

# HIIT动作识别 (Human Activity Recognition)

## 一、简介

本项目为作者大二第二学期实验设计课程的期末项目。

项目可分为两个板块：**前端（小程序）和后台（服务器）**。本项目通过**采集数据、数据处理、特征提取、训练机器学习模型、服务器部署算法**等过程，最终可实现用户手持手机做HIIT规范动作，微信小程序实时检测**动作类别并记录动作个数**的功能。

## 二、动作规范

- 本项目选取HIIT系列中的四个动作作为识别对象（**动作规范请参考HIIT健身系列动作**），分别为：
  - 徒手侧平举
  - 前后交叉小跑
  - 开合跳
  - 半蹲
- 为进一步规范动作，我们要求使用者左手持握手机，具体姿势可见作者小程序（HIIT动作识别）首页。因此在采集数据和动作识别的过程中，我们都严格要求使用者的动作标准。**如在使用时因动作不规范或持握方式错误等问题而出现识别误差，也属于正常情况。**

## 三、数据集

- 本数据集由作者以及其同学利用自己开发的采集数据小程序共同收集、共享使用。
- 每类动作约有30-40组不同数据，分别来自不同的测试者。
- 每组数据为手机**20s**中采取的六轴数据点。我们设置**IOS系统手机**的采样周期为**20ms**，**每组数据约有1000个数据点**；设置**Android系统手机**的采样周期为**60ms**，**每组数据约有333个数据点**。
- 通过微信小程序的数据采集功能，将测试者做动作时的六轴原始数据从微信小程序端导入电脑，得到对应JSON文件。
- 数据集分为**IOS系统数据**和**Android系统数据**，考虑到不同系统手机在采样时的频率差异，我们将不同类型的数据分开存放，方便后续数据处理和训练模型。

## 四、具体实现

### A：小程序端（用于实际使用）

#### 第一部分：采集数据功能（代码请见**动作识别小程序**）

- 数据采集：利用微信小程序自带的传感器调用函数获取手机内部六轴传感器的原始数据，并保存至云数据库。

- 数据导出：将云数据库数据以JSON格式导出至电脑本地，以便利用Python对其进行进一步操作。

## 第二部分：识别动作功能（代码请见[动作识别小程序](#)）

- 动作识别（60s内）：通过小程序与云服务器通信，定时将用户运动时产生的六轴数据发送至服务器，并投放至存储在服务器上的机器学习算法模型中进行动作识别，最后服务器返回识别结果至小程序端，并以页面展示和语音播报两种方式呈现，以达到实时检测的效果。
- 动作计数（60s内）：通过小程序与云服务器通信，在用户完成运动时，将运动期间存储的六轴数据发至服务器，并投放至[波峰检测](#)代码中，根据波峰和动作的对应关系计算出每种动作的完成个数，并返回至小程序端进行页面显示。

## B：电脑本地（用于数据处理、训练模型）

### 第一部分：数据处理（代码说明请见[Data\\_Preprocessed.ipynb](#)）

本部分主要用于对小程序采集的六轴原始数据进行特征提取，并将处理后的特征值和目标值通过TXT文件导出，以便于训练模型。

本部分主要包括：

- 滤波：对原始数据用三阶低通滤波器滤波
- FFT：对原始数据快速傅里叶变换
- 分割数据：对原始数据进行区间分割（将每组20s长的数据 分割成多个2.56s长的小区间数据。为减小误差，分割区间时要求前后两区间具有50%重叠区，因此20s数据可以最多被分成 13 个2.56s的小区间）
- 特征提取：对上述每个小区间进行特征提取
- 保存特征：将提取特征后的特征数据进行存储，以便投入机器学习算法模型训练。

### 第二部分：训练模型（代码说明请见[Model\\_Train.ipynb](#)）

本部分使用sklearn库调用不同模型，通过将第一部分数据预处理得到的特征数据集投入模型、优化参数、比较模型准确率，最终选择出效果较好的模型存至本地。优化参数和比较模型过程未在代码中体现，读者可以利用sklearn库中的相关评估函数对其他模型进行比较。作者发现随机森林和KNN模型效果较为优秀，最终选择使用[随机森林模型](#)。

本部分主要包括：

- 调用模型
- K-Fold交叉验证：利用k-fold方法比较模型的准确率，此方法相较于按比例划分训练集和测试集的方法更能避免偶然性。
- 优化参数
- 选取模型

### 第三部分：部署服务器运行代码（代码说明请见[Main\\_ActionRecognition.ipynb](#)）

本部分采用flask框架部署至服务器，作用为：

- 动作识别：用户在小程序端运动时，实时接收小程序实时发送的六轴原始数据，并对其进行数据预处理和特征提取等操作（同第一部分），并将处理好的特征数据投入预先训练好的模型中（由第二部分），得到模型分类结果，最后将模型分类结果返回到小程序端。

- 动作计数：用户在小程序端结束运动后，接收小程序发送的运动期间的六轴数据，并对其进行波峰检测，根据波峰和动作的对应关系，分析并计算出用户做的动作数。

本部分依然对IOS和安卓进行差别处理，具体方法见代码注释。

## C：服务器端（用于实际使用）

部署完成后运行上述第三部分python文件即可。

## 五、使用流程（针对开发者）

---

1. 采集数据：微信小程序采集数据功能（云数据库导出至电脑本地）
2. 数据处理：feature\_IOS(Android)\_Preprocessed.py
3. 训练模型：model\_Both\_Train.py
4. 部署服务器：main\_Both\_5Actions.py（上传至服务器运行）
5. 动作识别：微信小程序动作识别功能

## 六、结语

---

本项目暂仅用于课程考核。仍存在诸多不足，请批评指正。如有问题或建议反馈，请联系QQ：986927662。——山东大学（威海）18数据科学 许函嘉