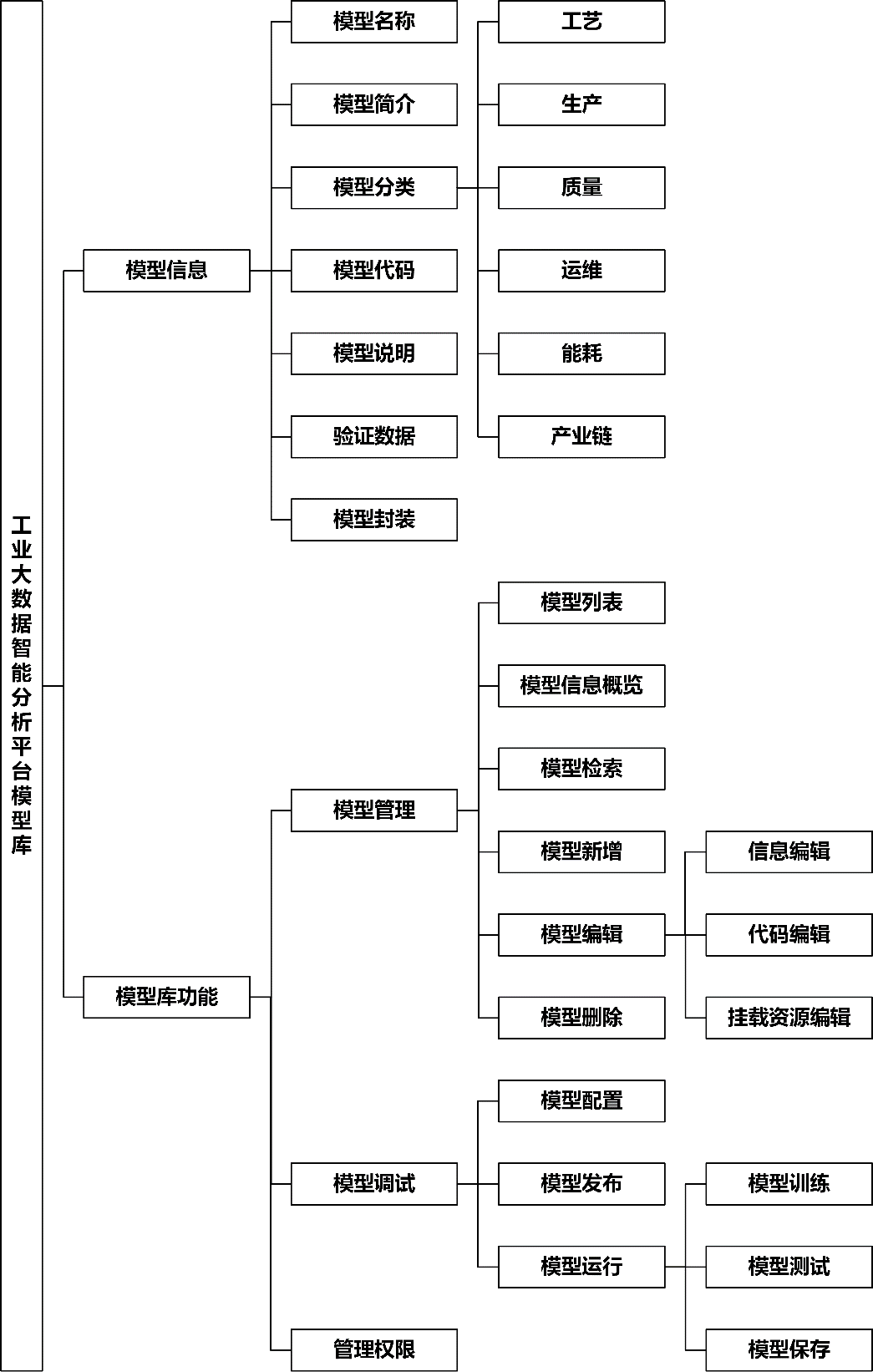
**工业大数据智能分析平台算法模型库开发方案**

**1. 模型库开发**

**1.1 模型库功能架构**



**图1 模型库功能架构**

**1.2 模型库开发方案**

**1.2.1 模型整理及要求**

**（1）模型名称：**

* **基于yolov7的人员检测模型**

**主要算法名称：yolov7，业务对象：人员，业务问题：人员检测**

**（2）模型简介：**

**利用coco数据集，采用人员特征提取的方法，构建人员检测模型，针对人员实现检测功能。**

**（3）模型分类：**

**模型应属于 “质量”，“运维”。**

**（4）模型代码：**

* **train.py: 模型训练**
* **test.py：模型评估**
* **detect.py: 模型测试**
* **yolo.py: 网络结构**
