



# 《YOLO 代码复现框架》

## 用户需求说明书

v1.0



版本历史

版本/ 状态	作者	参与 者	日期	备注



# 目 录

《YOLO 代码复现框架》 .....	1
第一部分 引言 .....	5
一、说明 .....	5
二、定义 .....	5
1. YOLO 目标检测 .....	5
2. 置信度 .....	5
3. 边界框 .....	6
4. 非极大值抑制 .....	6
5. 推理速度 .....	6
6. 平均精度均值 .....	6
7. 网格划分 .....	6
8. 交并比 .....	6
第二部分 综述 .....	7
一、项目背景 .....	7
二、建设目标 .....	7
三、建设原则 .....	7
四、用户业务需求说明 .....	8
1、整体业务需求示意图 .....	8
2、需求详细说明 .....	9
2.1. 数据输入 .....	9
2.2 基线能力与模型 .....	9
2.3 任务再训练 .....	9
2.4 模型优化与部署 .....	10
2.5 检测结果输出 .....	10
2.6 系统管理与运维 .....	10
2.7 性能验收与测试 .....	11
2.8 风险、边界与合规 .....	11
第三部分 需求分析 .....	11
一、用例分析 .....	11
1. 训练模型用例 .....	11
1.1、数据处理与增强用例描述 .....	11
1.2、图片特征提取用例描述 .....	12
1.3、模型训练用例描述 .....	12
1.4、模型评估用例描述 .....	13
1.5、模型参数保存用例描述 .....	13
2、调用模型用例 .....	13



---

2.1、导入测试视频数据用例描述 .....	14
2.2、模型使用与数据分析用例描述 .....	14
2.3、查看识别结果用例描述 .....	14
2.4、生成分析报告用例描述 .....	15
3、模型应用与部署用例 .....	15
3.1、数据上传 .....	15
3.2、模型二次训练 .....	16
3.3、评估模型性能 .....	16
3.4、模型部署到特定场景 .....	17
二、界面风格 .....	17
第四部分 验收标准 .....	18
一、功能范围定义 .....	18
二、性能指标定义 .....	19
第五部分 环境和部署要求 .....	20
一、网络部署图 .....	20
二、应用部署图 .....	21
三、运行环境说明 .....	21
1、服务器 .....	21
2、客户机器环境 .....	22



## 第一部分 引言

### 一、说明

编写本说明书的目的是为了准确阐述项目具体业务需求和需求边界，本说明书的作者是【YOLO代码复现框架】项目组，本说明书的确认者是【徐思远】，本说明书的读者是项目所有直接干系人。

本说明书是指导项目实施的重要指导性文件，也是用户最后进行验收（终验）的依据，说明书中内容一旦确认双方将以此为基础开展工作。如果需要变更说明书内容，必须走变更流程，变更必须得到甲乙双方书面确认，最后变更内容将作为本文的一部分，在项目实施过程中得以体现。

### 二、定义

#### 1. YOLO 目标检测

YOLO(You Only Look Once)是一种端到端的实时目标检测算法。YOLO 将目标检测任务转化为单一回归问题，直接从图像像素到边界框和类别概率的预测，具有速度快、精度高的特点，广泛应用于自动驾驶、安防监控等领域。

#### 2. 置信度

置信度（confidence）是目标检测模型对检测结果可信程度的评分，表示模型认为某个检



测框中确实存在目标的概率。

### 3. 边界框

**边界框 (Bounding Box)** 是目标检测算法用于标记目标在图像中的具体位置的矩形框，通常用坐标表示

### 4. 非极大值抑制

**非极大值抑制 (NMS)** 是一种后处理算法，用于去除目标检测结果中的重复边界框，保留置信度最高的检测结果。

### 5. 推理速度

**推理速度 (Inference Speed)** 指目标检测模型在实际应用中处理图像的速度，通常以帧每秒 (FPS) 为单位，影响系统的实时性。

### 6. 平均精度均值

**平均精度均值 (mAP)** 是目标检测模型性能评估的重要指标，衡量模型在不同类别上的检测准确率，mAP 越高表示模型检测效果越好。

### 7. 网格划分

**网格划分 (Grid Division)** 是 YOLO 算法的核心思想，将输入图像划分为  $S \times S$  个网格单元，每个网格单元负责检测其中心点落在该单元内的目标。YOLO v1 中通常使用  $7 \times 7$  的网格划分。

### 8. 交并比

**交并比 (IoU, Intersection over Union)** 是衡量两个边界框重叠程度的指标，计算公式为重叠面积除以并集面积。IoU 广泛用于训练损失计算、NMS 阈值判断和 mAP 评估。



---

## 第二部分 综述

### 一、项目背景

随着目标检测技术的快速发展，本项目以 YOLO (You Only Look Once) 作为通用背景技术进行复现与实现，侧重构建一个端到端、可复用的 YOLO 代码框架（从数据准备、模型实现、训练与评估，到部署与示例演示）。项目同时提供不同历史版本（从 YOLOv1 起）的实现与比较分析，并包含若干通用演示案例（如交通检测、农业目标检测与视频监控），以便后续在具体应用中进行定制和性能提升。

