

# ERP 无训练数据集

## 一、 实验范式

任务概述：本次 ERP 赛道采用快速序列视觉呈现（RSVP）范式。在本次实验中共有三类图像，包含两类目标（车，人）和一类背景（无车无人的街景），所有图像来自街景拍摄，并使用 RSVP 范式将图像序列呈现给被试，如图 1 所示。参赛者需要通过分析被试脑电信号，找到目标图像在序列中所在的位置并判断其类别（车或人）。

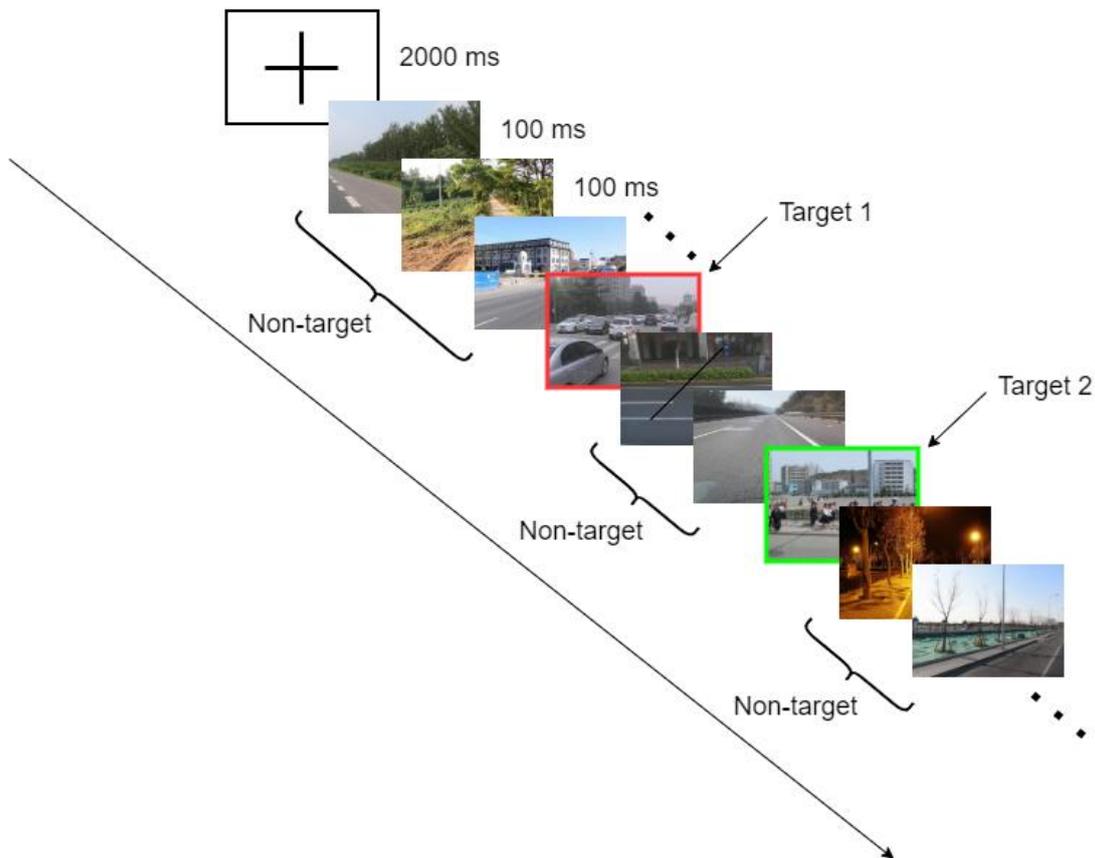


图 1 RSVP 范式

实验流程：每个被试以 block 为单位采集数据，在本次实验中每个被试均采集多次 block，每次 block 开始前，屏幕的中央会有提示。图片序列的呈现以 trial 为单位，每次 trial 开始时屏幕上会有十字提示被试关注屏幕中央，每次 trial 包含 50 张图片，其中目标种类和数量均不固定（最多 5 张目标图像），每幅图像都呈现在屏幕的中央，呈现速度为每秒 10 张，每个 block 包含 10 次 trial。整个实验流程如图 2 所示。