* **common.py 模块函数**

**（5）模型说明：**

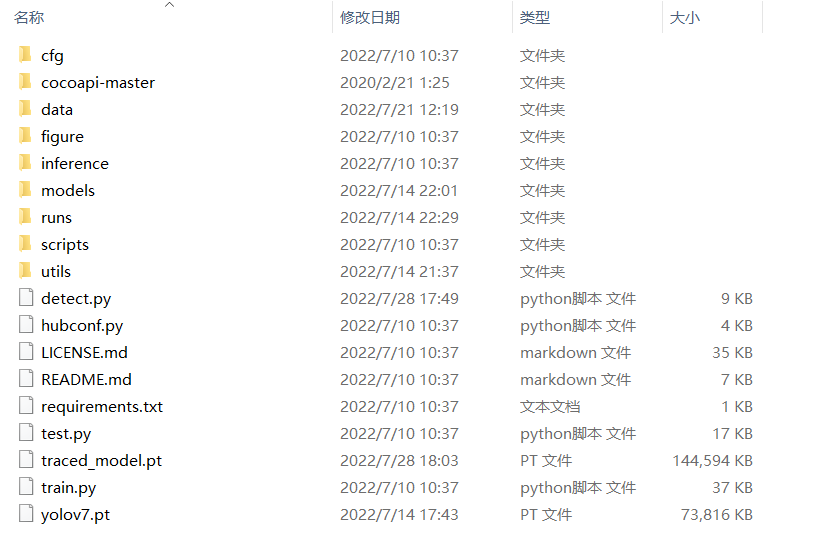
**1.模型概述**

**1.1 模型文件**

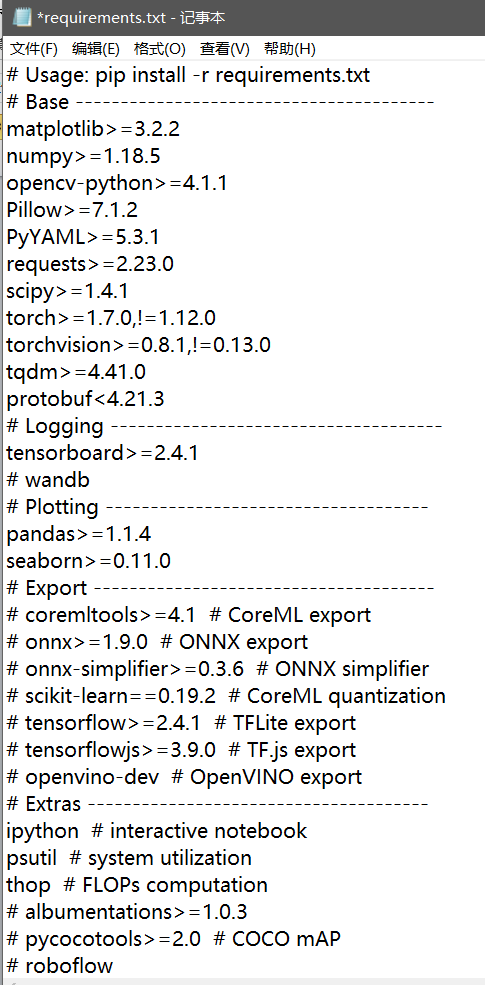
|  |  |  |
| --- | --- | --- |
| **序号** | **函数名称** | **函数说明** |
| **1** | **nms** | **非极大抑制，去除多余检测框** |
| **2** | **CrossConv** | **交叉卷积下采样** |
| **3** | **Model** | **模型构建** |
| **4** | **Detect** | **模型检测** |
| **5** | **FocalLoss** | **损失计算** |
| **6** | **create\_dataloader** | **数据加载器创建** |

