

# 项目说明文档

## 统计分析与建模课程项目

### ——选题二

小组成员：	学号	姓名
	1854062	许之博
	1951586	李朝义
	1952230	朱增乐
	1952395	沈韬

同济大学

Tongji University

目录

一、问题简介.....3

二、文献调研.....3

三、数据分析与预处理.....3

    1. 数据分析.....3

    2. 异常值处理.....3

    3. 特征筛选.....4

四、模型建立.....5

    1. 建模思路.....5

    2. 建模过程.....6

    3. 建模结果.....7

五、模型评估与应用.....7

    1. 模型质量.....7

    2. 模型质量.....9

六、总结与思考.....10

    1. 实验小结.....10

    2. 实验小结.....10

    3. 小组分工.....11

六、 参考文献.....11

## 一、问题简介

驾驶疲劳，是指驾驶人在长时间连续行车后，产生生理机能和心理机能的失调，而在客观上出现驾驶技能下降的现象。本文通过计算驾驶员的头部姿态，眼睛闭合频率和单位时间打哈欠的次数来判断驾驶员的疲劳程度，并基于时序模型和非时序模型针对驾驶员疲劳驾驶的面部特征构建三分类模型，对给定面部数据预测驾驶员是否疲劳驾驶，并给出模型预测的相关指标。

## 二、文献调研

通过对相关文献的调研，可以发现已有的文献之中对于司机疲劳驾驶的面部识别都是基于 dlib 人脸识别 68 关键点，文献针对面部特征的提取主要围绕以下三个方面：眼部、嘴部、头部，原因在于人困的时候眨眼、打哈欠和头部俯仰变化频率会提高，针对眼部和嘴部特征提取的方法比较类似，可以从面积或者纵横比进行特征提取，其中面积主要用于对眼部的特征提取，采取 PERCLOS 测量方法；对于头部俯仰角特征的提取一般步骤为：2D 人脸关键点识别、3D 人脸模型匹配、求解 3D 点和 2D 点的转换关系以及根据旋转矩阵求解欧拉角，之后可以通过姿态估计结果判断是否点头及幅度或者对鼻尖 30 号点的移动值的方差来判断司机是否处于疲劳驾驶。对于模型的建立，所阅读的文献都是采用深度学习的方法，比如 LSTM 网络。

以李小平的《基于深度学习的司机疲劳驾驶检验方法研究》为例，作者基于 PERCOLS 计算提取眼部特征，通过计算纵横比计算提取嘴部特征值，通过计算两眼 midpoint 到嘴部距离的变化作为头部俯仰的特征值，最后基于 LSTM 网络将一组司机面部特征作为输入，构建时序模型。关于俯仰角的计算，李小平文章中的算法并没有直接对俯仰角进行计算，而是用其他形式来体现俯仰角，为了更加准确的得到头部的摆动幅度，本小组阅读了其他的文献，在苏锦瑾的《基于面部特征的驾驶员疲劳驾驶监测系统》中采用引入 3D 模型计算欧拉角的方法，这也是计算俯仰角的一般方法，苏的文章中提出如果驾驶员在某个 10s 的时间段内 z 方向或 x 方向欧拉角的变化大于  $20^{\circ}$  的时间占比超过 30%，就可以认为驾驶员疲劳。

## 三、数据分析与预处理

### 1. 数据分析

针对给定的数据，本文从每组数据的缺帧数、最大帧数以及对应数量的分布情况入手，得到以下的结果：

- (1) 不考虑缺帧的情况，95%的数据的帧数大于 45 帧；
- (2) 缺帧方面不考虑最大帧之后缺值，有 14.2%的数据 (65 组) 存在缺帧的情况；
- (3) 在缺帧的数据中有 40 组数据缺帧数占比超过 10%，41 组数据缺帧数超过 5 帧，31 组缺帧数超过 10 帧