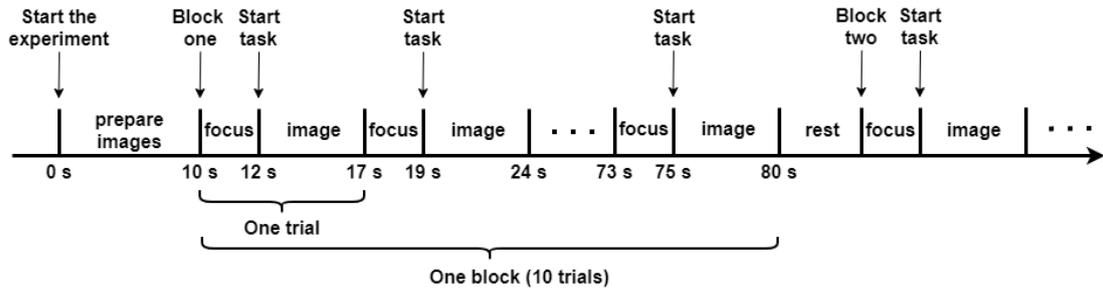


图 2 实验过程

实验数据使用博睿康 64 通道脑电采集设备采集,第 65 导联为 trigger 信息,原始采样率为 1000Hz,未做其他处理。

训练数据的具体 trigger 定义如表 1 所示。在线上测试时,赛题数据的三类图像 trigger 全部为 1,仅用作同步,不包含类别信息。

表 1 Trigger 定义

定义	Trial 开始	Trial 结束	Block 开始	Block 结束	系统预留
Trigger 号	240	241	242	243	244-255
定义	非目标图像	第一类目标 (人)	第二类目标 (车)		
Trigger 号	1	2	3		

表 2 导联序号-导联名称

导联序号	1	2	3	4	5	6	7	8
导联名称	Fpz	Fp1	Fp2	AF3	AF4	AF7	AF8	FZ
导联序号	9	10	11	12	13	14	15	16
导联名称	F1	F2	F3	F4	F5	F6	F7	F8
导联序号	17	18	19	20	21	22	23	24
导联名称	FCz	FC1	FC2	FC3	FC4	FC5	FC6	FT7
导联序号	25	26	27	28	29	30	31	32
导联名称	FT8	Cz	C1	C2	C3	C4	C5	C6
导联序号	33	34	35	36	37	38	39	40
导联名称	T7	T8	CP1	CP2	CP3	CP4	CP5	CP6
导联序号	41	42	43	44	45	46	47	48

号								
导联名称	TP7	TP8	Pz	P3	P4	P5	P6	P7
导联序号	49	50	51	52	53	54	55	56
导联名称	P8	POz	PO3	PO4	PO5	PO6	PO7	PO8
导联序号	57	58	59	60	61	62	63	64
导联名称	Oz	O1	O2	ECG	HEOR	HEOL	VEOU	VEOL

数据流采用模拟在线方式提供。每调用一次数据读取方法，可获得一个新数据包，数据包中包含一个 block 的实验 EEG 数据（10 次 trail 共 500 张图片），以及在该数据包记录过程中收到的 trigger 信息。若测试数据中包含多组 block 数据，则一组 block 数据发送完毕后，数据读取方法被再次调用时，将会开始下一组 block 数据的 EEG。而当所有实验数据发送完毕后，程序终止标记 finishedFlag 将被置为 1。参赛算法检测到 finishedFlag 为 1 后，需要自行结束 run() 方法执行。

## 二、 算法规范

参赛算法调用数据读取方法获取脑电数据。数据读取方法被调用一次，比赛系统会返回一个新数据包，参赛算法可以对新数据包进行缓存并处理。当算法认为接收到的数据足以满足判决条件时，需要调用反馈方法向比赛系统报告识别结果。比赛系统根据数据读取方法的调用次数计算出算法使用有效数据长度，并结合反馈正确率，综合计算出模拟信息传输速率。

参赛算法需要同时满足以下几个约束条件：

### 1、 试次起止约束：

在对单一试次数据的检测识别过程中，参赛算法需要在接收到该试次 trigger 之后开始检测，并且最迟在接收到下一 trigger 时进行反馈报告。否则，报告结果将被错判为后续试次的识别结果。

### 2、 返回长度约束：

本项目每次读取数据均为一个 block，其中共 500 有个图片呈现时的 trigger，trigger 值为 1，参赛者可根据各自算法需要，自行分割数据。处理结束后，返回长度为 500 的分类结果标签（非目标图像标签为 0，人为 1，车为 2）。

