**运动视频理解AI能力定制项目**

**详细设计方案**

咪咕互动娱乐有限公司

2017.10

**修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **修订日期** | **修订内容** | **修订人** | **核准人** |
| V1.0 | 2017/9/21 | 创建文档 | 侯逸文 |  |
| V1.1 | 2017/10/14 | 编写文档内容 | 侯逸文  杨建党  刘亮 |  |
| V1.2 | 2017/10/18 | 更新直播视频解析和接口 | 侯逸文 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目录**

[1 概述 5](#_Toc496097614)

[1.1 建设背景 5](#_Toc496097615)

[1.2 参考文档 5](#_Toc496097616)

[1.3 建设目标 5](#_Toc496097617)

[2 总体设计 7](#_Toc496097618)

[2.1 总体架构 7](#_Toc496097619)

[2.2 功能架构 8](#_Toc496097620)

[2.3 系统流程 10](#_Toc496097621)

[2.4 数据拓扑 12](#_Toc496097622)

[3 功能设计 13](#_Toc496097623)

[3.1 视频模型训练 13](#_Toc496097624)

[3.1.1 视频模型管理 13](#_Toc496097625)

[3.1.2 动作指标同步 20](#_Toc496097626)

[3.1.3 训练视频采集 22](#_Toc496097627)

[3.1.4 样本数据标注 24](#_Toc496097628)

[3.1.5 数据上传云平台 26](#_Toc496097629)

[3.1.6 视频模型训练 28](#_Toc496097630)

[3.1.7 训练结果发布 34](#_Toc496097631)

[3.2 视频模型使用 35](#_Toc496097632)

[3.2.1 视频直播及解析 35](#_Toc496097633)

[3.2.2 模型在线使用 37](#_Toc496097634)

[3.2.3 模型离线使用 39](#_Toc496097635)

[3.3 跑步路线推荐 41](#_Toc496097636)

[3.3.1 数据表 41](#_Toc496097637)

[3.3.2 运动轨迹接入 41](#_Toc496097638)

[3.3.3 热点区域划分 43](#_Toc496097639)

[3.3.4 推荐路线生成 45](#_Toc496097640)

[3.3.5 推荐路线查询 48](#_Toc496097641)

[3.4 咪咕善跑配套 49](#_Toc496097642)

[3.4.1 动作指标管理 49](#_Toc496097643)

[3.4.2 训练开始准备 51](#_Toc496097644)

[3.4.3 训练过程提示 53](#_Toc496097645)

[3.4.4 训练结果回放 54](#_Toc496097646)

[3.4.5 训练结果分享 57](#_Toc496097647)

[3.4.6 跑步路线推荐 58](#_Toc496097648)

[3.5 系统接口 59](#_Toc496097649)

[3.5.1 数据上传接口 59](#_Toc496097650)

[3.5.2 模型发布接口 60](#_Toc496097651)

[3.5.3 模型在线使用接口 60](#_Toc496097652)

[3.5.4 模型离线使用接口 60](#_Toc496097653)

[3.5.5 截图开始通知接口 61](#_Toc496097654)

[3.5.6 截图结束通知接口 61](#_Toc496097655)

[3.5.7 直播状态通知接口 61](#_Toc496097656)

[3.5.8 录像完成通知接口 61](#_Toc496097657)

[3.5.9 关闭直播接口 61](#_Toc496097658)

[3.5.10 删除截图接口 62](#_Toc496097659)

[4 系统响应码 63](#_Toc496097660)

# 概述

## 建设背景

运动视频AI目前已经进入大众消费级应用市场中，在游戏、安全、摄影等领域有着诸多应用，但在互联网健身行业中尚未有成功运用的案例。咪咕善跑希望通过接入运动视频AI理解能力实现用户在使用运动训练功能时能够获得即时反馈，在训练完成后能够获得训练效果评估。

运动视频理解AI能力定制项目（以下简称“项目”）应对当前的AI发展趋势，配合咪咕文化建设AI统一云平台，通过构建统一的、跨媒介的、多形态交互能力的AI云服务，为咪咕善跑提供统一的AI服务与能力。通过AI服务于咪咕善跑，形成差异化能力，提升客户体验，打造杀手级应用；通过AI提升业务全流程效率，节省研发/运营成本；通过AI整合并挖掘内部数据和能力，实现公司数据价值最大化；通过AI构建新的业务或商业模式。

## 参考文档

* 《灵犀云三期扩容工程项目（运动视频理解AI能力定制）技术服务规范书》
* 《灵犀云三期扩容工程项目（AI统一云平台与通用能力）技术服务规范书》
* 《咪咕公司AI能力数据规范》
* 《咪咕公司AI数据标注规范》
* 《运动视频理解AI能力定制项目需求分析方案》

## 建设目标

本项目的建设目标是使得咪咕善跑在互利网健身行业应用软件中具有短期内无法被竞品超越的特色功能，为用户带来前所未有的线上健身训练体验。主要实现以下四个场景：

1、运动训练指导

用户在使用善跑对照训练视频锻炼时，通过摄像头采集用户运动过程中的体态影像，客户端调用AI能力，实时跟踪用户头部、躯干和四肢运动，计算用户肢体角度、速度、位移等指标，比对标准动作指标值，得出用户动作与标准动作之间的差距，通过语音播报或图文说明等方式，指导用户纠正错误动作。

2、跑步路线推荐

用户在开始跑步前，能在善跑客户端中选择系统推荐给他的附件热门跑步路线。善跑调用AI能力，根据用户所处位置、当前时间、信息标签（里程、人数、耗时等）等条件，向用户推荐适合最佳跑步路线。

3、赛事在线报名

用户在善跑客户端中上传自己的身份证照片并自拍大头照，报名参加马拉松赛事，善跑调用人脸识别AI能力，识别身份证上的照片与用户自拍是否为同一个人，判读通过则允许报名直接分配给用户参赛胸牌号码。

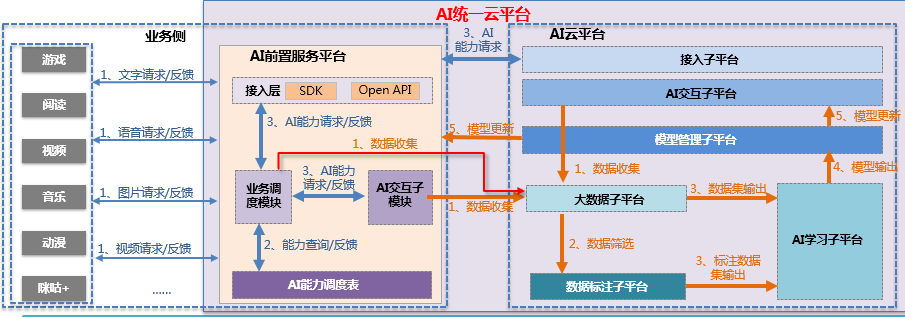
4、照片智能分拣

用户在参加完马拉松赛事后，善跑调用人脸识别AI能力，从赛事照片中识别各个参赛用户，包括用户的胸牌号码和人脸。用户在善跑客户端中可通过赛事的胸牌号查询自己的参赛照片。

# 总体设计

## 总体架构

咪咕AI云平台的总体架构如图1所示，整个平台由咪咕游戏、AI前置服务平台、AI云平台三大部分组成。



**图1 AI云平台架构示意**

AI云平台中各个部件的主要功能如下：

1. **咪咕善跑，产品侧-南京互娱**

* 负责运动训练样本视频采集、格式转换，向AI云平台上传样本视频
* 负责训练动作指标定义
* 负责咪咕善跑直播视频采集、格式转换
* 负责咪咕善跑前端功能呈现

1. **AI前置服务平台，能力侧-南京互娱**

* 负责从AI云平台下载模型文件，并载入机器学习框架运行
* 负责提供视频理解的AI能力，实现运动跟踪、动作评价、路线推荐
* 负责接入AI云平台提供的其他AI能力服务
* 负责各个AI能力服务的服务注册、服务编排、流程监控

1. **AI云平台，平台侧-成都大数据**

* 负责接收、清洗、存储训练视频
* 负责对训练视频进行数据标注，形成训练样本
* 负责载入训练样本，对视频模型进行训练，输出模型文件
* 负责对各类模型进行统一管理，包括注册、发布、评测、版本管理等
* 负责对外提供人脸识别等通用AI能力

## 功能架构

根据系统的总体功能和系统架构，将各部分功能细化，形成整体功能架构，如图2所示。



**图2 系统功能架构示意图**

整个系统由善跑客户端、善跑服务端、互娱大数据、视讯服务端、AI前置平台、AI云平台共六个部件构成，各部件的细化功能如下：

1. **善跑客户端**

* 与善跑服务端交互数据，向用户提供训练开始准备、训练过程提示、训练视频回放、训练结果分享、跑步路线推荐功能。
* 在用户训练过程中使用视讯直播SDK，调起移动终端摄像头拍摄用户视频并将视频流推送至视讯服务端。
* 在用户训练过程中定时向善跑服务端请求以获取用户动作评价，在训练结束后回放用户训练录制视频、训练处理视频。

1. **视讯服务端**

* 接收善跑客户端推送的训练视频流，在接收过程中将视频流按时间顺序逐帧转成图片保存在咪咕云上。
* 在视频接收结束后将视频流转成完整的视频保存在咪咕云上，向互娱大数据平台推送咪咕云上的视频地址。

1. **善跑服务端**

* 与善跑客户端端交互数据，向用户提供训练指导、路线推荐的相关功能。
* 向运营人员提供指标管理界面，定义各个动作的关键姿态、运动指标，并将动作、姿态、指标同步给互娱大数据平台。
* 在收到善跑客户端的运动评价请求后，调用互娱大数据平台的运动视频模型使用接口，在训练过程中获取用户训练动作评价，在训练结束后获取处理视频地址、分享截图地址。

1. **互娱大数据**

* 异步多线程方式从视讯咪咕云上下载视频截图、录制视频。
* 在接收到善跑服务端的模型使用请求后，调用AI前置平台的模型服务接口，在训练过程中获取用户训练动作评价，在训练结束后获取处理视频地址、分享图片地址。
* 2017年向算法人员提供模型管理功能，包括接收并查询善跑服务端同步来的指标数据、查询模型日志、查询和载入模型版本等功能。
* 2017年向标注人员提供数据标注功能，可完成源视频的样本标注。
* 按照AI数据规范要求将源视频、标注结果导出成规定格式，生成数据文件、字段文件、校验文件，并传输到FTP前置机上。
* 接收善跑服务端传来的跑步轨迹数据，每日定时进行跑步路线推荐相关的区域划分、路线合并计算。
* 调用AI云平台的人脸识别服务，支撑善跑的赛事报名、照片分拣等功能

1. **AI前置平台**

* 使用模型部署工具在Tensflow框架中部署最新版的运动视频理解模型文件。
* 对外提供HTTP形式的运动视频理解模型服务。

1. **AI云平台**

* 存储模型样本等数据。
* 向算法人员提供Tensflow框架进行运动视频理解模型训练，完成训练的模型可导出成模型文件下载到AI前置平台部署。
* 提供人脸识别的AI服务。
* 2018年向算法人员提供模型管理功能，包括接收并查询善跑服务端同步来的各类数据、查询模型日志、查询载入模型版本等功能。
* 2018年向标注人员提供数据标注功能，可完成源视频的样本标注。

## 系统流程

1. **视频处理流程**



1. **AI模型训练流程**



1. **AI模型使用流程**



## 数据拓扑



# 功能设计

## 视频模型训练

### 视频模型管理

#### 数据表

动作模型表（action\_model）、动作关键姿态表（action\_pose）、动作指标表（action\_index）、姿态与指标对应表（action\_pose\_index）、模型版本表（action\_model\_version）、模型日志表（action\_model\_log）

|  |  |  |  |
| --- | --- | --- | --- |
| action\_model动作模型表 | | | |
| 字段 | 类型 | 是否必填 | 说明 |
| **id** | Number(11) | Y | 动作id，自增id |
| action\_code | varchar2(100) | Y | 动作编码 |
| action\_name | varchar2(100) | N | 动作名称。例如：T字伸展。 |
| summary | varchar2(2048) | Y | 内容详情，含训练动作说明、注意事项、训练部位、训练器械等。例如：屈髋屈膝，俯身使身体与地面成60°左右夹角，手臂伸直自然下垂 |
| create\_time | date | N | 创建时间 |
| update\_time | date | N | 最新一次修改时间 |
| status | Varchar2(128) | N | 状态 |

|  |  |  |  |
| --- | --- | --- | --- |
| action\_index 动作指标表 | | | |
| 字段 | 类型 | 是否必填 | 说明 |
| **id** | Number(11) | Y | 动作指标id |
| action\_index\_name | Varchar2(100) | Y | 动作指标名称 |
| action\_index\_type | Varchar2(100) | Y | 动作指标类型：1-角度；2-距离；3-位移 |
| action\_index\_nodes | Varchar2(100) | Y | 指标涉及关节点编号列表 |
| action\_index\_summary | Varchar2(2048) | Y | 指标定义说明 |
| action\_index\_function | Varchar2(2048) | N | 指标计算方法 |
| create\_time | date | N | 创建时间 |
| update\_time | date | N | 最新一次修改时间 |
| status | Varchar2(128) | N | 状态 |

