

训练数据集介绍

本数据集包含二进制格式的信号以及对应的Json格式标签文件。数据集包含以下两个主要部分：

1. 二进制信号数据：

- **描述：**包含7500条iq采样数据，表示在2400 MHz到2500 MHz范围内的某一区间段采集的信号。每个数据文件的时长为20 ms到150 ms。数据文件记录的时段和频段内存在多个信号（1到8个）。这些信号可能在时域和频域上存在重叠，也就是说存在信号之间的干扰。文件后缀名为 .bin。
- **采样率：**数据的采样率可为5 Ms/s、20 Ms/s、30 Ms/s、40 Ms/s、50 Ms/s和80 Ms/s。采样率与观测频宽相等。
- **时长：**数据文件的时长可为20 ms、40 ms、60 ms、80 ms、100 ms和150 ms。
- **信号特征：**一条数据内包含多个信号（1、2、3、4、6、8个信号），观测区间能够涵盖信号完整的频率成分。信号个数小于4个时，文件时长为20 ms。信号包括Wi-Fi、BLE、ZigBee、LoRa、单载波、FM、AM以及其不同制式的类型共14种。部分信号的频宽很窄，准确识别较为困难，需要注意。
- **数据格式：**每个数据文件为16位浮点数格式存储的I（实部）与Q（虚部）交错序列。文件大小不一，具体取决于采样率和采样时长。示例读取代码如下：

```
import numpy as np
file_path = 'path/to/your/file.bin'
signal = np.fromfile(file_path, dtype=np.float16)
signal = signal[::2] + 1j * signal[1::2]
```

2. 标签数据：

- **描述：**Json格式的标签文件。标签文件与二进制信号数据文件一一对应，例如 42.bin 与 42.json 文件对应。标签文件包含每个信号的信号类型、占用频段、占用时间段信息，并且还记录了数据文件的观测频率区间信息。数据文件的采样率等于观测频宽，观测时长未直接给出，但可通过信号长度和采样率计算得到。
- **数据格式：**Json格式文件，包含以下字段：
 - **signals：**信号列表，每个信号包含以下字段：
 - `signal_id`：信号ID（整数）。
 - `start_frequency`：信号起始频率（浮点数，单位MHz）。
 - `end_frequency`：信号结束频率（浮点数，单位MHz）。
 - `start_time`：信号起始时间（浮点数，单位毫秒）。
 - `end_time`：信号结束时间（浮点数，单位毫秒）。
 - `class`：信号类型（整数，[0, 13]之间的类别编号）。
 - **observation_range：**观测频率区间（列表，包含两个浮点数，单位MHz），表示数据文件的观测频率范围。
- **数据示例：**标签文件示例内容如下：

```
{
  "signals": [
    {
      "signal_id": 0,
      "start_frequency": 2411.9,
      "end_frequency": 2412.1,
      "start_time": 0.0,
      "end_time": 20.0,
      "class": 12
    },
    {
      "signal_id": 1,
      "start_frequency": 2432.0,
      "end_frequency": 2452.0,
      "start_time": 10.0,
      "end_time": 15.0,
      "class": 0
    },
    {
      "signal_id": 2,
      "start_frequency": 2422.0,
      "end_frequency": 2442.0,
      "start_time": 5.0,
      "end_time": 15.0,
      "class": 3
    }
  ],
  "observation_range": [
    2408.0,
    2458.0
  ]
}
```

测试数据集介绍

与训练数据集类似，数据集包含二进制格式的信号以及对应的记录观测频率区间的Json格式文件。数据集包含以下两个主要部分：

1. 二进制信号数据（格式与训练数据集相同）：

- 描述：**包含1000条iq采样数据，表示在2400 MHz到2500 MHz范围内的某一区间段采集的信号。每个数据文件的时长为20 ms到150 ms。数据文件记录的时段和频段内存在多个信号（6到10个）。这些信号可能在时域和频域上存在重叠，也就是说存在信号之间的干扰。文件后缀名为 .bin。
- 采样率：**数据的采样率可为20 Ms/s、30 Ms/s、40 Ms/s、50 Ms/s和80 Ms/s。采样率与观测频宽相等。
- 时长：**数据文件的时长可为20 ms、40 ms、60 ms、80 ms、100 ms和150 ms。
- 信号特征：**一条数据内包含多个信号（6到10个，比训练集更密），观测区间能够涵盖信号完整的频率成分，不过时间上可能不完整。
- 数据格式：**每个数据文件为16位浮点数格式存储的I（实部）与Q（虚部）交错序列。文件大小不一，具体取决于采样率和采样时长。

2. 观测区间数据：

- 描述：**Json格式的文件，与二进制信号数据文件一一对应，例如 42.bin 与 42.json 文件对应。文件仅包含数据文件的观测频率区间信息。数据文件的采样率等于观测频宽，观测时长未直接给出，但可通过信号长度和采样率计算得到。
- 数据格式：**Json格式文件，包含以下字段：
 - **observation_range：** 观测频率区间（列表，包含两个浮点数，单位MHz），表示数据文件的观测频率范围。
- 数据示例：**标签文件示例内容如下：

```
{  
  "observation_range": [  
    2408.0,  
    2458.0  
  ]  
}
```

测试数据集分为public和private两部分。比赛期间，public测试集对选手可见，选手可提交预测结果文件（predictions.json）至系统评分，作为模型性能参考。另外还有一份选手不可见的private测试集，比赛截止后，我们将在private测试集上运行选手提交的推理代码，所得分数作为实际得分。