**主函数名称：train**

**运行环境：pytorch 1.7. CUDA版本：10.2. 数据集：coco2017**

****

**模型代码文件**

****

**运行环境**

**1.2 研究对象： 人员**

**1.3 实现功能： 实现人员检测**

**2.原理概述**

**在YOLO算法中，核心思想就是把物体检测（object detection）问题处理成回归问题，用一个卷积神经网络结构就可以从输入图像直接预测边界框和类别概率。用回归的方法去做目标检测，执行速度快，达到非常高效的检测，其背后的原理和思想也非常简单。**

**3.接口说明**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **算法调用** | **输入接口** | **输出接口** |
| **1** | **dataloader=create\_dataloader(data[task], imgsz, batch\_size, gs, opt)** | **Data:训练数据**  **Imgsz：图片尺寸**  **Batch\_size ：批次大小** | **加载后的数据送入模型进行训练** |
| **2** | **pred=model(img,augment=opt.augment)** | **Img：测试图片** | **测试结果** |
| **3** | **train(hyp, opt, device, tb\_writer)** | **hyp：超参数**  **opt：模型参数**  **device：训练的设备** | **训练模型与结果** |

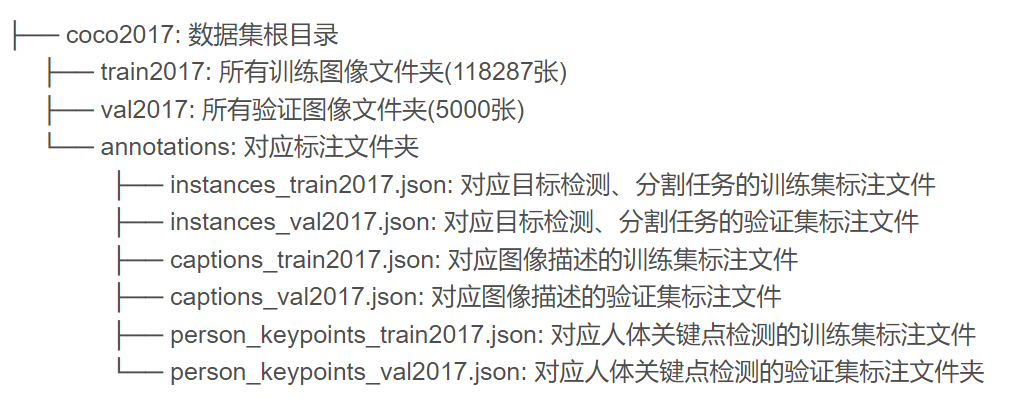