针对以上三个结果，本文分别进行如下处理：

- (1) 对于缺帧率大于 8%的数据和空数据采取直接删除
- (2) 对于保留的数据中缺帧的数据进行插值
- (3) 因为在插值之后数据的帧数几乎全部大于 50 帧，故以 50 帧为基准，不足 50 帧的数据忽略不计，超过 75 帧的数据采用拆分分为两组数据，提高数据的利用率

### 2. 异常值处理

在数据分析阶段，本文提出需要对空数据、缺帧数据以及补帧之后的数据进行处理，分别采用如下方式进行：

(1) 在读取数据时，对于空数据直接删除；

(2) 读入非空数据集后，通过现有帧数和最大帧数和比值可以判断每组数据的缺帧率，如果缺帧率=1-现有帧数/最大帧数，以分析阶段划分的 8%为基准，大于基准的数据进行删除；

(3) 对剩余缺帧率≤8%的非空数据进程插帧补足数据，本文所用的补帧方式为线性插值，采用该方式的理由是该方式在插值节点上的插值误差为零，且简单方便，适用于本文中的较小数据规模；

(4) 补帧之后需要对数据进一步处理，通过统计得到，补帧后的数据帧数≥50 帧，故在后续的数据模型构建时以此为基准，考虑到数据集中存在超过 70 帧的数据，如果对所有数据均采用直接截断的方式，会使得数据的使用率降低，影响模型结果，所以在本阶段对数据集进行截断拆分操作，数据量小于 75 (1.5 倍基准) 帧的直接截断，大于等于 75 帧的数据拆分为前 50 帧与后 50 帧两组，没有对所有数据进行拆分是为了减少一条数据的占比。

### 3. 特征筛选

(1) 眼部特征

(a) 纵横比 EAR: 驾驶过程中可以通过驾驶员的眼部动态进行捕捉判断驾驶员的疲劳程度，通过眼部定位点进行眼部的状态计算, 计算公式如下：

$$EAR = \frac{\|p_{37} - p_{41}\| + \|p_{38} - p_{40}\|}{2\|p_{36} - p_{39}\|}$$

为了保证准确度，使用六个眼部定位点，由该六个点进行计算得到 EAR，在对数据处理的过程中，我们发现左右眼的 ear 值差距过大，考虑是摄像头并未平视，而是具有一定的偏角，对其进行处理；先进行比较，若差值小于 0.01，则视为平视，取左右眼的平均值，否则取较大值，即离摄像头近的眼作为参照；计算公式可以计算出，csv 中每一帧的眼部纵横比用作 LSTM 的输入特征以及后续的计算输入

(b) Perclos: 通过计算闭眼阈值对驾驶员眼部状态进行判断：

$$EAR_{thresh} = \min(EAR_{List}) + (\max(EAR_{List}) - \min(EAR_{List})) * p$$

由于数据集已经分为三类，所以对于数据处理时，我们选择了对 level0 的闭眼阈值进行了平均值的计算，同时，考虑到疲劳因素，利用 level0 的平均闭眼阈值减去 level1 和 level2 闭眼平均阈值的差作为整个数据集的；其中 p 取 0.3，即眼睑遮住眼睛 70%的面积视为闭眼，对于整个 csv 的文件统计其闭眼频率，得到最终的疲劳程度 f 作为分类模型的输入；

(c) LongestCloseEye: 通过计算其闭眼时长将其作为一个特征输入到分类模型中进行建模；

(2) 嘴部特征

(a) 纵横比 MAR: 通过嘴部定位点计算嘴部纵横比，将其作为 LSTM 模型的输入，参与后续计算过程

$$MAR = \frac{\|p_{61} - p_{67}\| + \|p_{62} - p_{66}\| + \|p_{63} - p_{65}\|}{3\|p_{48} - p_{54}\|}$$

(b) 打哈欠次数 YAWNS: 在对嘴部特征提取时，由于疲劳时，打哈欠次数较多，即连续张嘴的现象较明显，所以提取特征哈欠次数这里选择讲超过三帧的连续张嘴记为一次打哈欠。