|  |  |  |  |
| --- | --- | --- | --- |
| action\_pose 动作关键姿态表 | | | |
| 字段 | 类型 | 是否必填 | 说明 |
| **id** | Number(11) | Y | 动作关键姿态id |
| action\_id | Number(11) | Y | 动作id |
| action\_pose\_code | Varchar2(100) | Y | 动作关键姿态编码 |
| action\_pose\_name | Varchar2(100) | Y | 动作关键姿态名称 |
| action\_ pose\_summary | Varchar2(2048) | N | 动作关键姿态说明 |
| create\_time | date | N | 创建时间 |
| update\_time | date | N | 最新一次修改时间 |
| status | Varchar2(128) | N | 状态 |

|  |  |  |  |
| --- | --- | --- | --- |
| action\_pose\_index动作姿态与指标对应表 | | | |
| 字段 | 类型 | 是否必填 | 说明 |
| **id** | Number(11) | Y | 动作关键姿态指标id，自增id |
| action\_pose\_id | Number(11) | Y | 动作关键姿态id |
| action\_id | Number(11) | Y | 动作id |
| action\_index\_id | Number(11) | Y | 动作指标id |
| summary | Varchar2(2048) | N | 说明 |
| create\_time | date | N | 创建时间 |
| update\_time | date | N | 最新一次修改时间 |
| status | Varchar2(128) | N | 状态 |

|  |  |  |  |
| --- | --- | --- | --- |
| action\_model\_version 动作模型版本表 | | | |
| 字段 | 类型 | 是否必填 | 说明 |
| id | Number(11) | Y | 自增id |
| action\_id | Number(11) | Y | 动作id |
| version | Number(11) | Y | 版本号 |
| model\_file\_path | Varchar2(2048) | N | 模型文件夹路径 |
| is\_used | Varchar2(128) | N | 已使用/未使用 |
| log | Blob | N | 日志内容 |
| create\_time | date | N | 创建时间 |
| update\_time | date | N | 最新一次修改时间 |
| status | Varchar2(128) | N | 状态 |

|  |  |  |  |
| --- | --- | --- | --- |
| action\_model\_publish\_log 动作模型发布日志表 | | | |
| 字段 | 类型 | 是否必填 | 说明 |
| id | Number(11) | Y | id |
| User\_id | Number(11) | Y | 操作员id |
| Model\_version | Number(11) | Y | 模型文件地址 |
| create\_time | date | N | 创建时间 |
| update\_time | date | N | 最新一次修改时间 |
| status | Varchar2(128) | N | 状态 |

|  |  |  |  |
| --- | --- | --- | --- |
| action\_model\_train\_log 动作模型训练日志表 | | | |
| 字段 | 类型 | 是否必填 | 说明 |
| id | Number(11) | Y | id |
| user\_id | Number(11) | Y | 操作员id |
| action\_id | Number(11) | Y | 动作id |
| model\_id | Number(11) | Y | 模型id |
| model\_version | Number(11) | Y | 模型版本 |
| model\_name | varchar2(70) | N | 模型名称 |
| model\_note | varchar2(1024) | Y | 模型说明 |
| file\_name | varchar2(70) | N | 模型文件名称 |
| file\_size | number(11) | Y | 模型文件大小 |
| file\_path | varchar2(200) | Y | 模型文件路径 |
| file\_time | time | N | 模型文件生成时间 |
| log | Blob | N | 训练过程日志 |
| create\_time | date | N | 创建时间 |
| update\_time | date | N | 最新一次修改时间 |
| status | Varchar2(128) | N | 状态 |

|  |  |  |  |
| --- | --- | --- | --- |
| action\_model\_execute\_log 动作模型使用日志表 | | | |
| 字段 | 类型 | 是否必填 | 说明 |
| id | Number(11) | Y | id |
| type | Number(11) | Y | 日志类型：1 -在线使用；2-离线使用 |
| ts | Number(11) | Y | 时间戳 |
| Req\_ip | Varchar2(128) | N | 请求地址 |
| Req\_url | Varchar2(128) | N | 请求url |
| Server\_ip | Varchar2(128) | N | 服务地址 |
| log | Blob | N | 日志内容 |
| Code | Varchar2(128) | N | 结果编码 |

#### 主要功能

1. **动作模型管理**



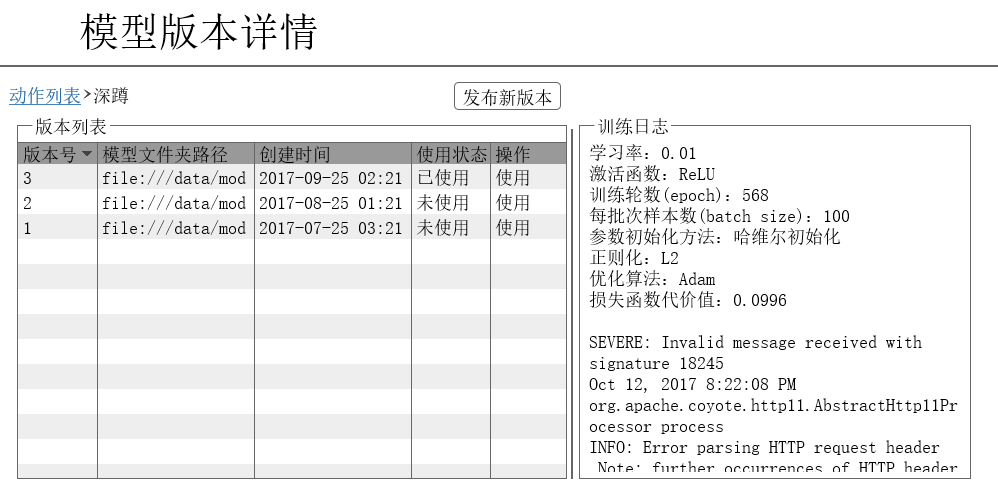
界面功能说明：

1. 此界面可以查看已经定义好的动作列表和动作描述情况。
2. 点击“动作详情”按钮可以查看动作的关键姿态定义和姿态对应的指标。
3. 点击“模型详情”按钮可以查看模型版本情况和模型训练日志。
4. 点击“日志详情”按钮可以查看模型的样本上传日志、训练日志、发布日志、使用日志等。
5. 点击“筛选”按钮，可以按文本框中的关键词筛选列表中的结果。
6. **动作详情**



界面功能说明：

1. 此界面可以查看动作的关键姿态定义和姿态对应的指标。
2. 单击关键姿态列表中某行，姿态对应的指标列表会显示在右侧表格中。
3. **模型详情**



界面功能说明：

1. 此界面可以查看模型版本情况和模型训练日志。
2. 点击“使用”按钮，会将对应版本的模型文件信息写入到tensorflow框架初始化配置文件中，当tensorflow重新初始化时，就会按照配置文件中模型版本载入相应的模型文件。点击使用的版本状态标记为“已使用”，其他版本状态标记为“未使用”。

配置文件model.conf格式如下：

[

{

"model\_id": "1",

"model\_name": "深蹲",

"model\_version": "3",

"file\_name": "mgyy\_mgAI\_Origin\_model\_3\_20170103\_00000001\_01",

"file\_size": "300M",

"file\_path":"file:///data/mgyy\_mgAI\_Origin\_model\_3\_20170103\_00000001\_01",

"file\_time": "2017-10-13 15:03:00"

},

{

"model\_id": "2",

"model\_name": "俯卧撑",

"model\_version": "4",

"file\_name": "mgyy\_mgAI\_Origin\_model\_4\_20170103\_00000001\_01",

"file\_size": "500M",

"file\_path":"file:///data/mgyy\_mgAI\_Origin\_model\_4\_20170103\_00000001\_01",

"file\_time": "2017-10-13 15:03:00"

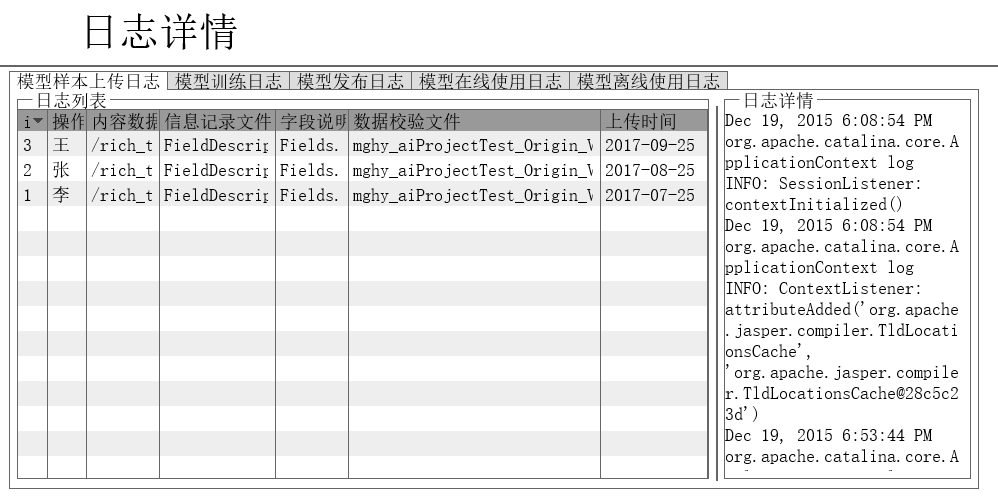
}

]

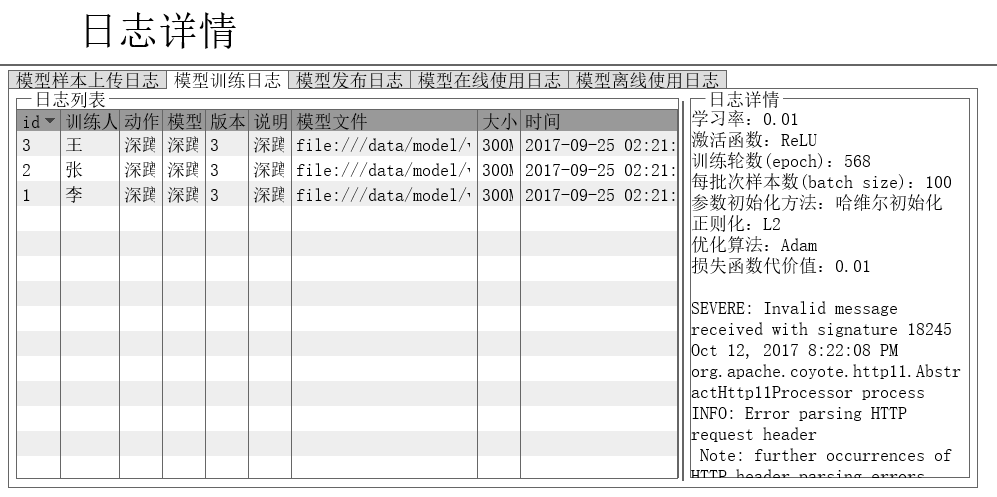
1. 点击“发布新版本”按钮，服务会到指定的文件夹下读取新的模型文件：如果有新的模型文件，则将模型版本加1，同时将模型信息添加到版本列表中，同时生成一条模型发布日志。如果没有新的模型文件，则不做处理。支持手动点击按钮触发，也支持后台每小时定时执行。
2. **日志详情**

界面功能说明：此界面可以查看以下日志内容。

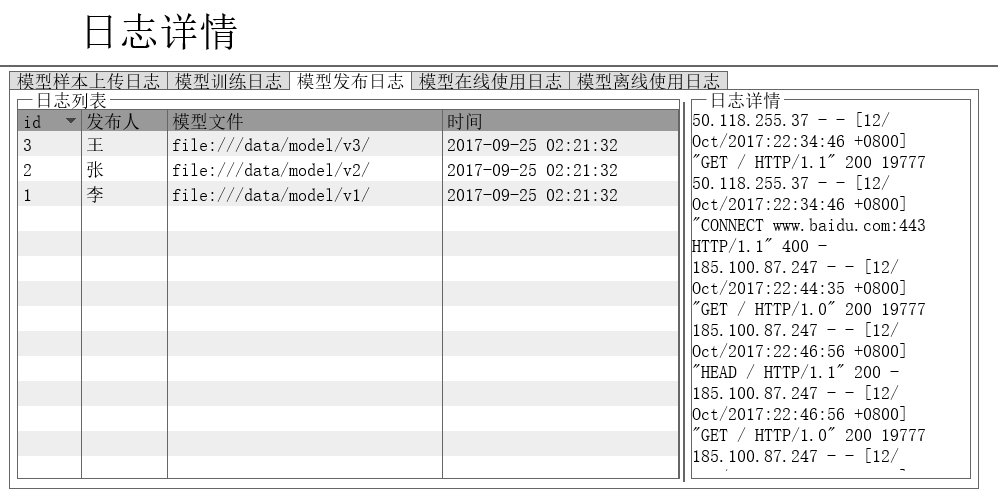
1. 模型样本上传日志。



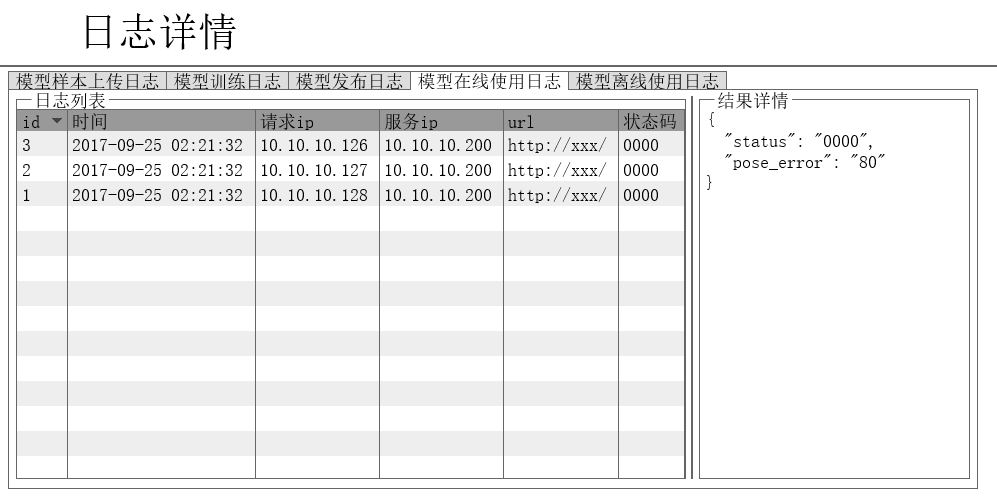
1. 模型训练日志



1. 模型发布日志。



1. 模型在线使用和离线使用日志。



### t动作指标同步

#### 数据表

动作模型表（action\_model）、动作关键姿态表（action\_pose）、动作指标表（action\_index）、姿态与指标对应表（action\_pose\_index）