**4.数据集情况**

**4.1数据集来源**

**COCO数据集是微软公司出资标注的数据集，主要用于目标检测、分割和图像描述。**

**4.2 数据集组成**

**首先从coco官网下载数据集，本文以coco2017为例，下载得到train,val以及annotations，创建一个coco2017目录：**

****

**coco数据集**

**5.分析过程**

**输入训练数据，运行train.py文件，得到训练后的模型，通过test.py进行评估，最后通过detect.py进行模型测试，得到测试图像。**



**测试结果**

**（6）验证数据：**

* **train2017: 训练数据**
* **val2017: 测试数据**

**（7）模型封装：**

**模型应完成封装，封装过程应通过调用主函数实现模型构建和模型测试的功能。**

**1.2.2 模型库功能要求**

**针对模型库功能需求，可按照下述标准进行功能设计：**

**（1）模型管理：**

**① 模型列表：以列表或框图形式展示模型库模型整体情况。**

**② 模型信息概览：应对“模型名称，模型简介，模型分类，模型数量”等信息进行展示。**

**③ 模型检索：应能根据关键词对模型进行检索和筛选。**

**④ 模型新增：可以新增模型，应按“1.2.1 模型整理及要求”中信息进行填写。**

**⑤ 模型编辑：可对模型信息（“名称，简介，类型等信息”），代码，挂载资源（“数据，说明文档”）进行增删改查。**

**⑥ 模型删除：可删除模型。**

**（2）模型调试：**

**① 模型配置：可根据挂载资源配置模型训练数据和测试数据，可为模型配置超参数。**

**② 模型发布：针对模型完成数据和超参数等配置后，通过模型发布，进入待训练阶段。**

**③ 模型调试：可对发布后的模型进行训练，保存，和测试。**

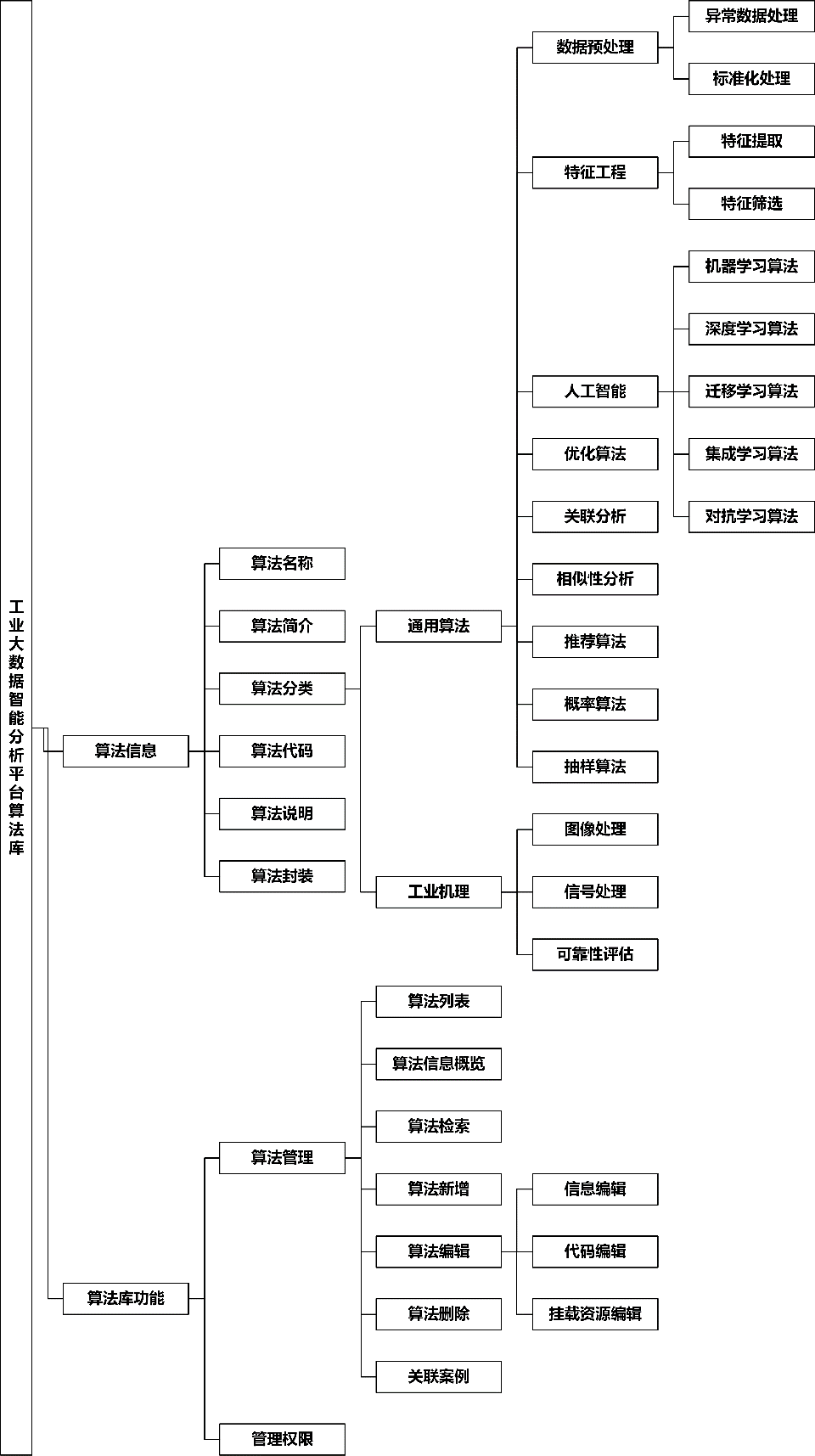
* **模型训练：针对发布后的模型，利用训练数据进行模型训练。**
* **模型测试：针对训练好的模型，利用测试数据进行测试。**
* **模型保存：将训练后的模型或测试符合要求的模型进行保存固定。**