(3) 头部特征: 采用固定的世界坐标系，将 2D 平面中人脸的提取特征点，通过相机坐标系

与图像中心坐标系和像素中心坐标系将其匹配到 3D 模型中,对 pitch、roll、yaw 角度进行计算,从而计算后续的头部特征

(a) Pitch (垂直方向): 在一个时间段内 10 s 内,当  $Pitch \geq 20^\circ$  或者  $Roll \geq 20^\circ$  的时间比例超过 0.3 时,就认为驾驶员处于打瞌睡的状态预警;由于数据的来源存在偏角,所以我们这里考虑 pitch 角度的变化对瞌睡点头的特征进行提取,其作为后续计算使用同时作为 LSTM 模型的特征输入

(b) Roll (水平方向): 由于具有一定的偏角,所以不对其进行特征提取,其仅作为 LSTM 模型的参数进行输入;

(c) Yaw: 作为 LSTM 模型的参数输入

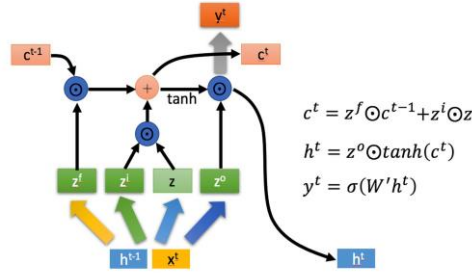
## 四、模型建立

### 1. 建模思路

考虑到本文中所使用的数据是时序的,并且已经被划分为三种类型,因此本文将选取时序模型(LSTM、GRU 等)进行模型构建三分类模型。

#### 1.1 时序建模

循环神经网络(Recurrent Neural Network, RNN)是一种用于处理序列数据的神经网络。相比一般的神经网络来说,它能够处理序列变化的数据。长短期记忆(Long short-term memory, LSTM)是一种特殊的 RNN,主要是为了解决长序列训练过程中的梯度消失和梯度爆炸问题。简单来说,就是相比普通的 RNN, LSTM 能够在更长的序列中有更好的表现。LSTM



LSTM 内部主要有三个阶段: 忘记阶段、选择记忆阶段以及输出阶段。

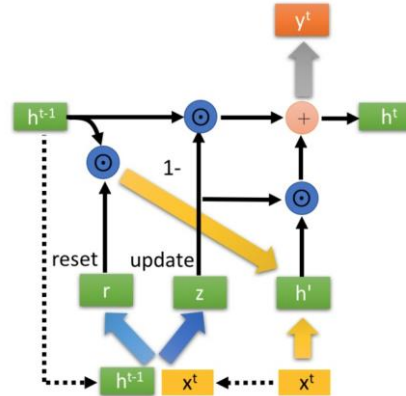
(a) 忘记阶段: 主要是对上一个节点传进来的输入进行**选择性**忘记。简单来说就是会“忘记不重要的, 记住重要的”, 具体来书, 具体来说是通过对计算得到的  $z^f$  (f 表示 forgot) 来作为忘记门控, 来控制上一个状态的  $c^{t-1}$  哪些需要遗忘的;

(b) 选择记忆阶段: 这个阶段将这个阶段的输入有选择性地“记忆”。主要是会对输入  $x^t$  进行选择记忆, 当前的输入内容由前面计算得到的  $z$  表示, 而选择的门控信号则是由  $z^i$  (i 代表 information) 来进行控制;

(c) 输出阶段: 这个阶段将决定哪些将会被当成当前状态的输出。主要是通过  $z^o$  来进行控制, 并且对上一阶段得到的  $c^o$  进行缩放(通过一个  $\tanh$  激活函数进行变化), 与普通 RNN 类似, 输出  $y^t$  往往最终也是通过  $h^t$  变化得到的。