#### 主要功能

善跑服务端将运营人员配置的动作、关键姿态、运动指标通过互娱内部的kafka消息队列同步给互娱大数据平台，互娱大数据平台接收后保存入数据表供样本数据标注时使用，标签数据为JSON格式，实时增量同步，主要字段如下：

|  |  |
| --- | --- |
| 功能 | 咪咕善跑向大数据平台同步动作 |
| Topic | t\_mgsp\_to\_bigdata\_action |
| 数据表 | 动作模型表（action\_model） |
| action\_id | 动作编码 |
| action\_name | 动作名称 |
| action\_desc | 动作描述 |
| oper\_type | 操作类型，1-新增，2-修改 |

|  |  |
| --- | --- |
| 功能 | 咪咕善跑向大数据平台同步关键姿态 |
| Topic | t\_mgsp\_to\_bigdata\_action\_pose |
| 数据表 | 动作关键姿态表（action\_pose） |
| action\_id | 动作编码 |
| pose\_id | 姿态编码 |
| pose\_name | 姿态名称 |
| pose\_desc | 姿态描述 |
| oper\_type | 操作类型，1-新增，2-修改 |

|  |  |
| --- | --- |
| 功能 | 咪咕善跑向大数据平台同步动作指标 |
| Topic | t\_mgsp\_to\_bigdata\_action\_index |
| 数据表 | 动作指标表（action\_index） |
| index\_id | 指标编码 |
| index\_name | 指标名称 |
| index\_type | 动作指标类型：1-角度；2-距离；3-位移 |
| index\_nodes | 指标涉及关节点编号列表 |
| index\_desc | 指标定义说明 |
| index\_function | 指标计算方法 |
| oper\_type | 操作类型，1-新增，2-修改 |

|  |  |
| --- | --- |
| 功能 | 咪咕善跑向大数据平台同步动作姿态和指标对应 |
| Topic | t\_mgsp\_to\_bigdata\_action\_pose\_index |
| 数据表 | 姿态与指标对应表（action\_pose\_index） |
| action\_id | 动作编码 |
| pose\_id | 姿态编码 |
| index\_id | 指标编码 |
| oper\_type | 操作类型，1-新增，2-修改 |

### 训练视频采集

#### 数据表

动作样本表（action\_sample）

|  |  |  |  |
| --- | --- | --- | --- |
| action\_sample 动作样本表 | | | |
| 字段 | 类型 | 是否必填 | 说明 |
| **id** | Number(11) | Y | 样本id |
| action\_id | Number(11) | Y | 动作id |
| pose\_path | Varchar2(2048) | Y | 动作捕捉仪数据地址 |
| video\_path | Varchar2(2048) | Y | 视频文件地址 |
| gender | Number(11) | N | 教练性别：1-男，0-女 |
| height | Number(11) | N | 教练身高 |
| summary | Varchar2(2048) | N | 说明 |
| is\_labeled | Varchar2(128) | N | 已标注/未标注 |
| is\_uploaded | Varchar2(128) | N | 已上传/未上传 |
| create\_time | date | Y | 创建时间 |
| status | Varchar2(128) | N | 状态 |

#### 主要功能

1. **模型样本管理**



**界面功能说明：**

1. 此界面可以查看已经采集好的样本列表。
2. 点击“采集新样本”按钮，进入样本采集界面。
3. 点击“标注”按钮，进入标注界面，对未标注的样本进行标注操作。
4. 点击“上传”按钮，可以将标注完成的样本按照3.1.6的文件命名规范进行处理，并存储到指定目录下。同时产生一条样本上传纪录到样本上传日志表。
5. 点击“删除”按钮，可以删除当前纪录对应的样本数据。并将数据库纪录状态标记为已删除。
6. **模型样本采集**



**界面功能说明：**

数据采集平台负责采集的原始样本供下一步规范化的数据标注，即完成对多个摄像头数据/动作捕捉仪数据流数据的同步与读写，记录动作名，教练属性，样本数据地址，并将样本写入数据库中。具有连接设备、录制/存储数据，记录教练身高、性别，显示动作所涉及的关键姿态与各个关键姿态对应的指标名等功能。

**使用流程：**

* 输入教练身高、选择所做的动作种类、选择输出路径。
* 点击连接查看4个数码摄像机/手机的视频画面和一个动作捕捉仪的数据仿真画面；
* 调整机位与教练运动的位置，直至该动作种类所涉及的关键人体关节能够在所有画面中无遮挡且全程可见；
* 点击录制按钮开始采集训练数据，教练开始运动，执行同一类动作（如深蹲），且一段视频中完成动作数不低于10次；
* 点击结束录制按钮完成录制，存储视频文件与动作捕捉仪数据文件到指定目录，并将样本记录写入动作样本表。

### 样本数据标注

#### 数据表

动作姿态的动作指标标注表（action\_pose\_index\_label）

|  |  |  |  |
| --- | --- | --- | --- |
| action\_pose\_index\_label动作姿态的动作指标标注表 | | | |
| 字段 | 类型 | 是否必填 | 说明 |
| **id** | Number(11) | Y | 指标标注id，自增id |
| action\_pose\_index\_id | Number(11) | Y | 动作姿态的动作指标id |
| action\_sample\_id | Number(11) | Y | 样本id |
| action\_index\_values | Varchar2(2048) | N | 动作指标值列表{时间:指标值} |
| summary | Varchar2(2048) | N | 说明 |
| create\_time | date | N | 创建时间 |
| update\_time | date | N | 最新一次修改时间 |
| status | Varchar2(128) | N | 状态 |

#### 主要功能

1. **模型样本标注**



**界面功能说明：**

数据标注平台负责将采集的原始样本进行规范化的数据标注，即完成对原始视频/动作捕捉仪数据流数据打标签，人工标注出其中的各个关键姿态种类与时刻，并通过自动的就算该关键姿态所涉及的动作指标值记录该关键姿态的标准标注结果，并将标注结果写入数据库中。具有播放/暂停样本数据、逐帧向前/向后查看样本数据，显示实时样本画面与动作指标值等功能。

**标注流程：**

* 在模型样本管理界面上，点击某条样本视频记录后的“标注”按钮进行该视频的标注界面。
* 点击“存储目录选择”按钮，选择输出路径
* 点击“播放/停止”按钮，找到关键姿态时刻，下拉选择关键姿态名，点击“添加时刻”按钮将会在关键姿态时刻列表中添加一条记录，并通过程序自动计算获得该时刻的指标值，依次标注完整个样本；
* 点击“保存标注”，将完整的标注结果保存到标注xml文件中，包含该段样本中的所有关键姿态名与出现的时刻、各个关键姿态对应的运动指标值。
* 最终输出的XML文件格式如下，与数据表字段相互对应。

<annotation verified="no">

<ID>样本id</ID>

<actionid>动作id</actionid>

<source>

<mcpath>动作捕捉仪数据文件路径</mcpath>

<campath>视频数据文件路径</campath>

</source>

<height>教练身高</height>

<gender>教练性別</gender>

<action\_pose>

<id>姿态id1</id>

<action\_index>

<id>指标id1</id>

<value>value</value>

</action\_index>

<action\_index>

<id>指标id2</id>

<value>value</value>

</action\_index>

</action\_pose>

<action\_pose>

<id>姿态id2</id>

<action\_index>

<id>指标id1</id>

<value>value</value>

</action\_index>

<action\_index>

<id>指标id2</id>

<value>value</value>

</action\_index>

</action\_pose>

</annotation>

### 数据上传云平台

#### 数据表

样本上传表（aciton\_sample\_upload）

|  |  |  |  |
| --- | --- | --- | --- |
| action\_sample\_upload 样本上传表 | | | |
| 字段 | 类型 | 是否必填 | 说明 |
| id | Number(11) | Y | id |
| User\_id | Number(11) | Y | 操作员id |
| sample\_id | Number(11) | Y | 样本id |
| rich\_text | Varchar2(2048) | Y | 内容数据文件 |
| Fields | Varchar2(2048) | Y | 信息记录文件 |
| FieldDescription | Varchar2(2048) | Y | 字段说明文件 |
| Verf | Varchar2(2048) | Y | 数据校验文件 |
| create\_time | date | N | 创建时间 |
| update\_time | date | N | 最新一次修改时间 |
| status | Varchar2(128) | N | 状态 |

#### 主要功能

参照《咪咕AI能力数据规范》，定义了数据上传云平台的文件名命名规则，以下分别对内容数据文件、信息纪录文件、字段说明文件、标注结果文件、数据校验文件、校验日志和样本数据上传表进行了详细定义。在模型样本管理界面上，点击某个样本视频记录后的“导出”按钮，依据数据规范导出数据到本地电脑，最后由运维人员手工拷贝到互娱的FTP前置服务器上。

1. 内容数据文件

该文件为咪咕善跑向AI云平台提供的符合数据规范要求的内容数据，包括善跑的训练视频、标注结果等数据。该内容数据需要提供给AI云平台中的学习子平台进行模型能力训练，数据全部为预处理后数据，源数据与预处理数据为1：1对应，即1个源数据会对应1个预处理后的数据。此时预处理后的数据命名规则如下，其中[项目名]中不能包含下划线“\_”，[重传序号]默认为两位，为数字型，第一次传输是采用“00”，每次重传该值递增1。

mg[hy\_shanpaoAI\_Origin\_Video\_20170103\_00000001\_Preprocessing\_00.mp4](mailto:yy_mgAI_Origin_Image@1504529951_00000001_Preprocessing_00000001.jpg)

[子公司名]\_[项目名]\_ [Origin] \_[源文件类型]\_[源日期]\_[源自增标号（8位）]\_ [Preprocessing]\_[重传序号].文件类型

1. 信息记录文件
2. 当接口中有多个内容数据文件时，信息记录文件只有一个。
3. 信息记录文件记录内容数据中的各信息字段的取值，例如文件id、所属项目、生成日期等。
4. 信息记录文件统一采用Fields\_[重传序号].json或Fields\_[重传序号].csv命名。标识该接口的内容数据文件信息。
5. 字段说明文件

字段说明文件用以说明信息记录文件中各个字段的含义。信息字段说明文件统一命名为FieldDescription\_[重传序号].txt，为UTF-8编码的文本格式。

1. 标注结果文件

善跑的内容数据文件与标注结果文件为一一对应关系，标注结果文件命名规则如下，其中[重传序号]默认为两位，为数字型，第一次传输是采用“00”，每次重传该值递增1。

[mghy\_shanpaoAI\_Origin\_Video\_20170103\_00000001\_](mailto:yy_mgAI_Origin_Image@1504529951_00000001_Preprocessing_00000001.jpg)Preprocessing\_00\_label\_00\_00.xml

[被标注的文件名称]\_ [标注日期]\_[label]\_[标注序号]\_[重传序号].文件类型

1. 数据校验文件

当接口中有多个文件时，校验文件只有一个。标识该接口内所有数据校验信息，包含文件大小、生成日期、校验码等。该文件用于数据提供方向数据接收方提供的接口数据校验的动态校验规则。

校验文件命名规则如下，校验文件命名规则中的[子公司名]取值为mghy， [分类]字段取值为Origin或Preprocessing，根据校验文件所在目录属于预处理前（Origin）或预处理后（Preprocessing）决定；[日期]字段为该校验文件生成的时间，格式为YYYYMMDD。[重传序号]默认为两位，为数字型，第一次传输是采用“00”，每次重传该值递增1。重传校验文件中只记录重传数据的校验信息。

mg[hy\_shanpaoAI\_Origin\_20170712\_00.verf](mailto:yy_mgAI_Origin_Image@1504529951_00000001.jpg)

[子公司名]\_[项目名]\_[分类]\_[日期] \_ [重传序号].文件类型

### 视频模型训练

#### 数据表

动作模型表（action\_model）、模型版本表（action\_model\_version）、模型日志表（action\_model\_log）

#### 主要功能

根据数据标注的结果，对每个动作分别进行模型训练，形成模型文件。模型要可以跟踪并识别用户运动全过程各个肢体部位的动作。训练准确度不低于90%，，在单片GPU下1秒处理至少15帧480p图像。