### 3、 算法终止约束：

当接收到数据包中 Endflag = 1 时，意味着所有实验数据均已发送完毕，参

赛算法需要停止处理并自行退出。

### 三、 赛题框架

#### 1. 参赛者用例

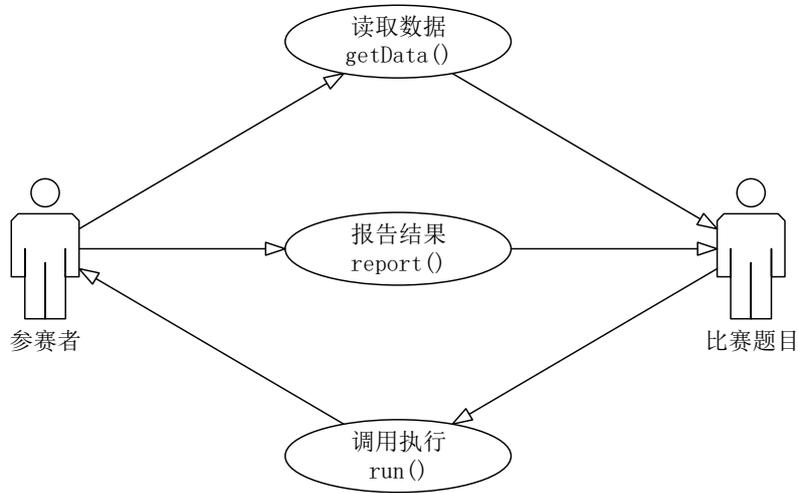


图 2 参赛者用例

#### 2. 系统主体框架如图 3 所示。

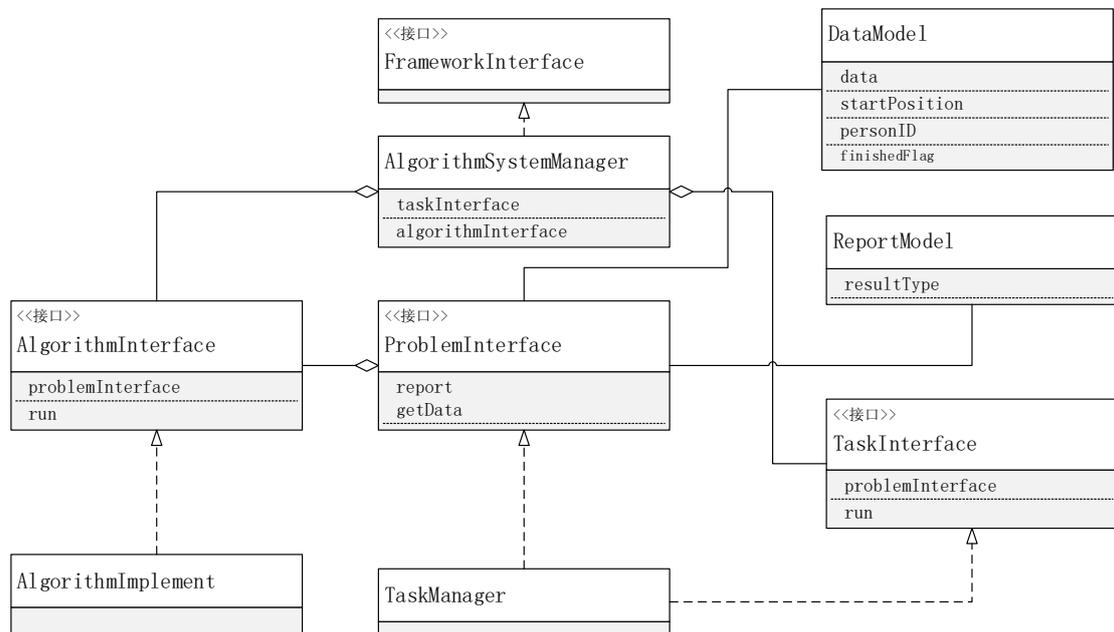


图 3 系统主体框架

##### (1) FrameworkInterface 框架接口

该接口主要负责赛题程序与外部执行系统的连接。该框架接口的实现类为 AlgorithmSystemManager，实现接口定义的所有函数。

##### (2) ProblemInterface 题目接口

该接口是**面向参赛者**的赛题接口，主要负责题目与参赛算法之间的数据传递及结果报告。参赛者可以通过该接口获取比赛数据，并通过该接口报告识别结果。比赛题目需要根据参赛算法获取比赛数据的次数，以及报告结果的正确率综合给出比赛评分。

### (3) TaskInterface 赛题接口

该接口主要负责实现赛题的数据填充、获取得分、清除数据以及清除报告结果，框架通过该接口实现对赛题的调用。

### (4) AlgorithmInterface 算法接口

通过该接口比赛题目可以对参赛算法进行验证计算。**参赛者需要实现该接口**。在执行过程中，参赛算法需要通过 **ProblemInterface** 接口获取数据，并且通过该接口报告结果。同时，参赛者需要控制算法的计算复杂度，否则当运行时间超过预定长度时，系统将自动终止该计算进程，所获成绩无效。