本文中将通过特赠提取得到的特征向量划分训练集和测试集, 将训练集输入 LSTM 网络, 训练得到模型之后使用测试集进行模型检验。

但是，在实际 LSTM 建模中，我们发现由于数据集规模较小，模型过拟合比较严重。在最好的情况下会出现训练集准确率为 90%，但是测试准确率只有 60%左右的情况。而 GRU 模型的参数相对于 LSTM 更少，因此，我们尝试用 GRU 模型用来减少过拟合。



### 1.1.1 NNI 自动调参

NNI 是微软于 2021 年 8 月推出的一款用于支持机器学习实验的工具，它可以以自动化的方式辅助用户进行神经网络架构的设计、模型参数的调节等。NNI 具有方便易用、可扩展性强、灵活度高和调节效率高等优点。给定一个机器学习模型，可以通过以下四步来进行自动调参确定最优值：

- (a) 确定搜索空间，在 `search_space.json` 文件中写下参数名和取值范围
- (b) 更新代码，在原代码中加入少量几行代码，将参数传入模型
- (c) 编辑配置文件，`config.yaml`，确定自动调参一些参数，如是否使用 GPU，最大搜索轮次等
- (d) 执行自动搜索，打开 WebUI 即可实时查看参数搜索的过程。

在本文通过神经网络模型训练时为了更好的训练结果往往需要反复调参，为了减少调参的工作量，借助 NNI 实现自动调参。

## 1.2 非时序建模

数据来源已经分类为不疲劳，轻度疲劳和重度疲劳三种状态，每一个 csv 文件都代表一个视频，每一个视频都对应着三种状态中的一种，因此也可以通过从视频 csv 文件中抽取特征，再利用已经确定好的标签，建立非时序分类模型进行训练。常见的非时序分类模型有以下几种：

### (1) 决策树模型

决策树是一种非参数的监督学习方法，它主要用于分类和回归问题。决策树模型通过一系列 if then 决策规则的集合，将特征空间划分成有限个不相交的子区域，对于落在相同子区域的样本，决策树模型给出相同的预测值。

### (2) 朴素贝叶斯模型

朴素贝叶斯的“朴素”，是指样本的特征之间相互独立。朴素贝叶斯是找出特征输出  $Y$  和特征向量  $X$  之间的联合分布  $P(X, Y)$ ，然后用  $P(Y|X) = P(X, Y) / P(X)$  得出特征输出  $Y$  和特征向量  $X$  之间的关系。

### (3) 随机森林模型

建立多个决策树并将他们融合起来得到一个更加准确和稳定的模型，是 bagging 思想和随机选择特征的结合。随机森林构造了多个决策树，当需要对某个样本进行预测时，统计森林中的每棵树对该样本的预测结果，然后通过投票法从这些预测结果中选出最后的结果。

## 2. 建模过程

### 2.1 预处理及特征提取

预处理阶段包括处理异常值、补足缺失帧和截断拆分，预处理结束之后，按照数据分析阶段对特征值的提取方法，提取特征向量，每组数据提取出的特征向量将会被传递到后续的模型中构建，

## 2.2 时序模型构建

- (1) 按 7:3 将提取的特征划分训练集与测试集；
- (2) 将训练集作为模型的输入进行模型训练；
- (3) 在测试集上测试模型的结果，并调整参数找到最优模型
- (4) 比较 LSTM 和 GRU 两种模型的建模结果。

## 2.3 非时序建模

- (1) 按照 4:1 将提取的特征划分训练集与测试集
- (2) 将训练集作为模型的输入进行传统机器学习模型训练
- (3) 在测试集上测试模型的结果

## 3. 建模结果

### 3.1 训练结果

在我们的模型训练中，发现我们的模型是有些过拟合的。在传统机器学习模型中，训练集准确率达到到了 90%，而测试集准确率则只有 50%。而深度学习模型在训练集的准确率都可以达到 95%以上，但测试集结果的准确率却也是只有 60%左右。