**（3）管理权限：**

**根据人员角色设定其对应权限。**

**2. 算法库开发**

**2.1 算法库功能架构**



**图6 算法库功能架构**

**2.2 算法库开发方案**

**2.2.1 算法整理及要求**

**（1）算法名称：**

* **Yolov7**

**（2）算法简介：**

* **Yolov7：是一种用于物体(人员)检测的监督学习方法。**

**（3）算法分类：**

* **通用算法：该类型算法可在多个领域进行复用，无明显领域限制，按功能进一步划分为“人工智能”。**

**（4）算法代码：**

* **yolo.py: 网络结构**
* **common.py：模块函数**
* **train.py: 模型训练**
* **test.py：模型评估**
* **detect.py: 模型测试**

**（5）算法说明：**

**近年来，模型结构重参化和动态标签分配已成为网络训练和目标检测中的重要优化方向。在本文中，作者提出一些已经发现的问题，例如:对于模型结构重参化，用梯度传播路径的概念分析了适用于不同网络中各层结构重参化策略，提出了规划的模型结构重参化；当使用动态标签分配策略时，多输出层的模型在训练时会产生新的问题，比如怎样才能为不同分支更好的输出分配动态目标。针对这个问题，作者提出了一种新的标签分配方法，称为coarse-to-fine（由粗到细)引导标签分配策略。**

**（6）算法封装：**

**算法应完成封装，算法过程应通过调用算法接口支撑模型构建和模型测试的功能。**

**2.2.2 算法库功能要求**

**针对算法库功能需求，可按照下述标准进行功能设计：**

**（1）算法管理：**

**① 算法列表：以列表或框图形式展示算法库模型整体情况。**

**② 算法信息概览：应对“算法名称，算法简介，算法分类，算法数量”等信息进行展示。**

**③ 算法检索：应能根据关键词对算法进行检索和筛选。**

**④ 算法新增：可以新增算法，应按“2.2.1 算法整理及要求”中信息进行填写。**

**⑤ 算法编辑：可对算法信息（“名称，简介，类型等信息”），代码，挂载资源（“说明文档”）进行增删改查。**

**⑥ 算法删除：可删除算法。**

**⑦ 关联案例：应与模型库中使用该算法模型进行关联，提供关联案例。**

**（2）管理权限：**

**根据人员角色设定其对应权限。**

**3. 进度安排：**

**（1）时间计划：待定。**

**（2）工作计划：先开展模型库开发，再开展算法库开发。工业互联网中心配合完成页面及功能点整改，其他各中心按照“整理及要求”先统计模型库/算法库列表信息，并选取一个模型/算法作为示范模型/算法进行信息整理，代码整理，和代码封装等，上传模型库/算法库成功后，以示范模型/算法为例进行其他模型算法的整改。**