模型训练系统环境要求：

|  |  |  |  |
| --- | --- | --- | --- |
| 运行环境 | 要求 | 备注 | 运行环境 |
| gcc版本 | 4.8及以上 |  | gcc版本 |
| Python版本 | 2.7.10以上，3.0以下 | python3.0及以上版本与2.x系列版本不兼容 | Python版本 |
| Cuda/cublas版本 | 推荐6.5及以上 |  | Cuda/cublas版本 |
| Cudnn版本 | 推荐5.0及以上 |  | Cudnn版本 |
| GPU驱动 | NV驱动 381.04 |  | GPU驱动 |
| Linux | 操作系统内核Centos7.2 |  |  |
| TensorFlow | 推荐1.2.0及以上 |  |  |

模型训练需要依赖标注好的标准训练动作数据和网络结构，模型训练步骤：

1. **训练环境部署**

在进行训练之前，需要将训练所需的预训练数据集、代码、预处理脚本等都配置到统一工程目录中，在服务器上建立目录结构如下：

Work

-dataset(训练数据目录)

-code(训练代码目录)

- script(脚本目录)

- model(训练模型结果)

- YYMMDDHH(训练目录)

* 数据集部署

预训练数据目录下存放作为模型的预训练数据，并和自有的预标注数据结合进行训练。训练前需要将各个数据集的数据与标注格式统一，拟采用MSCOCO的格式作为标准格式，通过脚本可以将各个数据集的格式转换为标准的数据格式。

1. 从外置存储设备中获取MSCOCO、MPII、Human3.6M等人体位姿估计相关的公开数据集；
2. 修改脚本中预设的各数据集路径，并执行脚本bash script/getData.sh获取所有的原始数据到dataset/raw\_data/，同时获取关键点标注到dataset/annotations/；
3. 从AI云平台获取自有的标注过的样本数据；
4. 执行脚本bash script/split.sh将所有原始数据随机划分为训练集(80%)，验证集(20%)，划分的数据索引将存储在dataset/split/train.txt和dataset/split/val.txt中。

* 训练依赖环境部署

在完成数据准备后，需要配置训练环境，单目人体位姿估计的网络是在原有的网络结构基础上进行修改，将网络输出模块修改为可以回归位姿的卷积姿态机(CPM)模型。

1. 从外部存储设备中获取人体位姿估计网络的代码，放置code/pose目录下。
2. 从外部存储设备中获取VGG19的网络结构文件vgg-19.prototxt和初始参数模型vgg-19.caffemodel，放置在model/VGG/目录下。
3. 执行mkdir model/YYMMDDHH在model/目录下新建训练输出文件夹model/YYMMDDHH，并执行cp model/VGG/vgg-19.prototxt model/YYMMDDHH/vgg-19-cpm.prototxt复制原始网络结构文件到该文件夹下。
4. 修改model/YYMMDDHH/vgg-19-cpm.prototxt文件，在文件中添加cpm层如下：

layer {

name: "data"

type: "CPMData"

top: "data"

top: "label"

data\_param {

source: "data/lmdb\_train"

batch\_size: 10

backend: LMDB

}

cpm\_transform\_param {

stride: 8

max\_rotate\_degree: 40

visualize: false

crop\_size\_x: 368

crop\_size\_y: 368

scale\_prob: 1

scale\_min: 0.5

scale\_max: 1.1

target\_dist: 0.6

center\_perterb\_max: 40

do\_clahe: false

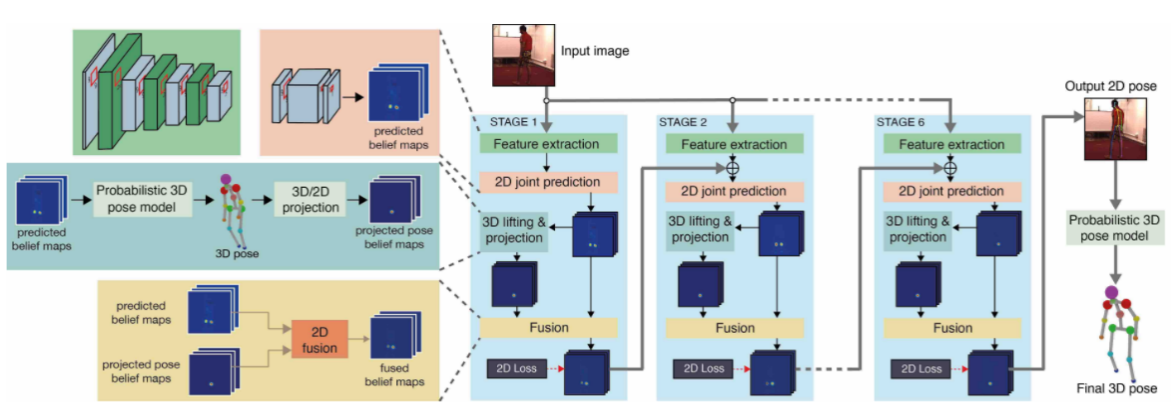
num\_parts: 56

np\_in\_lmdb: 14

}

}

网络采用多阶段的卷积神经网络，融合3D人体姿态的概率知识完成从单张原始RGB图像中估计3D人体姿态，联合训练2D关节估计和3D姿态重建两个任务。利用经验上的3D结点位置的先验关系来优化寻找更好的2D位置。整个任务设计为高效的端对端的结构，兼顾准确性和效率。生成的网络结构图如下：



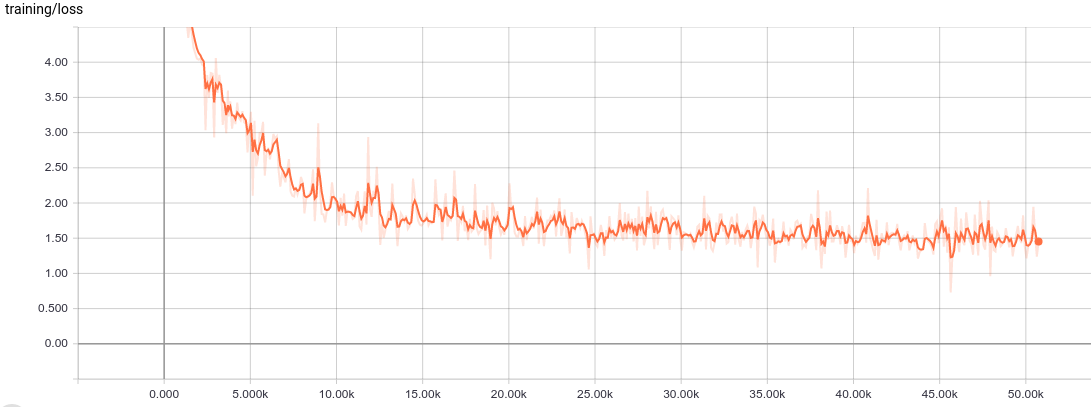
1. **模型训练**

数据与环境准备完成后就可以开始执行训练，需确定训练的超参数，学习平台所可配置的超参数包括而不限于以下：

|  |  |  |
| --- | --- | --- |
| 超参数名称 | 可取值 | 默认值 |
| 学习率 | (0, 1) | 0.01，并随训练周期衰减 |
| 激活函数 | Sigmoid, Tanh, ReLU, Leaky ReLU, PReLU, SeLU | ReLU |
| 训练轮数(epoch) | 实验中调优 |  |
| 每批次样本数(batch size) | 实验中调优 |  |
| 参数初始化方法 | 高斯、有偏高斯、哈维尔(Xavier) | 哈维尔初始化 |
| 正则化 | L1、L2、不启用 | 不启用 |
| dropout | 实验中调优 |  |
| 优化算法 | SGD、Momentum、RMSProp、Adam | Adam |

给定训练的超参数(如学习率、最大迭代次数、权重衰减系数、正则化系数等参数)等，即可以开始迭代训练，步骤如下：

1. 执行 python script/setLayers.py model/YYMMDDHH/vgg-19-cpm.prototxt --exp 1生成网络训练的shell脚本文件train\_pose.sh，该脚本文件里包含有训练的各种超参数
2. 执行bash train\_pose.sh 0,1开始训练(使用GPU0和GPU1)，并将实时打印出训练过程中的代价日志model/YYMMDDHH/logfile.log
3. 训练过程中使用脚本python monitor\_log.py model/YYMMDDHH/logfile.log –verbose –loss 1绘制训练代价值与迭代次数的关系，如下图所示，横坐标为训练次数，纵坐标为训练代价值。



等待至训练代价值完全收敛或达到最大迭代次数则训练结束。

1. **训练结果处理**

当模型训练完成后，可以选择保存模型，或者调整训练参数重新训练。

每次训练完成后就会产生一个新的模型版本，通过脚本输出模型版本结果文件存储在训练服务器上，对于每个训练得到的模型版本还需记录以下日志信息。

|  |  |  |  |
| --- | --- | --- | --- |
| **字段** | **类型** | **可为空** | **含义** |
| user\_name | varchar2(70) | N | 用户名 |
| model\_name | varchar2(70) | N | 模型名称 |
| model\_note | varchar2(1024) | Y | 模型说明 |
| file\_name | varchar2(70) | N | 模型文件名称 |
| file\_size | number(11) | Y | 模型文件大小 |
| file\_path | varchar2(200) | Y | 模型文件路径 |
| file\_time | time | N | 模型文件生成时间 |

日志文件示例

{

"user\_name": "王",

"model\_name": "深蹲",

"model\_note": "深蹲",

"file\_name": "mgyy\_mgAI\_Origin\_model\_1\_20170103\_00000001\_01",

"file\_size": "300M",

"file\_path":"file:///data/mgyy\_mgAI\_Origin\_model\_1\_20170103\_00000001\_01",

"file\_time": "2017-10-13 15:03:00"

"log":"学习率：0.01,激活函数：ReLU,训练轮数(epoch)：568,每批次样本数(batch size)：100,参数初始化方法：哈维尔初始化,正则化：L2,优化算法：Adam,损失函数代价值：0.01"

}

将上述日志信息写入日志文件后，与模型版本文件共同存储在训练服务器的指定目录model/YYMMDDHH中，并执行model.zip model/YYMMDDHH将模型训练目录压缩打包等待发布。

### 训练结果发布

#### 数据表

动作模型版本表（action\_model\_version）、动作模型发布日志表（action\_model\_publish\_log）

#### 主要功能

在模型训练人员完成模型训练后，可以对模型文件新版本进行发布，即算法人员手工将模型文件从AI云平台下载到AI前置平台接口机上。模型的存储路径与命名规范如下图示：

mg[yy\_mgAI\_Origin\_model\_1\_20170103\_00000001\_01.zip](mailto:yy_mgAI_Origin_Image@1504529951_00000001.jpg)

-model\_20170103\_00000001\_01.prototxt(模型网络结构)

-model\_20170103\_00000001\_01.caffemodel(模型文件)

- model\_20170103\_00000001\_01.json(模型日志文件)

-model\_20170103\_00000001\_01.verf(数据校验文件)

[子公司名]\_[项目名]\_[Origin]\_[文件类型]\_[版本]\_[日期]\_[自增标号（8位）]\_[重传序号].文件类型

注：

1. [日期]为年月日日期（YYYYMMDD）格式，必须与其存储目录中的[日期]一致；[重传序号]默认为两位，为数字型，第一次传输是采用“00”，每次重传该值递增1。
2. [项目名]中不能包含下划线“\_”。
3. 数据校验文件第一列代表id文件名（包括相对目录，格式为rich\_text\id），第二列为MD5值，第三列为文件大小。所有文件都需要校验，包括Fields.json和FieldDescription.txt。

具体操作步骤为：

1. 算法人员通过scp命令将待发布文件传输到AI云平台接口机的指定目录下。
2. 算法人员通过手工方式将模型文件从AI云平台接口机下载到AI前置服务平台的指定目录下。
3. 算法人员通过模型管理界面上的“发布新版本”功能，通知系统到接口机的指定的文件夹下读取新的模型文件。如果有新的模型文件，则根据模型日志文件中的内容向模型版本表中新增一条版本记录，同时生成一条模型发布日志，日志字段定义见3.1.1节。
4. 算法人员在模型管理界面上选择某个模型版本记录，点击“使用”按钮，将该版本的模型文件发布到Tensorflow学习框架中。

## 视频模型使用

### 视频直播及解析

1. 数据表

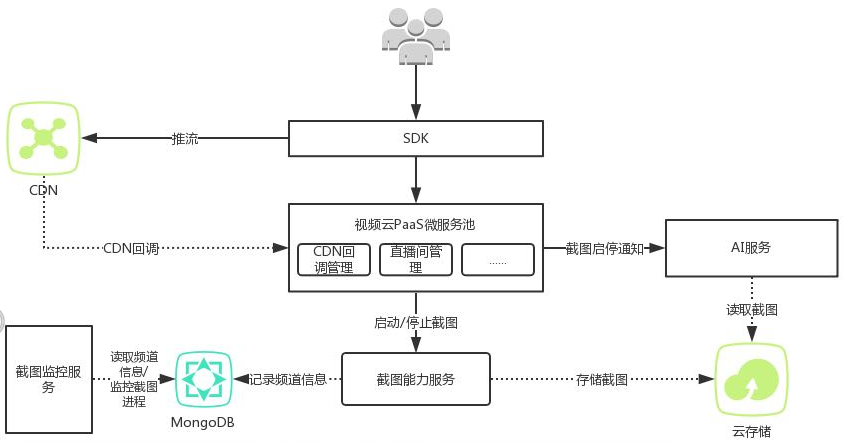
动作截图表（action\_image）、动作处理视频表（action\_deal\_video）

|  |  |  |  |
| --- | --- | --- | --- |
| action\_image动作截图表 | | | |
| 字段 | 类型 | 是否必填 | 说明 |
| **id** | Number(11) | Y | 动作id，自增id |
| user\_id | varchar2(100) | Y | 善跑用户ID |
| action\_id | varchar2(100) | Y | 动作编码 |
| video\_id | varchar2(100) | Y | 拍摄视频ID |
| image\_name | varchar2(200) | Y | 截图名称 |
| image\_url | varchar2(1024) | Y | 截图路径 |
| image\_frame | number | N | 截图帧号 |
| image\_time | Number | N | 截图时间点 |
| in\_time | date | Y | 入库时间 |

|  |  |  |  |
| --- | --- | --- | --- |
| action\_deal\_video动作处理视频表 | | | |
| 字段 | 类型 | 是否必填 | 说明 |
| **id** | Number(11) | Y | 动作id，自增id |
| user\_id | varchar2(100) | Y | 善跑用户ID |
| action\_id | varchar2(100) | Y | 动作编码 |
| video\_id | varchar2(100) | Y | 拍摄视频ID |
| video\_name | varchar2(200) | Y | 视频名称 |
| video\_url | varchar2(1024) | Y | 视频路径 |
| in\_time | date | Y | 入库时间 |