## 3. 数据模型

### (1) DataModel 参赛者数据模型

1) **data**: float 类型矩阵，分段数据。例如包含有 64 导 EEG 数据+1 导 trigger 信号。

2) **startPosition**: int 类型标量，当前分段数据起始时刻相对于该 block 数据起始时刻的索引位置。

3) **personID**: int 类型标量，当前数据来源受试者序号。

4) **finishedFlag**: bool 类型标量，测试结束标志。当参赛算法通过 **ProblemInterface.getData()**函数获取数据包中该字段为 **True** 时，需要自行退出程序运行。

## 4. 参赛者相关接口函数

### (1) ProblemInterface

该接口由出题方负责实现，包括数据获取方法及结果反馈方法。在算法运行前，该接口的实现类会被注入参赛算法实现类中。算法执行过程中，可以调用该接口获取数据，并通过结果反馈方法报告识别结果。出题方根据数据获取方法的被调用次数，及结果反馈的正确性进行综合评分。

1) **def getData(self):**

输入参数：无

输出参数：DataModel

实现功能：获取下一分段实验数据。

2) def report(self, reportModel):

输入参数：ReportModel

输出参数：无

实现功能：反馈识别结果。

(2) AlgorithmInterface

参赛者需要将程序运行过程填入 run 函数中。在算法执行过程中，通过 ProblemInterface 接口 getData 方法获取 DataModel 类型数据，并通过 report 方法返回 ReportModel 类型结果。当通过 getData 获取的 DataModel 数据中 finishedFlag 为 true 时，意味着数据处理完毕，该函数需要自行退出运行。

1) def run(self):

输入参数：无

输出参数：无

实现功能：算法分析过程。

提交格式

本赛题程序使用 python 语言编写，需提交基于 python 3.8 版本的扩展名为.pyc 的文件。

## 5. 提交样例

参考配套代码。

参赛者可通过修改 Algorithm 文件夹中的代码完成算法，为了避免未知错误，请勿在主目录内添加文件夹。完成后重新打包程序（包含 AlgorithmImplement 文件夹和 config.toml）--> 分组 --> 具体分组 --> 计算单元 --> 定义计算单元 --> 上传程序包 --> 提交到比赛 --> 选择比赛 --> 部署 -->完成比赛。

部署完成后在具体到比赛中的排行榜中查看 比赛成绩；

需要注意的是，为防止参赛者修改代码框架作弊，保护评分程序会完全覆盖参赛者的代码（除了 AlgorithmImplement 目录和 config.toml）在提交到比赛 -> 部署时，启动的实际为 评分程序 + 参赛者的 AlgorithmImplement 目录，其余运行配套代码均为服务器内置程序（包括 main.py 等文件，服务器内置评分程序与范例中程序框架基本相同，但包含评分功能和读取服务器比赛数据功能）。

## 6. 评分方式

本系统以 UAR 作为评分标准:

$$UAR = \sum_{r=1}^n w_r \frac{c_r}{t_r}$$

其中  $n$  表示类别总数, 在本题目中  $n = 3$ ,  $w_r$  表示为每个类别施加的权重因子, 目前被设置为  $[0.33, 0.33, 0.33]$ ,  $t_r$  表示每类图像真正的数量,  $c_r$  表示每类图像预测正确的数量, 特别注意, 系统以每次 trail 为单位计算 UAR, 最终平均所有 block 进行评分。

## 7. 性能评估方法

参赛算法通过数据读取方法获取新数据包。当所得数据包内含有 trigger 信号时, 评分系统将自动开始记录算法识别过程中所使用 EEG 信号的长度, 直至反馈方法被调用。从 trigger 开始到反馈方法被调用时所获取的 EEG 数据长度将作为该试次的模拟试次时长。

需要特别指出, 在本比赛项目中每一个包含 trigger 的数据包, 其依然被视为是前一试次的数据。而新试次数据是从包含 trigger 数据包的下一个数据包开始计算。因此参赛算法不可在获取到包含 trigger 信号的数据包时立刻反馈, 而最早需获取到下一数据包后才可反馈。

## 8. 结果反馈异常处理

### 1) 重复多次报告

在一个试次时间内, 参赛算法多次反馈结果, 则仅按照第一次反馈的时间及结果进行记录。

### 2) 结果未反馈

若在一个试次时间内, 参赛算法未反馈结果, 同时判决结果将被记录为误判。

### 3) 结果反馈超时

结果反馈时, 同时该试次时长将按照从 trigger 开始时刻到结果反馈时刻的数据长度进行计算。

### 4) 算法执行超时

为满足脑-机接口系统实时处理需求, 本项目同时对参赛算法的计算复杂度有一定要求。本比赛项目将会根据比赛数据量大小确定一个计算时间。若算法复

杂度过高导致系统运行超时，则该算法比赛成绩将被视为无效。

#### 5) 分类结果异常

在一次试次时间内，所有返回的类别标签中，标签数量不等于 50，成绩将被视为无效。