### 3.2 模型结果比较

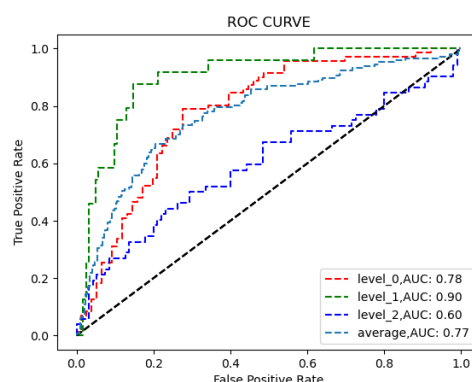
时序模型略强于非时序模型，具体模型质量见本文档后部分。而 GRU 和 LSTM 进行对比后发现，二者从准确率、F1 值等实际上差别并不大。

## 五、模型评估与应用

### 1. 模型质量

#### 1.1 LSTM 模型

##### 1.1.1 ROC 曲线



##### 1.1.2 混淆矩阵

	Level_0	Level_1	Level_2
level0	52	3	16
level1	1	17	6
level2	20	10	22

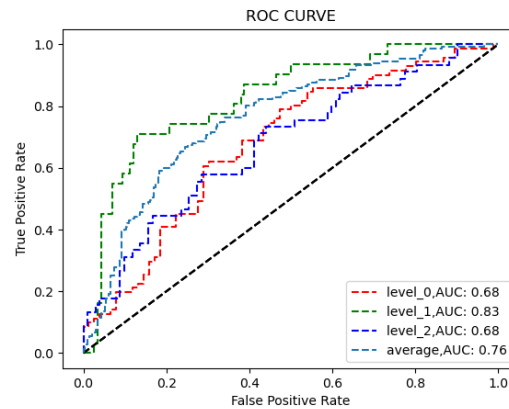
##### 1.1.3 F1 值

	Recall	Precision	F1
Level_0	0.71	0.73	0.72

Level_1	0.57	0.71	0.63
Level_2	0.50	0.42	0.46

## 1.2 GRU 模型

### 1.2.1 ROC 曲线



### 1.2.2 混淆矩阵

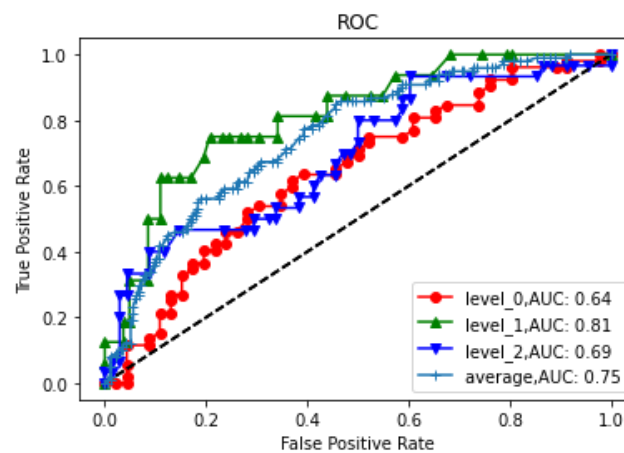
	Level_0	Level_1	Level_2
Level_0	57	5	9
Level_1	11	17	3
Level_2	26	4	15

### 1.2.3 F1 值

	Recall	Precision	F1
level0	0.80	0.61	0.69
level1	0.55	0.65	0.59
level2	0.33	0.55	0.41

## 1.3 非时序模型 (以随机森林为例)

### 1.3.1 ROC 曲线



### 1.3.2 混淆矩阵

	Level_0	Level_1	Level_2
level0	33	7	12



level1	5	8	3
level2	13	3	14

### 1.3.3 f1 值

	Recall	Precision	F1
level0	0.63	0.65	0.64
level1	0.50	0.44	0.47
level2	0.47	0.48	0.47