1. 主要功能

善跑用户使用训练指导功能时需要直播自己的训练过程，善跑客户端内嵌咪咕视讯的直播SDK，直播SDK会调起移动终端的摄像头拍摄用户训练视频，实时将视频流程压缩转换协议传输至咪咕视讯服务端。咪咕视讯服务端接收到视频流后，实时将视频流按时间序列逐帧转成图片存储在咪咕云上，同时将视频流录制成视频，视频也保存在咪咕云上。咪咕视讯服务端实时回调互娱大数据平台的接口地址，将视频截图、录制视频在咪咕云上的编号和地址回传给互娱大数据平台，互娱大数据平台接收后将地址保存入数据库，作为运动视频理解模型的输入。



主要流程如下：

1. 善跑客户端调用直播SDK的创建直播接口，将善跑用户的userid、训练动作的actionid拼接成userid\_actionid作为接口的userid输入参数，直播SDK返回chnannlid。
2. 善跑客户端调用直播SDK的采集直播接口，开始向视讯CDN推送视频流，CDN通知视讯的PaaS服务接收视频流启动截图服务。
3. 视讯PaaS服务调用互娱大数据平台上的截图开始通知接口，传递截图在咪咕云上的存储目录地址、时间戳、chnannlid、userid，目录以userid\_actionid\_channelid形式命名。
4. 视讯PaaS服务开始对视频流截取图片，每秒视频截取5张图片，图片的命名规范为userid\_actionid\_channelid\_时间戳\_序号.jpg，图片分辨率遵循直播流的分辨率（推荐640\*480），时间戳序号为1-5，截图存储在第3步提及的咪咕云存储目录中，时间戳为系统时间（非视频播放时间），格式为YYYYMMDDHHMMSS。
5. 互娱大数据平台从接收到开始截图通知后，从接口返回参数的时间戳+1秒开始，每隔1秒创建一个线程，首先按照第4步中的视频截图命名规范构造截图名称，再从咪咕云上下载截图到本地。如果下载失败需要重试3次。图片下载成功后将图片的本地路径保存入本地Redis。
6. 用户如果主动停止直播或由于APP异常中断直播，用户正常结束直播善跑客户端需调用直播SDK的直播关闭接口，异常中断由咪咕视讯服务端自动检测后并通知互娱大数据平台。
7. 直播结束后，视讯服务端检测到直播状态为结束或异常，会通知互娱大数据平台，如直播状态为暂停，则大数据平台调用视讯服务端的直播关闭接口中止直播，关闭成功后停止线程下载截图，再调用视讯服务端的删除图片接口，将咪咕云上的视频截图目录及其中图片一并删除。
8. 视讯服务端在直播关闭后，录制视频并保存到咪咕云上，通知互娱大数据平台视频的保存地址。

直播SDK支持Android（5.0系统及以上）、iPhone（iPhone5及以上iOS8及以上）设备前、后摄像头进行用户动作视频拍摄，视频格式为MP4，时长不超过90秒；视频截图图片格式为JPG，分辨率不低于640\*480像素，每秒截图不少于5张。

### 模型在线使用

模型在线使用是指善跑用户在训练过程中，从AI模型动态获取训练评价。善跑AI模型在线服务部署在AI前置服务平台上，服务方式为RESTAPI服务。

咪咕善跑客户端调用咪咕善跑服务端的AI服务接口，咪咕善跑服务端调用互娱大数据平台的AI服务接口，互娱大数据平台根据userid、actionid、channelid从Redis中查询用户最新的截图地址，再将请求转发到AI前置服务平台，获取结果后逐级返回至善跑客户端向用户呈现。

运动训练指导模型在线使用服务的RESTful URL为http://migu\_url/shanpao/training/action/online\_instruction/v1/，使用POST方法调用RESTAPI，参数设计如下：

**传输方式：**HTTP-REST JSON

**传输方式：**GET/POST

**数据编码：**UTF-8

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **描述** | **是否必填** | **备注** |
| userid | 善跑用户编号 | Y | 善跑用户编号 |
| actionid | 训练动作编号 | Y | 训练动作编号 |
| channelid | 直播频道号 | Y | 直播频道号 |
| frames\_img | 图像帧地址 | Y | 图像帧地址数据，每地址以逗号隔开。图像的命名规范为[userid]\_[actionid]\_[channelid]\_[帧对应的时刻]\_[序号].jpg |

**返回结果：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **描述** | **备注** |
| code | 状态码 | 0000：成功；状态码详细定义见4 系统响应码 |
| pose\_error | 动作偏差 | 动作偏差百分比 |

**请求示例：**

http://migu\_url/shanpao/training/action/online\_instruction/v1?userid=mgsp201700001&actionid=action0001&channelid=channel1&frames\_img=user1\_action1\_channel1=\_1.jpg,user1\_action1\_channel1\_6.jpg,user1\_action1\_channel1\_11.jpg

**JSON返回示例：**

{

"status": "0000",

"pose\_error ": "80"

}

**接口性能：**最大并发用户100，响应时间不高于1.5秒

### 模型离线使用

模型离线使用是指善跑用户在训练接收后，从AI模型观看处理后视频并获取训练评价。善跑AI模型离线服务部署在AI前置服务平台上，服务方式为RESTAPI服务。

咪咕善跑客户端调用咪咕善跑服务端的AI服务接口，咪咕善跑服务端调用互娱大数据平台的AI服务接口，互娱大数据平台根据userid、actionid查询用户最新的录制视频，再将请求转发到AI前置服务平台，获取结果后逐级返回至善跑客户端向用户呈现。

运动训练指导模型离线使用服务的RESTful URL为http://migu\_url/shanpao/training/action/offline\_instruction/v1/，使用POST方法调用RESTAPI的参数设计如下：

**传输方式：**HTTP-REST JSON

**传输方式：**GET/POST

**数据编码：**UTF-8

**输入参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **描述** | **是否必填** | **备注** |
| user\_id | 善跑用户编号 | 是 | 善跑用户编号 |
| action\_id | 训练动作编号 | 是 | 训练动作编号 |
| channel\_id | 直播频道号 | Y | 直播频道号 |
| video\_path | 视频地址 | 是 | 视频地址。视频的命名规范为[userid]\_[actionid]\_[channelid].mp4 |

**返回结果：**

|  |  |  |
| --- | --- | --- |
| **参数名称** | **描述** | **备注** |
| code | 状态码 | 0000：成功；状态码详细定义见4 系统响应码 |
| pose\_error | 动作偏差 | 动作偏差百分比加权平均值 |
| video\_path | 视频地址 | 处理后的视频地址。视频的命名规范为[userid]\_[actionid]\_[channelid]\_labeled.mp4 |
| image\_path | 图像地址 | 选取的图像地址。图像的命名规范为[userid]\_[actionid]\_[channelid]\_[帧对应的时刻]\_[序号]\_labeled.jpg |

**请求示例：**

http://migu\_url/shanpao/training/action/offline\_instruction\_result/v1?userid=mgsp201700001&actionid=action0001&channelid=channel1&video\_path=user1\_action1\_channel1.mp4

**JSON返回示例：**

{

"status": "0000",

"pose\_error": "85",

"video\_path":"file:///data/mghy/shanpao/training/action/offline\_instruction\_result /12/video/user1\_action1\_channel1\_labeled.mp4",

"image\_path":

[

<file:///data/mghy/shanpao/training/action/offline_instruction_result/12/image/user1_action1_channel1_labeled_image0.jpg>,

"file:///data/mghy/shanpao/training/action/offline\_instruction\_result/12/image/user1\_action1\_channel1\_labeled\_image1.jpg",

"file:///data/mghy/shanpao/training/action/offline\_instruction\_result/12/image/user1\_action1\_channel1\_labeled\_image2.jpg"

]

}

**接口性能：**最大并发用户100，响应时间不高于3秒

## 跑步路线推荐

### 数据表



### 运动轨迹接入

1. **运动轨迹接入**
2. 每日凌晨1:00使用CrossData从善跑的Oracle数据库的T\_MGSP\_SPORT\_RECORD表中同步前一日的运动记录数据至大数据平台的Oracle数据库的ods\_sp\_sport\_record表中，表结构以善跑侧为准。
3. 运行Oracle存储过程，处理善跑数据，将数据从ods\_sp\_sport\_record表转移至ods\_sp\_record\_deal表，其中所属区域、所属城市、起点经度、起点纬度、推荐轨迹5个字段暂为空。各标签的等级暂定如下

|  |  |  |
| --- | --- | --- |
| **序号** | **指标** | **分类** |
|  | 时间 | 深夜23-5，早上5-8，上午8-12，中午12-14，下午14-18，晚上18-23 |
|  | 里程 | A>42.195km，B 20-42.195，C 10-20，D 5-10，E 1-5，F<1 |
|  | 人数 | A >10000人次，B 1000-10000，C 100-1000，D 10-100，E <10 |
|  | 耗时 | A>300分钟，B 180-300，C 60-180，D 10-60，E1-10，F<1 |

1. 在互娱大数据平台的服务中心中为善跑开发定制化数据处理接口，http形式，逐行读取所属日期下的所有运动轨迹，根据轨迹id去hbase表中查询轨迹的起点、终点，调用算法中心接口获取轨迹所属区域的编号，调用GIS引擎接口获取轨迹所属的城市。获取上述数据后，update互娱大数据平台Oracle的ods\_sp\_record\_deal表中的记录。

**二、轨迹点接入**

1. 目前善跑服务端会将跑步路线+轨迹点数据组装后，实时发送到善跑内部的Kafka队列上，每条路线形成一条消息，大数据平台需要从Kafka上接入路线和轨迹点。

Kafka地址：172.19.219.139:9092,172.19.219.140:9092,172.19.219.141:9092

Topic：t\_mgsp\_medal\_sport

1. 在互娱大数据平台配置Streaming从善跑Kafka上拉取路线消息记录保存入Oracle的临时表中tmp\_sp\_sport\_record，临时表存储消息原始的JSON字符串，不做任何解析，需要加上时间戳。消息示例如下，字段含义同现有Oracle运动记录表ods\_sp\_sport\_record和HBase的轨迹点表ods\_sp\_sport\_point，JSON对应的实体类。
2. 在互娱大数据平台的Safe中心开发数据接入定制服务，首先配置定时任务定时读取Oracle临时表中的固定条数记录（按时间从前到后排序），其次即系JSON后将跑步路线接入运动记录表ods\_sp\_sport\_record，然后将轨迹点按照2.3.2节的处理逻辑处理后接入轨迹点表ods\_sp\_sport\_point，最后将记录汇总后存入运动记录宽表ods\_sp\_record\_deal，汇总SQL见现有的统计存储过程。同时调用GIS引擎获取路线起点、终点的地理位置名称，存入Oracle的ods\_sp\_record\_deal表。
3. 记录成功后，更改Oracle临时表中的对应记录的status字段值。
4. 从善跑内部的kafka接口接入轨迹数据，需要解析其中JSON格式的步幅点数据，数据字段含义为a:海拔高度、b:步频、c:步幅、d:排序号、s:速率。输入直接使用insert into方式保存入hbase表ods\_sp\_sport\_cadence\_hb。

### 热点区域划分

1. **每月更新区域**
2. 按城市将近3个月（首次为2017年10月-2017年12月）的轨迹点进行划分，使用ods\_sp\_sport\_point表中的city\_code、record\_date字段。（1）如某个城市的运动记录数小于1000不参与计算。（2）如轨迹的status字段<>1、suspicious\_leave<>1、2、3则轨迹不参与计算。
3. 以城市为单位使用密度算法将轨迹点分成多个区域，然后判断某个运动记录的起点在一个区域内，则认为该记录属于该区域。此步计算区域半径取0.003度，需要按照轨迹点密集程度分波次计算，分别为>180个点、120个-180个、60个-120个。
4. 第一轮计算后如某个区域内的轨迹点总数大于阀值，说明该区域由多个小区域合并而成，需要对这个大区域中的轨迹点缩小聚类半径重新进行聚类，依次类推直至区域中的轨迹点数满足要求。
5. 统计每个区域内所有轨迹点的经纬度坐标的平均值，以平均值作为中心点代表本区域，区域信息保存入dim\_sp\_area表，需要更新batch\_no字段，每个批次的区域不一样。
6. 轨迹点和区域对应关系覆盖更新HT中的ods\_sp\_sport\_point\_bak表。
7. 生成轨迹ID后，更新ods\_sp\_record\_deal表中的batch\_no、area\_id字段值，标识出每个运动记录所属的区域。
8. 最新的区域城市、编号、中心经纬度保存入Redis，分成HMap和Sorted-Set两种格式。HMap格式为：<城市名,< 区域编号,经度\_维度>)，如<南京市,<201，21.45178911\_32.98112822>>。Sorted-Set格式见下方第二个表。
9. Redis表设计

|  |  |
| --- | --- |
| 功能 | 保存热点区域 |
| 类型 | Hash |
| Key | 城市名 |
| Field | 区域编号 |
| Value | 经度\_维度 |
| 前缀 | sp\_hot\_area |
| 示例 | sp\_hot\_area:南京市,<batch\_no,20170109>,<10001, 21.45178911\_32.98112822>,<10002, 21.45178911\_32.98112822> |

