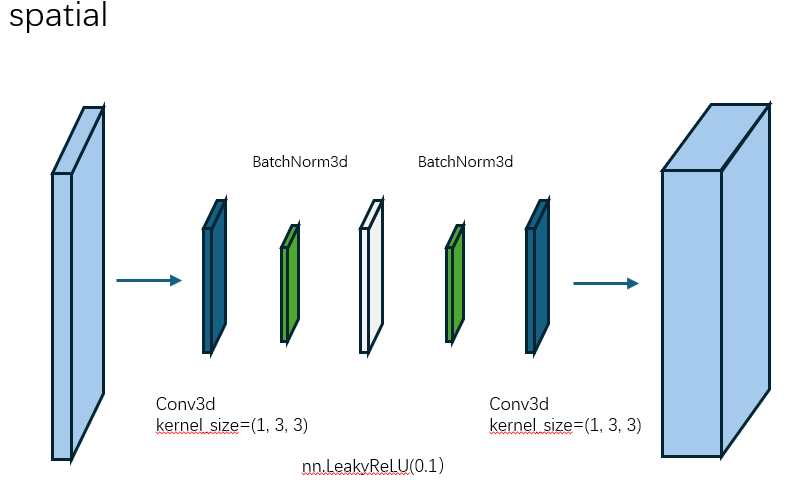
## 算法实现

由于基于时间序列对视频文件进行微表情识别的开源项目较少，本小组努力过后没能复现任何开源算法，因此决定自己制作!

为了提取时间序列中的微表情特征，首先提出一个时空间分块处理模块。通过对conv3d卷积层中的kernel\_size参数控制网络层分别在空间维度卷积和在时间维度卷积。

 图示

AI 生成的内容可能不正确。

再利用残差连接，构建时空间网络层

图示, 示意图

AI 生成的内容可能不正确。

通过Conv3d、MaxPool3d等常用卷积层与SpatioTemporalBlock网络层构建主干网络，该架构采用渐进式下采样策略，最终经自适应时空池化(Adaptive SpatioTemporal Pooling)将维度规约至(\_,1,1)，接续LSTM时序建模与全连接层，最终得到形状为（B,7）的张量，推断视频所属分类。

## 2、损失函数

使用了torch自带的torch.nn.CrossEntropyLoss()函数

该函数计算输入logits和目标之间的交叉熵损失，即用于衡量模型输出的分布与真实标签的概率分布之间的差异。

**输入** ：

input ：模型的输出，形状为 [*N*,*C*]，其中 *N* 是样本数量，*C* 是类别数量。它通常是经过 softmax 函数之前的 logits 值。

target ：真实标签，形状为 [*N*]，其中每个元素是一个整数，表示样本所属的类别索引，取值范围是 [0,C]

**输出** ：

一个标量值，表示损失值。它衡量了模型输出与真实标签之间的差异。

**3. 计算过程**

torch.nn.CrossEntropyLoss() 的计算过程可以分为两步：

**softmax** ：对模型的输出 *x* 应用 softmax 函数，将其转换为概率分布。softmax 函数的公式为：

其中，是模型输出的第 *i* 个元素，*C* 是类别数量。

**交叉熵** ：计算 softmax 输出的概率分布与真实标签的概率分布之间的交叉熵。交叉熵的公式为：

其中，​ 是真实标签的概率分布（在多分类问题中，真实标签的概率分布是一个 one - hot 编码向量，即只有对应类别的元素为 1，其他元素为 0）， 是 softmax 输出的概率分布，*N* 是样本数量，*C* 是类别数量。

## 3、评价指标

**精确率（Precision**）

精确率指模型预测为正的样本中实际也为正的样本占被预测为正的样本的比例。

计算公式为：

其中TP表示实际为正被预测也为正的样本数量，FP表示实际为负但被预测为正的样本数量。

**召回率（recall）**

召回率指实际为正的样本中被预测为正的样本所占实际为正的样本的比例。

计算公式为：

其中TP表示实际为正被预测也为正的样本数量，FN表示实际为正但被预测为负的样本数量。

**F1 score**

F1 score是精确率和召回率的调和平均值。

计算公式为：

其中Precision表示精确率，recall表示召回率。

Precision体现了模型对负样本的区分能力，Precision越高，模型对负样本的区分能力越强；Recall体现了模型对正样本的识别能力，Recall越高，模型对正样本的识别能力越强。F1 score是两者的综合，F1 score越高，说明模型越稳健。

## 4、结果处理

本实验对情绪的分类结果为

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 类别 | precision | recall | F1-sorce | support |
| others | 1 | 0.04 | 0.08 | 24 |
| happiness | 0 | 0 | 0 | 6 |
| sadness | 0 | 0 | 0 | 1 |
| surprise | 0 | 0 | 0 | 4 |
| disgust | 0.16 | 1 | 0.28 | 8 |
| repression | 0 | 0 | 0 | 7 |
| fear | 0 | 0 | 0 | 1 |
| accuracy |  |  | 0.18 | 51 |
| macro | 0.17 | 0.15 | 0.05 | 51 |
| Weighted avg | 0.5 | 0.18 | 0.08 | 51 |

可以看出效果极差，算法仍有待改进。目前也有小组成员在同步尝试复现基于图片进行表情识别的开源代码，后续可能还是得基于开源代码进行改进。