## 2. 模型质量

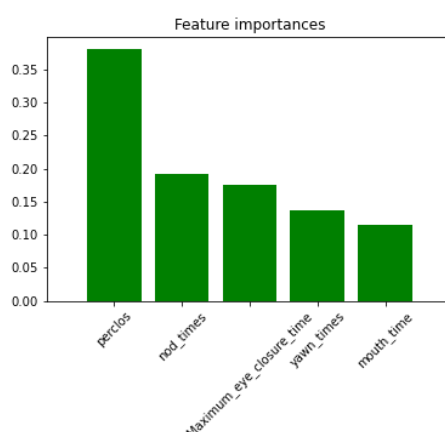
### 2.1 传统机器学习(朴素贝叶斯为例)

从 ROC 曲线图和混淆矩阵可以看出, 模型对 level\_0, 即完全不疲劳数据的分类效果较好, 这可能是因为 level\_0 数据占比较大。level\_1 的特征较为明显, 而 level\_0 和 level\_2 的特征不显著。

下图为非时序模型在训练集上的测试结果。与前文中在测试集上得出的结果对比可以发现模型出现了过拟合现象, 这可能是因为训练集的数量级和模型的复杂度不匹配, 也有可能是噪音数据的干扰。

	precision	recall	f1-score	support
0.0	0.89	0.95	0.92	183
1.0	0.97	0.92	0.95	78
2.0	0.93	0.88	0.90	128

我们模型中五个特征的特征贡献度如下图所示, 从图中可以看出, 从眼部提取出来的 perclos 特征贡献度较大, 从嘴部和头部提取出的相关特征贡献度较小。经分析, 由于人在正常活动情况下眼部一般会保持打开状态, 尤其是在驾驶过程中需要保持精神集中, 眼部会很少出现闭合。而从头部和嘴部提取特征考虑到的是打盹时的点头与张嘴情况, 而驾驶员在正常活跃情况下也有可能会有频繁点头或张嘴情况, 如唱歌、交谈。而且由于摄像头角度原因, 测量得到的 68 点点位和头部偏向角会出现偏移, 在特征提取时还需要另加考虑。



在使用单独的机器学习模型后, 本实验也考虑使用集成学习模型, 将多种机器学习模型结合起来, 分配以不同的权重, 经测试后发现, 集成学习得出的结果更加稳定。本实验中使用的是逻辑回归模型、随机森林模型和朴素贝叶斯模型, 分配以 2: 1: 2 的权重。集成学习就是组合这里的多个弱监督模型以期得到一个更好的模型强监督模型, 集成学习的潜在思想是即便某个分类器得到了错误预测, 其他模型也能将错误纠正回来。

	precision	recall	f1-score	support
0.0	0.58	0.87	0.69	52
1.0	0.47	0.50	0.48	16
2.0	0.33	0.03	0.06	30

## 2.2 时序模型建模

我们所采用的深度学习模型是 LSTM 和 GRU。LSTM 的网络架构为输入特征个数为 5，隐藏单元个数为 50，隐藏层个数为 5。在 LSTM 后我们加入了一个全连接层用于分类。GRU 的结构同样如此。而在损失函数上，我们选择的是交叉熵函数，优化器选用的是 Adam 优化器。

时序模型在训练集上的准确率可达 95%，而在测试集上的表现却非常一般，准确率只有 60% 左右，显然这是发生了过拟合，本文认为这是由于数据量过少导致，为了寻求更好的模型，本文在模型建立时尝试不同的训练次数，随着训练次数的增加，在测试集上的表现刚开始逐渐增强，而后续开始降低稳定在 63% 附近。

# 六、总结与思考

## 1. 实验小结

针对本文中的数据，时序模型与非时序模型得到的模型结果都比较一般，相比之下时序模型的结果更加好一点，这说明针对时序数据，时序模型往往能取得更好的预测结果。

本文中的模型结果并不如意，本文认为一方面是数据量的大小不足以构建一个准确的模型，另一方面是本文中特征值提取阶段以及模型训练阶段存在不足。

同时，在特征工程中进行特征的提取后，进行特征的画图统计操作，可以看出三类数据的特征提取其分类特征并不明显，甚至在一些参数比较极端的情况下，可以成为二分类模型（level0 和 level2）（level1）；由此考虑数据来源于已经分类好的视频然后经过代码对面部 68 点定位提取后，然后再经过处理后的原因，所以分类数据中的特征并不明显，同时，由于数据量较少，模型存在过拟合的情况，训练集较少，所以模型效果并不尽如人意，可视化文件进行数据的查看时，难以判断其疲劳程度。