**第一个field是批次号，如：**<batch\_no,20170109>

|  |  |
| --- | --- |
| 功能 | 按中心点经纬度顺序保存热点区域 |
| 类型 | Sort-Sets |
| Key | 城市名\_Lat（维度）, 城市名\_lon（经度） |
| Field | 区域编号 |
| Value | 经度或维度的值 |
| 前缀 | 无 |
| 示例 | 南京市\_lat,<10001, 21.45178911>,<10002, 32.98112822>  南京市\_lon,<10001, 21.45178911>,<10002, 32.98112822> |

**二、实时更新区域**

1. 按城市将新增的轨迹点进行划分，临时表ods\_sp\_record\_deal\_tmp记录新增的轨迹。（1）如新增轨迹的status字段<>1、suspicious\_leave<>1、2、3则轨迹不参与计算。
2. 以城市为单位使用密度聚类算法将新增轨迹点与现有热点区域的轨迹点进行聚类，先统计新增轨迹点与现有热点区域中的轨迹点的距离小于300米的所有轨迹点。
3. 如果步骤2的结果中存在某个热点区域的轨迹点数大于阈值，则判断该新增轨迹点属于该热点区域的核心点
4. 如果不满足步骤3，步骤2的结果中存在一个核心点，则判断该新增轨迹点属于该核心点对应的热点区域的边界点
5. 如果不满足步骤3、4，步骤2的结果中噪声点的数目大于阈值，则判断该新增轨迹点与这些噪声点新生成一个热点区域，该新增轨迹点为新增热点区域中的一个核心点。
6. 如果不满足步骤3、4、5，该新增点就是个噪声点。
7. 如果新增热点区域，统计当前区域内所有轨迹点的经纬度坐标的平均值，以平均值作为中心点代表本区域，区域信息保存入dim\_sp\_area表，需要更新batch\_no字段，每个批次的区域不一样
8. 如果新增轨迹点加入现有热点区域中，更新ods\_sp\_record\_deal表中的batch\_no、area\_id字段值，标识出该新增轨迹点所属的区域。
9. 将该新增点的状态存入ods\_dbscan\_result.
10. 最新的区域城市、编号、中心经纬度保存入Redis，分成HMap和Sorted-Set两种格式。HMap格式为：<城市名,< 区域编号,经度\_维度>)，如<南京市,<201，21.45178911\_32.98112822>>。Sorted-Set格式见下方第二个表。
11. Redis表设计

|  |  |
| --- | --- |
| 功能 | 保存热点区域 |
| 类型 | Hash |
| Key | 城市名 |
| Field | 区域编号 |
| Value | 经度\_维度 |
| 前缀 | sp\_hot\_area |
| 示例 | sp\_hot\_area:南京市,<batch\_no,20170109>,<10001, 21.45178911\_32.98112822>,<10002, 21.45178911\_32.98112822> |

**第一个field是批次号，如：**<batch\_no,20170109>

|  |  |
| --- | --- |
| 功能 | 按中心点经纬度顺序保存热点区域 |
| 类型 | Sort-Sets |
| Key | 城市名\_Lat（维度）, 城市名\_lon（经度） |
| Field | 区域编号 |
| Value | 经度或维度的值 |
| 前缀 | 无 |
| 示例 | 南京市\_lat,<10001, 21.45178911>,<10002, 32.98112822>  南京市\_lon,<10001, 21.45178911>,<10002, 32.98112822> |

### 推荐路线生成

1. **每月生成路线**
2. 从ods\_sp\_record\_deal表中分城市分区域获取近3个月（首次为2016年12月-2017年3月）每个区域中所有运动记录的起点、终点，随机取1个起点或终点计算与其他起点、终点的距离，如果小于阀值（暂定100米），则将这些起点或终点聚为一类。如此反复，将所有起点聚类。如轨迹的status字段<>1、suspicious\_leave<>1、2、3则轨迹不参与计算。
3. 聚为一类的起点中取里程最短的一条运动记录，取上面所有的轨迹点，与相近起点的其他运动记录，按照轨迹点的顺序计算欧式距离的平均值，如果距离小于阀值（暂定500米），则认为这2个运动记录相近归为一类。如此反复，将相近起点的运动记录归为几类。如果第一个点为终点，则要按照反序两两计算轨迹点之间的距离。
4. 每类相近运动记录中取里程最长的运动记录作为推荐路书，以此找到每个起点对应的1-n个推荐路书（n<=起点个数）。推荐路书结果保存入ods\_sp\_hot\_record表，该表每次全量更新（merge into形式）。
5. 更新ods\_sp\_record\_deal表中的recommand\_id字段，指定每个运动记录对应的推荐记录，两者应该属于同一城市、同一区域、起点相近、轨迹点综合距离相近。
6. 将推荐路书的城市、对应区域、起点经纬度、路书编号保存入Redis。HMap格式为：<城市名\_区域编号,<路书编号,经度\_维度>>，如<南京市\_201,<234567，21.45178911\_32.98112822>>。
7. 将推荐路书的城市、对应区域、路书编号、起点名称、终点名称、路书的各个标签、轨迹点序号、轨迹点的经度和维度保存入Redis，格式为Sorted-Sets。
8. Redis表设计

|  |  |
| --- | --- |
| 功能 | 保存推荐路书 |
| 类型 | Hash |
| Key | 城市名\_区域编号 |
| Field | 路书编号 |
| Value | 经度\_维度 |
| 前缀 | sp\_hot\_record |
| 示例 | sp\_hot\_record:南京市\_10001,<234567, 21.45178911\_32.98112822>,<234568, 21.45178911\_32.98112822> |

|  |  |
| --- | --- |
| 功能 | 保存推荐路书轨迹点 |
| 类型 | Sort-Sets |
| Key | 城市名\_区域编号\_Lat（维度）, 城市名\_区域编号\_lon（经度） |
| Field | 路书编号\_起点名称\_终点名称\_里程\_耗时\_时间\_人数\_配速\_热量\_爬坡值\_轨迹点序号 |
| Value | 经度或维度的值 |
| 前缀 | 无 |
| 示例 | 南京市\_201\_lat,<10001\_1, 21.45178911>,<10001\_2, 32.98112822>  南京市\_201\_lon,<10001\_1, 21.45178911>,<10001\_2, 32.98112822> |

**二、实时生成路线**

1. 按城市将新增的轨迹点进行划分，临时轨迹点表ods\_sp\_sport\_point\_hb\_tmp记录新增的轨迹对应的轨迹点，条件STATUS =1
2. 以热点区域为单位，取当前热点区域中轨迹点的栅格信息，计算当前热点区域新增轨迹轨迹点的栅格信息，根据2.7中的步骤计算新增轨迹在同一个热点区域下与其他轨迹的拟合情况，剩下的轨迹即为当前区域对外的推荐路数
3. 如果新增的轨迹将其他轨迹拟合，将新增轨迹保存入ods\_sp\_hot\_record表，更新runner\_value,更新值等于被拟合轨迹的runner\_value。更新runner\_time，更新值等于被拟合轨迹的runner\_value\*runner\_time再求和除以runner\_value的更新值
4. 如果新增的轨迹被其他轨迹拟合，更新拟合新增轨迹的推荐路线的runner\_value=原来runner\_value+1
5. 更新ods\_sp\_record\_deal表中的recommand\_id字段，指定每个运动记录对应的推荐记录，两者应该属于同一城市、同一区域、起点相近、轨迹点综合距离相近。
6. 将推荐路书的城市、对应区域、起点经纬度、路书编号保存入Redis。HMap格式为：<城市名\_区域编号,<路书编号,经度\_维度>>，如<南京市\_201,<234567，21.45178911\_32.98112822>>。
7. 将推荐路书的城市、对应区域、路书编号、起点名称、终点名称、路书的各个标签、轨迹点序号、轨迹点的经度和维度保存入Redis，格式为Sorted-Sets。
8. Redis表设计

|  |  |
| --- | --- |
| 功能 | 保存推荐路书 |
| 类型 | Hash |
| Key | 城市名\_区域编号 |
| Field | 路书编号 |
| Value | 经度\_维度 |
| 前缀 | sp\_hot\_record |
| 示例 | sp\_hot\_record:南京市\_10001,<234567, 21.45178911\_32.98112822>,<234568, 21.45178911\_32.98112822> |

|  |  |
| --- | --- |
| 功能 | 保存推荐路书轨迹点 |
| 类型 | Sort-Sets |
| Key | 城市名\_区域编号\_Lat（维度）, 城市名\_区域编号\_lon（经度） |
| Field | 路书编号\_起点名称\_终点名称\_里程\_耗时\_时间\_人数\_配速\_热量\_爬坡值\_轨迹点序号 |
| Value | 经度或维度的值 |
| 前缀 | 无 |
| 示例 | 南京市\_201\_lat,<10001\_1, 21.45178911>,<10001\_2, 32.98112822>  南京市\_201\_lon,<10001\_1, 21.45178911>,<10001\_2, 32.98112822> |

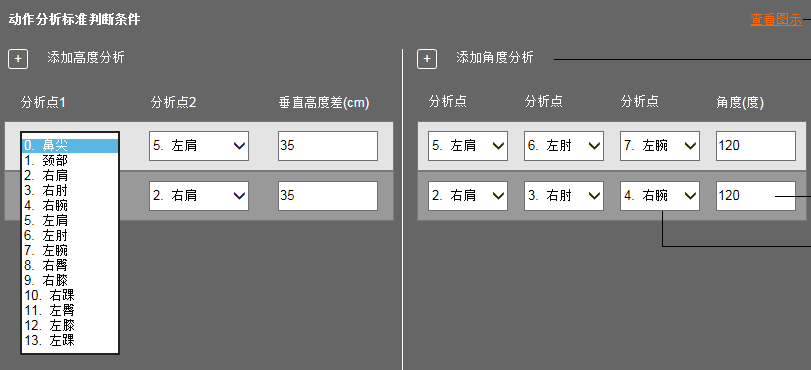
### 推荐路线查询

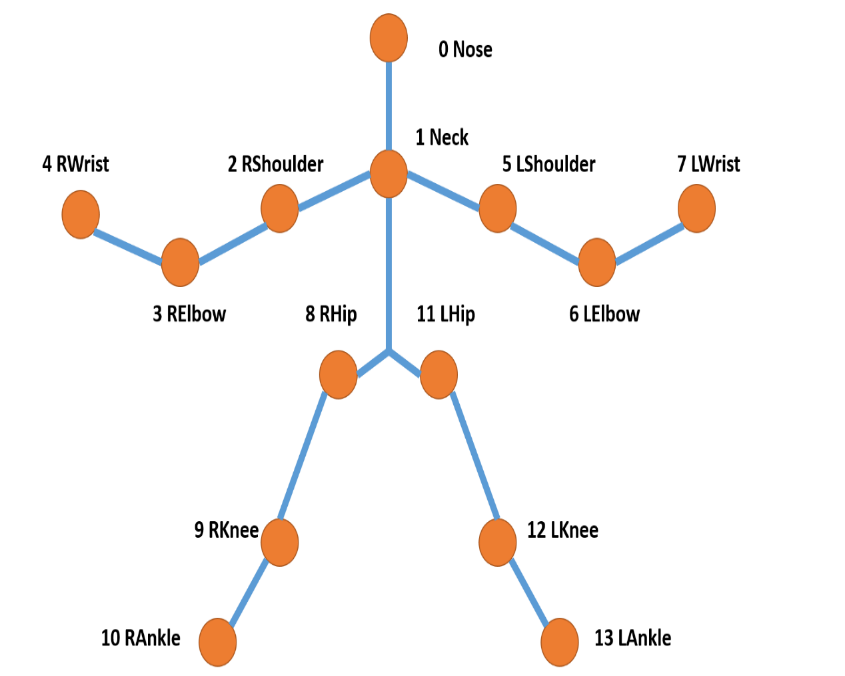
1. 接口形式：Http，集成入安全中心
2. 善跑输入：用户所在位置（经度、维度、不能为空）、用户所在城市中文名称（可为空）
3. 平台输出：路书对应的运动记录编号和对应标签，最近的轨迹点序号、经度、维度，需按照由近及远排序
4. 内部逻辑：
5. 如城市名称为空，则根据经纬度调用GIS接口查询用户所在城市，不为空则跳过。
6. 找出用户所在城市所有区域列表，以用户所在点为中心，画一个指定半径的正方形区域（半径暂定5000米，约合0.045度），找出落在正方形内的区域中心点，对应区域作为候选区域。并计算正方形内区域中心点与用户所在位置之间的距离，按从小到大排序。如果正方形内没有区域，则流程结束，返回空值。
7. 取最近一个区域内落在正方形内的所有轨迹点，并计算所有轨迹点与用户所在位置之间的距离，并按从小到大排序。依次判断轨迹点所归属的跑步路线，属于同样轨迹点的跑步路线在首次记录后忽略调过，直到所有轨迹点判断完毕。
8. 如果该区域在正方形内没有轨迹点，则由近及远找下一个区域。如果该区域内找到的跑步路线大于规定值（暂定30），则流程结束，否则也由近及远找下一个区域。在下一个区域内和以前区域中累计找到的跑步路线大于规定值（暂定30），则流程结束。
9. 向善跑服务端返回推荐路线编号、路线起点中文名、路线终点中文名、路线的里程、耗时、时间、人数、配速、热量、爬坡值、用户ID（固定1）、海拔落差值（暂定0）以及最近轨迹点的序号、经度、纬度，JSON字符串格式，格式含义如下：。