输出格式与评分

输出格式

选手需输出名为 predictions.json 的文件，包含选手模型的预测结果。文件格式如下：

- **格式结构：**每个键对应一个测试样本的预测结果，键为样本的 id （文件名），值为该样本的预测信息。值与训练数据集的标签格式类似，每个信号额外输出 confidence 字段（范围在[0, 1.0]，非必须项，若不提供则默认为1.0，不过可能会影响得分的mAP计算）。也就是说，预测结果为训练标签格式形式增加 confidence 字段后，再嵌套一层样本 id 。
- **示例文件：**

```
{  
  "0": {  
    "signals": [  
      {  
        "signal_id": 0,  
        "start_frequency": 2411.9,  
        "end_frequency": 2412.1,  
        "start_time": 0.0,  
        "end_time": 20.0,  
        "class": 12,  
        "confidence": 0.9  
      },  
      {  
        "signal_id": 1,  
        "start_frequency": 2416.5,  
        "end_frequency": 2417.5,  
        "start_time": 0.0,  
        "end_time": 20.0,  
        "class": 9,  
        "confidence": 0.8  
      },  
    ],  
    "observation_range": [  
      2408.0,  
      2458.0  
    ]  
  },  
  "1": {  
    "signals": [  
      {  
        "signal_id": 0,  
        "start_frequency": 2464.5,  
        "end_frequency": 2465.5,  
        "start_time": 102.0,  
        "end_time": 150,  
        "class": 9,  
        "confidence": 0.9  
      },  
      {  
        "signal_id": 1,  
        "start_frequency": 2445.5,  
        "end_frequency": 2446.5,  
        "start_time": 73.0,  
        "end_time": 86.0,  
        "class": 6,  
        "confidence": 0.85  
      },  
    ],  
    "observation_range": [  
      2426.0,  
      2476.0  
    ]  
  },  
}
```

评分规则

本任务采用 **mAP@0.50:0.95** 作为主评估指标，具体评分机制如下：

- 对每个样本，使用 **mAP@0.50:0.95** (IoU 阈值从 0.50 到 0.95，步长为 0.05，共 10 个) 计算评分，取这 10 个 AP 的平均值。
- 每个样本的评分范围为 [0.0, 1.0]，越高表示预测越准确。

1. 定义时频 IoU (Intersection over Union):

IoU 是衡量预测信号框 (P) 与真实信号框 (GT) 重叠程度的指标。对于二维矩形，IoU 定义为交集面积除以并集面积。

$$IoU = \frac{Area(P \cap GT)}{Area(P \cup GT)}$$

其中，`Area` 是指时频平面上的矩形面积：`(end_frequency - start_frequency) * (end_time - start_time)`。

2. 设定 IoU 阈值：

- **多阈值平均 (mAP@0.5:0.95)**: 在多个 IoU 阈值 (0.50, 0.55, ..., 0.95) 下计算 AP，然后取平均。

3. 计算步骤 (按每个类别和总体的 mAP):

对于每条的观测数据：

• 步骤 3.1: 匹配预测框和真实框

- 将预测框按照 confidence 分数从高到低排序。
- 遍历每个预测框。对于一个预测框 P_i ：
 - 找到所有与 P_i 类别相同且 IoU 超过 IoU 阈值的真实框 GT_j 。
 - 如果找到多个 GT_j ，选择与 P_i 的 IoU 最大的那个 GT_j 进行匹配。
 - **去重**: 一个真实框 GT_j 只能被一个预测框 P_i 匹配一次。如果 GT_j 已经被别的预测框匹配过，则 P_i 即使 IoU 超过阈值也算作 **FP (False Positive)**。
 - **判断 TP/FP**:
 - 如果 P_i 成功匹配到一个未被匹配的 GT_j ，则 P_i 记为 **TP (True Positive)**。
 - 否则，记为 **FP**。
- 所有未被匹配的真实框 GT_k 记为 **FN (False Negative)**。

• 步骤 3.2: 计算给定阈值的 mAP (Mean Average Precision)

- 对每个类别独立进行上述匹配过程。
- 根据排序后的预测框，计算累积的 TP 和 FP。
- 对每个类别，计算 AP：
 - **计算 Precision 和 Recall**:
 - $Precision = TP / (TP + FP)$
 - $Recall = TP / (TP + FN)$
 - **计算 AP**:
 - 通过插值法计算Precision-Recall曲线下的面积 (AP)。
- 将所有类别的 AP (Average Precision) 值求平均，得到该阈值下的 mAP。

$$mAP = \frac{1}{N_{classes}} \sum_{i=1}^{N_{classes}} AP_i$$

• 步骤 4.3: 计算最终的 mAP@0.5:0.95

- 对每个 IoU 阈值 (0.50, 0.55, ..., 0.95) 重复上述匹配和 AP 计算过程。
- 最终取所有阈值下的 mAP 值的平均，得到最终的 **mAP@0.5:0.95**。

$$mAP@0.5 : 0.95 = \frac{1}{10} \sum_{i=1}^{10} mAP_{IoU_i}$$

mAP是目标检测任务中常用的评估指标，更详细的计算细节可以在[这一链接](#)找到。

4. 特殊情况处理

- 若 `data_id` 存在于 GT 中，但不在预测中：该样本得分为 0；
- 若 `data_id` 存在于预测中，但不在 GT 中：该样本得分也为 0；
- 若预测结果无法解析（如格式错误、缺字段等）：该样本得分为 0。

5. 最终得分

- 对所有样本的 mAP 取算术平均，即为最终提交的评分。