## 2. 实验小结

### 2.1 数据预处理

在数据预处理中，我们对缺帧的数据进行了补帧处理。此外，我们采用数据增强的思想。将过长的数据分为两段，对有限的数据进行了一定扩充。

### 2.2 特征工程

在特征工程中，对于特征的提取，由于角度存在一定的偏差，所以我们对于眼部和嘴部的 EAR 值和 MAR 值进行了一定的处理使之符合我们的模型检验；

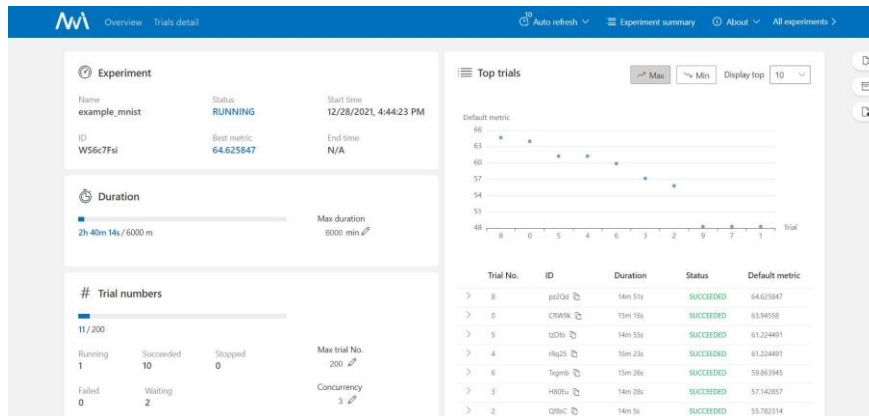
同时，在原有的特征上，新添加了特征，例如：眼部闭眼的最长时长，嘴部张开的时长，以及瞌睡点头的特征进行处理。

### 2.3 模型构建

在模型构建中，我们采用了两种思路，传统机器学习和深度学习。在传统机器学习中，我们尝试了集成学习，尽可能融合多种分类器。而在深度学习中，我们对比了 LSTM 和 GRU 两种不同的 RNN 网络。

### 2.4 模型训练

在深度学习训练中，我们尝试了 nni 自动调参工具。但是碍于算力的原因，我们并没有能够选择最好的一组参数进行训练。



### 3. 小组分工

许之博：25%，非时序建模

李朝义：25%，文献调研，文档撰写，PPT

朱增乐：25%，时序建模

沈韬：25%，特征提取

## 六、参考文献

- [1] 柳龙飞,伍世虔,徐望明.基于人脸特征点分析的疲劳驾驶实时检测方法[J].电视技术,2018,42(12):27-30+55.
- [2] 苏锦瑾,霍春宝,王特特.基于面部特征的驾驶员疲劳检测系统[J].信息技术与信息化,2021,(09):114-116.
- [3] R. Jimenez, F. Prieto and V. H. Grisales, "Detection of the Tiredness Level of Drivers Using Machine Vision Techniques," 2011 IEEE Electronics, Robotics and Automotive Mechanics Conference, 2011, pp. 97-102, doi: 10.1109/CERMA.2011.23.
- [4] J. -J. Yan, H. -H. Kuo, Y. -F. Lin and T. -L. Liao, "Real-Time Driver Drowsiness Detection System Based on PERCLOS and Grayscale Image Processing," 2016 International Symposium on Computer, Consumer and Control (IS3C), 2016, pp. 243-246, doi: 10.1109/IS3C.2016.72.