## 咪咕善跑配套

### 动作指标管理

咪咕善跑管理后台提供全量的人体部位关节点选项，全部13个点位如下图所示。动作指标管理人员根据每个动作项的指标差异，选择不同关节点，定义针对该动作的名称、对应关节点、关键姿态。

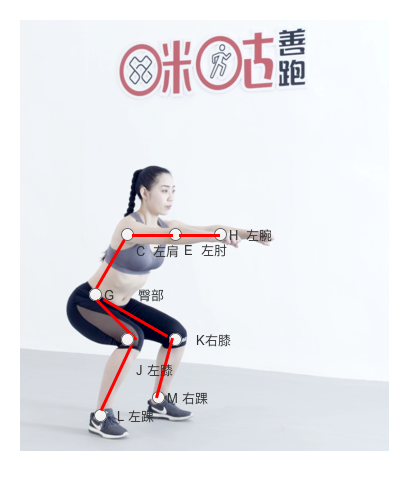




每个动作的关键指标，分高度、角度两类，每类各支持添加10个指标，即每个动作最多可定义20个关键指标，指标分成距离、角度两类：

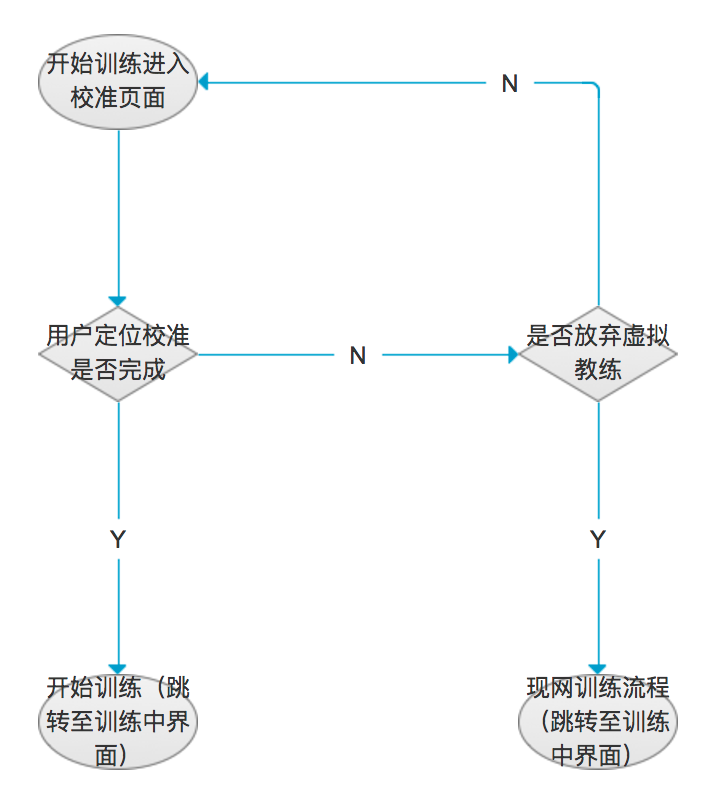
1. 距离类指标：可选择两个点位，分别捕捉两个点位到地面的垂直距离，并根据点位顺序，计算出两者之间的距离差。即距离=分析点1到地面的垂直高度-分析点2到地面的垂直高度。
2. 角度类指标：可选择三个点位，根据点位顺序，以中间的点位，作为角度的中心，即选择点位为A、B、C，定义的指标即为角ABC，选择点位为A、C 、B、，定义的指标即为角ACB。

以缓冲深蹲动作为例：

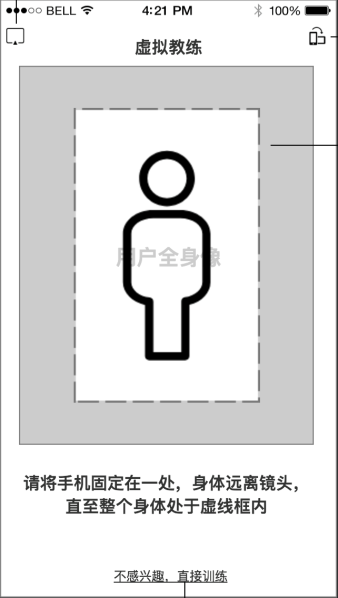
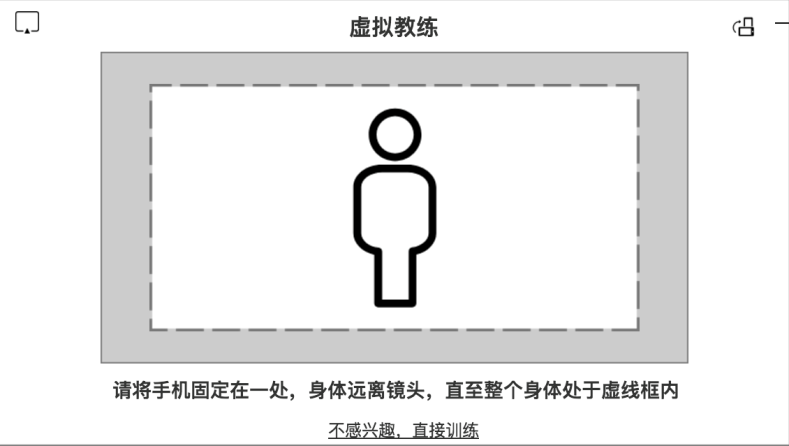


运营人员在咪咕善跑管理后台该动作的指标管理页面，选择臀部点位G，左膝点位J，左踝点位L，分别组成线段GJ与线段JL，再将线段GJ与线段JL组成角度GJL，角度GJL即为该缓冲深蹲动作的一个衡量指标。以此类推，可定义出其他角度、高度等指标。

### 训练开始准备



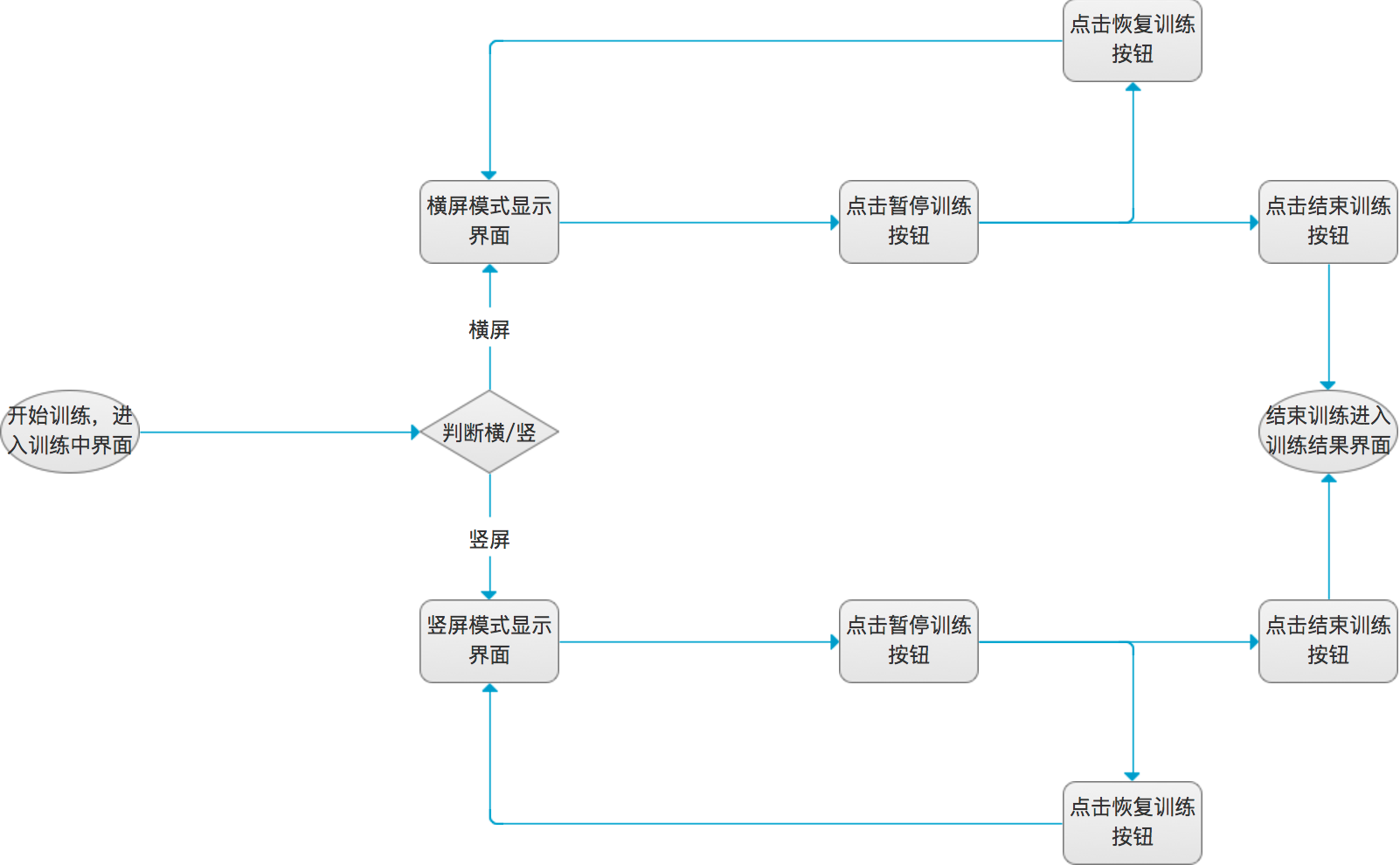
善跑用户在善跑客户端的训练详情界面上下载训练视频后，点击“开始训练”按钮进入训练准备界面。

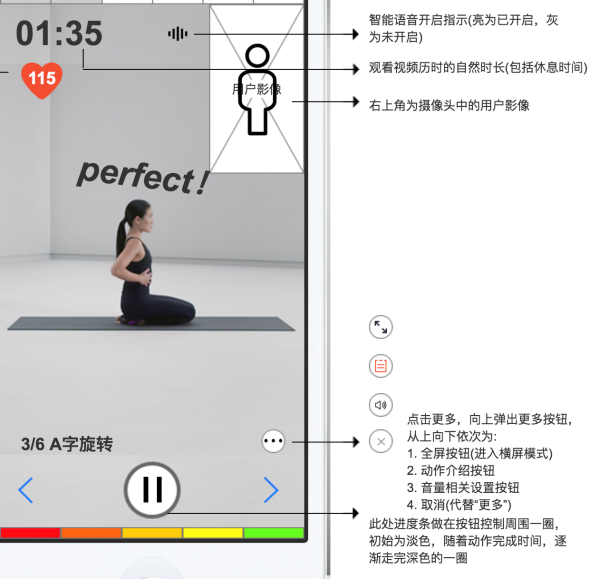
**准备训练界面包含元素和规则如下：**

1. 准备训练界面包含元素：返回按钮、取景器、前后摄像头切换按钮、定位提示、使用帮助、跳过准备、投屏引导；
2. 取景器：默认通过前置摄像头取景，在取景器中提供用户定位区域提示，在定位成功后结束准备训练界面，进入训练中界面；
3. 跳过准备：用户放弃使用虚拟教练功能，进入常规训练流程。

### 训练过程提示



完成准备训练中的定位环节后自动进入训练中界面：



**一、训练中界面元素与规则**

1. 训练中界面元素包含：训练指导视频、训练进度、取景器、能量槽、视频播控、语音控制提示，心率数据、动作完成状态；
2. 能量槽：根据目前完成的所有动作的匹配度（百分比）取平均值，并通过能量槽的形式体现；
3. 视频播控：视频播控包括暂停、上一个动作、下一个动作，所有视频播控功能需支持语音控制；
4. 动作评判状态：通过perfect、good、common、bad、miss、combo（状态名称暂定）等状态表现用户动作完成情况；
5. 界面需支持横、竖屏两种显示模式，两种模式界面元素一致。

**二、时间性能要求**

1. 每一帧图片的机器视觉对比结论（包括动作评分、评分等级等），需在该帧动作完成后准实时反馈到客户端并展现。
2. 每一套动作的机器视觉人体标签、机器视觉对比结论（包括动作评分、评分等级等），需在该套动作完成后1s内，反馈到客户端并展现。
3. 每一组教程（包含多套训练动作）的机器视觉对比结论（包括训练评分、评分等级等），需在该组教程完成后3s内，反馈到客户端并展现。

