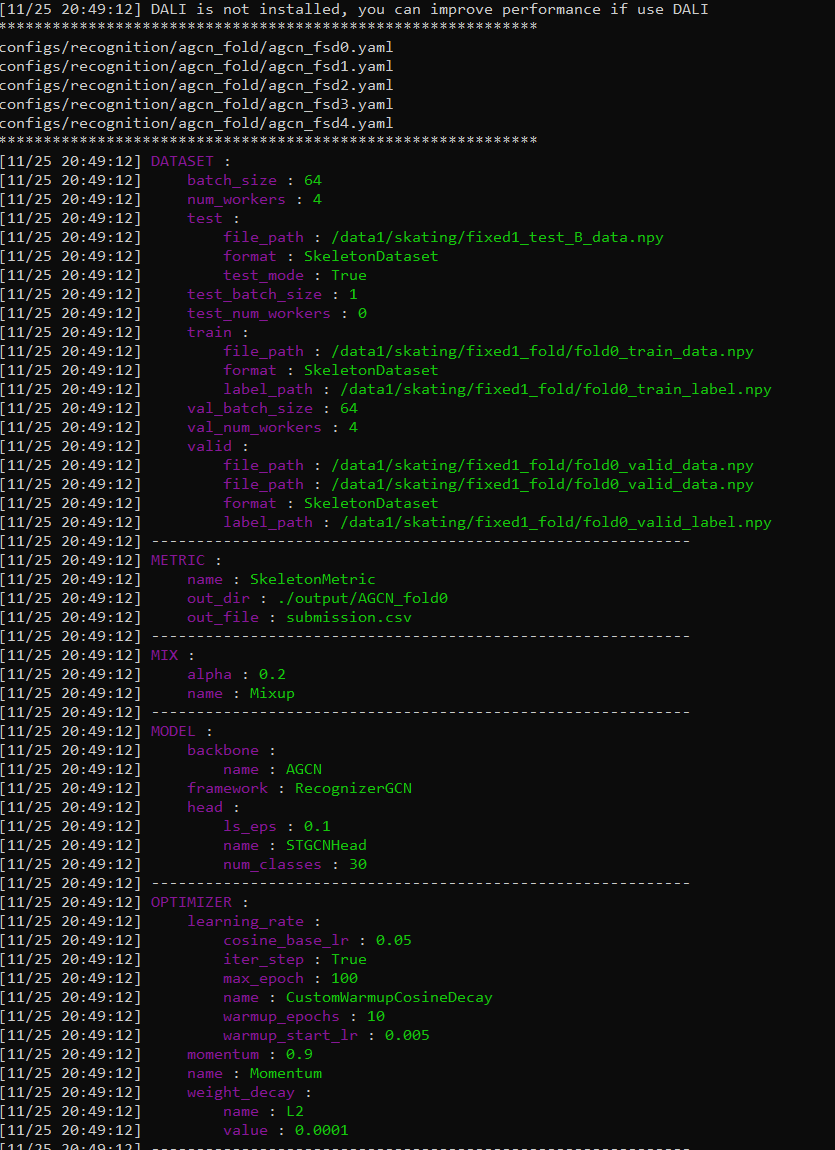
1. 训练脚本/代码

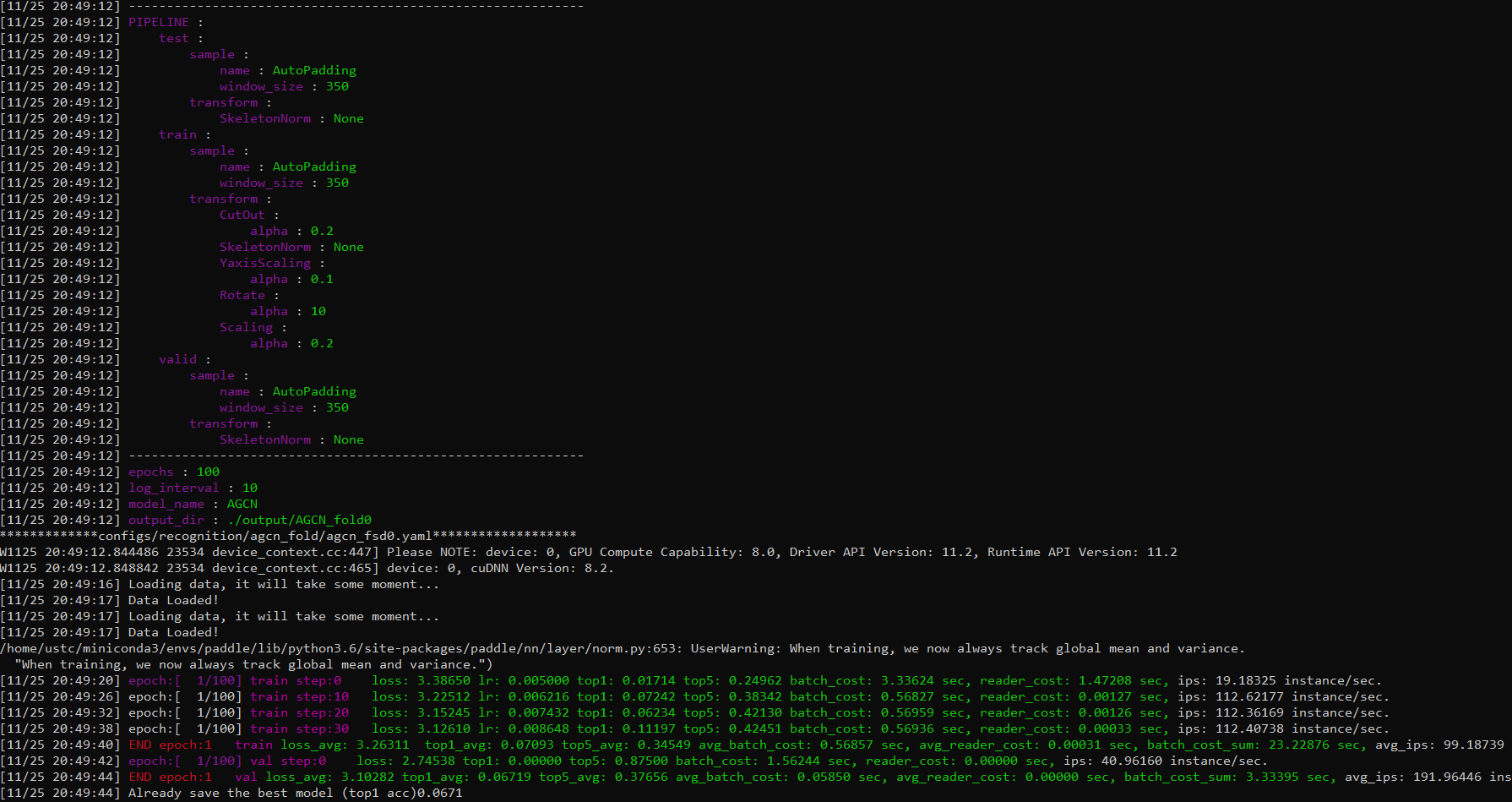
在训练前需要执行数据预处理



训练命令：  


训练日志：



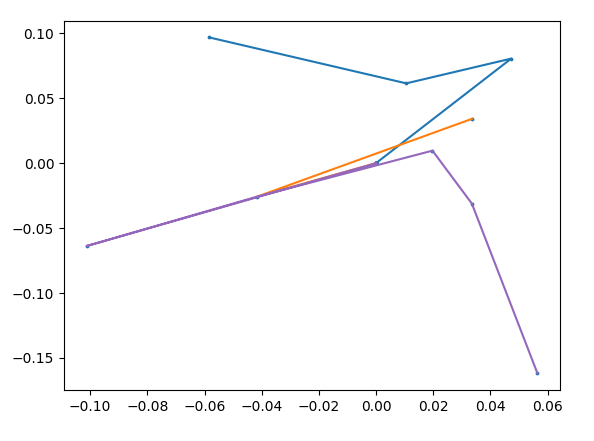
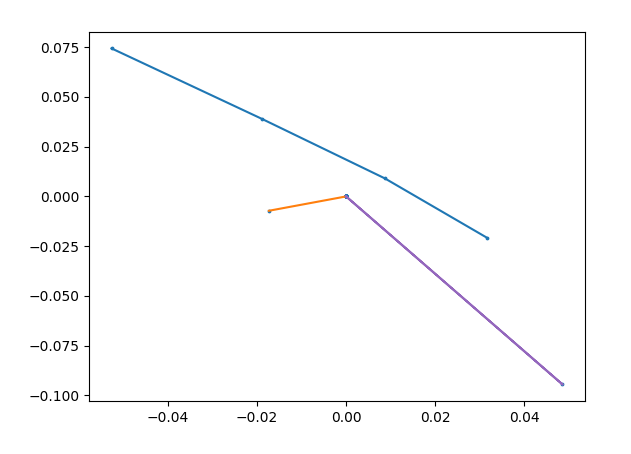


1. 测试脚本/代码



日志由于过大，存放在同级目录中

1. AI模型开发过程、训练技巧、创新思路等其它需要说明的内容
2. 我们尝试了对数据进行数据清洗，但经过实验发现效果并不理想，最后没有采用。
3. 我们对数据进行了可视化，并发现有大量置信度为0的点我们搜索并删去了置信度全为0的帧，并考虑到中心化时中心点8号可能会失效，对其进行了修复，策略是取其他置信度非0点的中心。

1. 我们实现了在骨骼点上的CutOut数据增强策略，随机对骨骼点进行遮挡。
2. 我们尝试了在骨骼数据上使用CutMixUp，但经过实验发现效果并不理想，最后没有采用。
3. 训练中采用的Multi-WindowSize：我们在每一个epoch都随机使用了一个不同的window\_size，来保证模型的鲁棒性。最后对使用该策略的模型我们采用了[300,450]的window\_size
4. 我们用5折交叉验证和直接使用训练集进行训练，对比结果后最终采用5折交叉验证的策略。
5. 在训练过程中，我们尝试使用weighted sampler 来解决样本类别不平衡问题，但效果一般
6. 在训练过程中，我们尝试对loss进行过滤，对loss过大的样本考虑其有噪声，减少对其的学习，但效果一般
7. 运行前请校对data\_process.py中的数据路径，以及configs/recognition中yaml的数据路径。在运行时，显存不够请调小对应模型batch\_size

注意：我们将训练好的权重放在了数据集output\_backup中，在使用权重复现时，需要将

其内容复制到PaddleVideo/output中进行覆盖，并且注意，由于提交的时候失误，在用权重复现时，需要将ctrgcn\_fold中每一个YAML上面的name:’CTRGCN2’改成name:’CTRGCN’,正常使用时该回去。