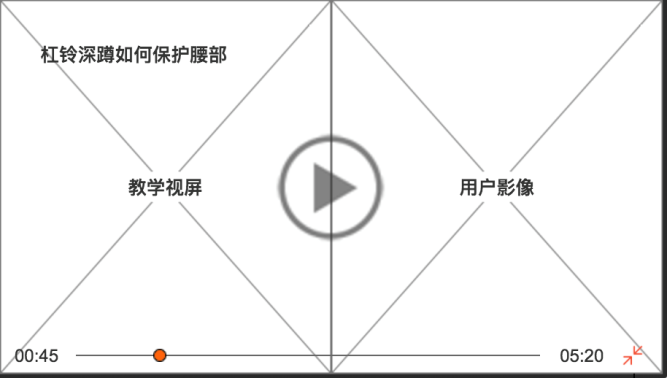
**三、动作评判标准**

动作偏差百分比加权平均值，具体含义如下：

1. 单帧动作偏差百分比=\*100%
2. 动作偏差百分比加权平均值=，初期权值均设为1
3. 教程偏差百分比加权平均值=，初期权值均设为1

### 训练结果回放

当用户完成训练或主动结束训练，进入训练结束界面：



**一、训练结束界面元素与规则**

1. 训练结束界面包含以下元素：训练封面图片、训练名称、训练结束时间、完成动作、训练时长、消耗卡路里、动作分析（动作列表）、综合评价、分享；
2. 消耗卡路里：根据动作完成契合度调整消耗卡路里，如：徒手深蹲动作预计消耗卡路里50，动作契合度80%，则卡路里消耗调整为40；
3. 动作分析：动作分析通过动作列表呈现，动作列表包含动作名称、契合度、用时、回看。点击回看后播放用户在进行该动作过程中的视频录像；
4. 综合评价：根据此次训练的综合契合度（所有训练动作的契合度的平均值）给予用户对应评价，包含：完美的训练、标准的训练、及格的训练、马马虎虎的训练（暂定）。

**二、动作标准度评判规则**

1. 将每套动作全过程中的每一帧图像，与教练标准动作图像进行对比，计算出该动作的偏差百分比，再求出该套动作中全部动作的偏差百分比加权平均值。
2. 根据加权平均值的范围，给出该动作的评判结论。

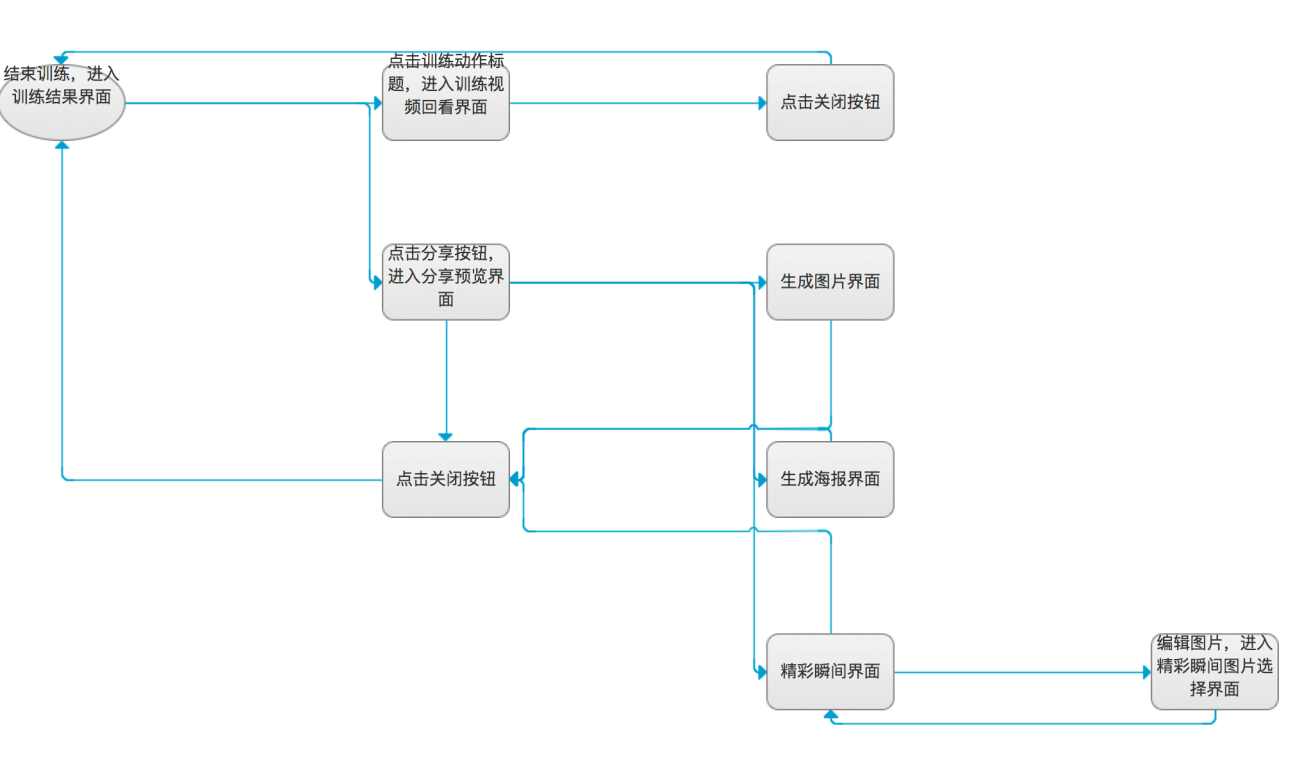
|  |  |
| --- | --- |
| 偏差百分比加权平均值范围 | 评判结论 |
| 90%-100% | perfect |
| 70%-90% | good |
| 50%-70% | common |
| 30%-50% | bad |
| 0-30% | miss |

**三、教程标准度评判规则**

1. 将每一组教程全过程中的每一套动作的评判结论，即每一套动作的偏差百分比加权平均值，再取加权平均值，得到该组教程的偏差百分比加权平均值。

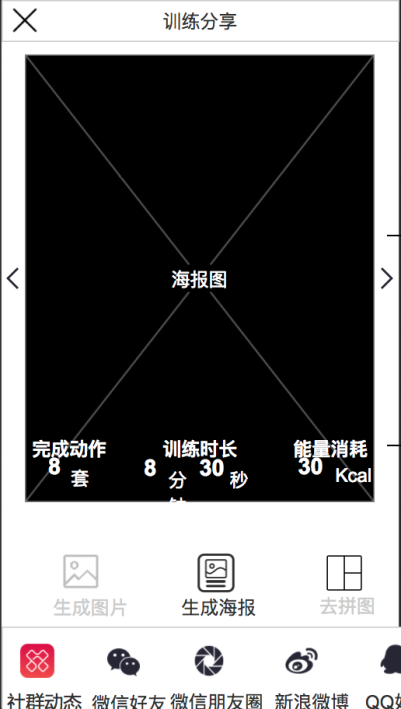
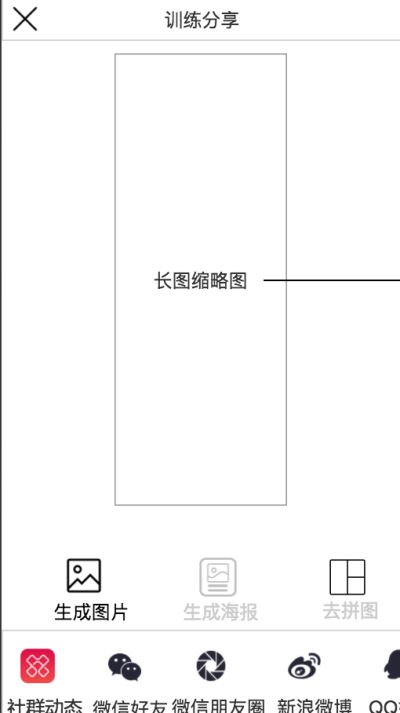
|  |  |
| --- | --- |
| 偏差百分比均值范围 | 评判结论 |
| 90%-100% | 健身王者 |
| 70%-90% | 健身达人 |
| 50%-70% | 健身学徒 |
| 30%-50% | 健身小白 |
| 0-30% | 健身菜鸟 |

### 训练结果分享



用户点击分享提供长图、海报、拼图三种分享样式；

* 1. 长图：训练结束页面的截图；
  2. 海报：后台可配置海报图片与匹配的海报水印，海报水印包含：训练名称、训练时长、消耗卡路里等参数；
  3. 拼图：拼接多张图片，用户可选图片内容，默认内容包含训练封面图片、用户训练中照片（取用户在训练过程中的三张照片，获取规则为完成动作匹配度最高的前三个动作时的照片）。



### 跑步路线推荐

**一、路线首页**



客户端侧，【推荐路书】区域数据显示如下：

1. 客户端调用善跑服务端的大数据推荐接口时，需传递用户当前所在的经纬度、省市名称参数；
2. 推荐路书区域由原来的3条数据，改为只显示2条数据：第一条是后台设置的推荐路线数据随机取一条；第二条是大数据推荐的数据中的第一条。
3. 路线的显示内容跟之前一样，包括运动方式、里程数、轨迹截图、发布人头像、路线名称。点击路线内容后可跳转到路线详情页面，与一期的功能相同。

**二、路线详情页**



客户端界面显示如下：

1. 【精品】tab页：显示后台配置的“类型”为“官方推荐”的路线；排序后台可配置；支持分页，每页最多加载10条；
2. 【附近】tab页：显示大数据推荐的内容；推荐路线按距离用户由近至远排列；支持分页，每页10条数据。

## 系统接口

### 数据上传接口

* 功能描述：将训练样本视频及配套标注、视频元数据说明文件由AI前置服务平台上传至AI云平台，上传结果需要记录日志。遵循《咪咕文化AI能力数据规范》的相关要求，详见第3.1.5节。
* 接口方式：FTP，使用现有的大数据平台与各子公司之间的数据传输通道。
* 传输数据：视频源文件、标注文件、数据信息文件、数据校验文件

### 模型发布接口

* 功能描述：将训练生成的tensorflow或caffe模型文件、模型元数据说明文件从AI云平台下载至AI前置服务平台，在AI前置服务平台上发布模型，解析并保存模型日志，下载和发布模型需要记录日志，详见第3.1.7节。
* 接口方式：FTP，使用现有的大数据平台与各子公司之间的数据传输通道。
* 传输数据：模型文件、模型日志说明文件

### 模型在线使用接口

* 功能描述：训练过程中，客户端采集每一帧训练动作图像，同时客户端调用AI能力，分析图像中的动作指标与标准动作的差值。根据差值评价动作状态：通过perfect、good、common、bad、miss、combo（状态名称暂定）等状态表现用户动作完成情况。
* 接口方式：HTTP，RESTFUL
* 传输数据：参见第3.2.2节

### 模型离线使用接口

* 功能描述：当用户完成训练或主动结束训练，对动作做完整分析和综合评价。对比视频中，用户动作画面显示人体的关节信息；根据此次训练的综合契合度（所有训练动作的契合度的平均值）给予用户对应评价，包含：完美的训练、标准的训练、及格的训练、马马虎虎的训练（暂定）。最终输出标注后的新视频地址、契合度最高的3张图像地址。
* 接口方式：HTTP，RESTFUL
* 传输数据：参见第3.2.3节

### 截图开始通知接口

* 功能描述：视讯服务端通知互娱大数据平台直播视频截图开始
* 接口方式：HTTP
* 传输数据：输入userid（userid\_actionid）、channelid、时间戳、咪咕云目录地址，返回是否处理成功，新开发

### 截图结束通知接口

* 功能描述：视讯服务端通知互娱大数据平台直播视频截图结束
* 接口方式：HTTP
* 传输数据：输入userid（userid\_actionid）、channelid，返回是否处理成功，新开发

### 直播状态通知接口

* 功能描述：视讯服务端通知互娱大数据平台直播状态为开始、结束、暂停
* 接口方式：HTTP
* 传输数据：输入channelid、直播状态，返回是否处理成功，参见咪咕直播标准API接口规范

### 录像完成通知接口

* 功能描述：视讯服务端通知互娱大数据平台直播录像的存放地址
* 接口方式：HTTP
* 传输数据：输入channelid、录像地址，返回是否处理成功，参见咪咕直播标准API接口规范

### 关闭直播接口

* 功能描述：互娱大数据平台通知视讯服务端关闭直播流
* 接口方式：HTTP
* 传输数据：输入channelid，返回是否处理成功，参见咪咕直播标准API接口规范

### 删除截图接口

* 功能描述：互娱大数据平台通知视讯服务端删除咪咕云上的截图目录及截图
* 接口方式：HTTP
* 传输数据：输入咪咕云上截图目录的路径，返回是否处理成功，新开发

# 系统响应码

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **场景** | **响应码** | **说明** |
|  | 调用接口 | 0000 | 返回成功 |
|  | 调用接口 | 0001 | 输入参数为空 |
|  | 调用接口 | 0002 | 输入参数格式错误 |
|  | 调用接口 | 0003 | 输入参数类型错误 |
|  | 调用接口 | 0004 | 输入文件路径不存在 |
|  | 调用接口 | 0005 | 模型错误 |
|  | 调用接口 | 0006 | 图片格式异常 |
|  | 调用接口 | 0007 | 视频格式异常 |
|  | 调用接口 | 0008 | 处理视频异常 |
|  | 调用接口 | 0009 | 输出图片异常 |
|  | 调用接口 | 0010 | 输出视频异常 |
|  | 调用接口 | 0011 | 输出推荐路线异常 |
|  | 调用接口 | 0012 | 请求地址无响应 |