### 二、建设目标

- 1、技术复现：多个 YOLO 版本全流程复现（含训练、损失、数据增强），性能偏差控制在  $\pm 5\%$  内，确保可复现实验结果。
- 2、性能优化：在标准数据集（VOC、COCO）上针对精度、速度或模型规模做改进，至少一项指标优于官方基线。
- 3、应用验证：选取真实场景（如交通或农业），完成模型适配与 Demo 开发，并输出场景性能分析报告。
- 4、代码与生态：产出高质量、模块化、可扩展的开源代码与文档，促进社区协作与持续迭代。
- 5、模块化扩展：采用可插拔架构，支持从 YOLOv1 到后续版本的平滑升级、模型压缩与不同推理后端的快速适配，并便于新增目标类别与任务微调。

### 三、建设原则

#### （一）实用有用

以实际应用价值为导向，构建端到端、易用的目标检测平台，覆盖数据准备、训练、评估与部署，优先解决通用场景中的识别与响应问题。

#### （二）灵活先进



系统应具备高度可配置性与扩展性，适配多种输入源与硬件平台，采用成熟且可持续演进的模型框架，便于未来 3-5 年内迭代升级。

### （三）界面友好

提供直观的可视化界面与 API，满足开发、测试和运维不同角色需求；可视化包括检测叠加、统计报表与日志检索。

### （四）兼容扩展

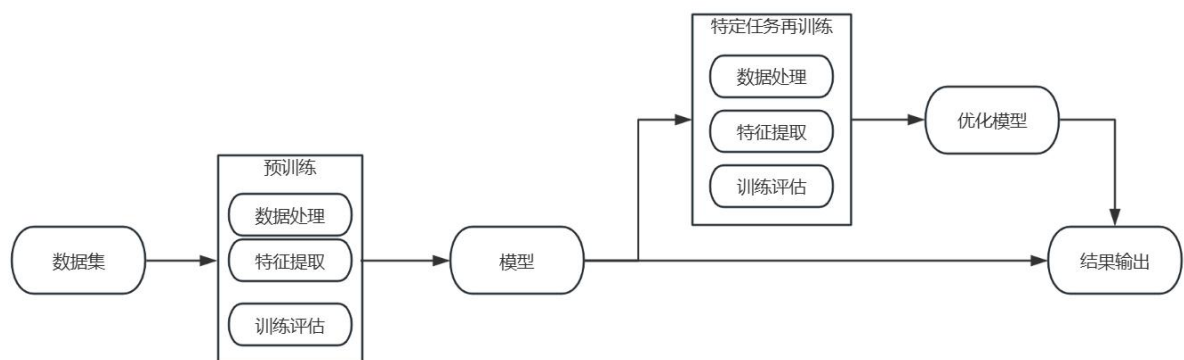
系统设计需预留标准化接口（REST/gRPC/MQTT），便于与上游/下游系统（如分析平台、决策模块）集成，并支持模型版本与配置管理。

### （五）安全可靠

确保数据与模型的安全（传输加密、模型签名、访问控制）、运行时稳定性（监控、告警、自动回退）与合规的数据保留策略，保障平台长期可用性与可审计性。

## 四、用户业务需求说明

### 1、整体业务需求示意图







## 2、需求详细说明

### 2.1. 数据输入

输入类型：支持多种来源的图像与视频流（静态图片、视频文件、实时摄像头/RTSP 流等），接受 RGB、灰度及多光谱输入，能导入标准数据集（COCO、VOC、VOL 等）或用户自定义标注数据用于训练与评估。

数据预处理：提供可配置的数据预处理模块（分辨率调整、畸变修正、色彩与亮度归一化、数据增强策略），以适配不同采集设备与场景。

数据质量控制：支持样本完整性校验、异常帧过滤、样本分布统计与分层抽样，为预训练与再训练提供数据健康度报告。

### 2.2 基线能力与模型

通用检测能力：支持通用目标类别（如人、车辆、物体、标志等）并允许扩展自定义类别，输出类别、边界框与置信度。

实时性与资源适配：提供多种模型配置（高精度/低延迟/轻量化），并给出在常见硬件下的性能预估（FPS、延迟、模型大小、显存/算力需求）。

评估指标：提供标准化评估（mAP、Precision/Recall、F1、IoU 分布）与类别级别报告，支持对不同场景或子集（夜间、低光、雨雪等）单独评测。

鲁棒性要求：在常见光照、天气与遮挡条件下保持稳定性能，支持多尺度检测、低光增强与多帧融合等提升手段。

### 2.3 任务再训练

可复用微调流水线：支持从预训练模型基线进行微调，包含数据划分、样本平衡、冻结策略、学习率调度与早停等常用配置。

迁移与扩展：提供类别映射与新类别增量训练方案，尽量减小新任务所需标注量与训练时间。



## 2.4 模型优化与部署

加速与压缩：支持量化、剪枝、蒸馏、ONNX/TensorRT/NCNN 等导出与适配，生成针对不同部署目标（边缘/服务器/移动设备）的优化模型。

部署适配：提供多种推理后端接入方式（PyTorch、ONNX Runtime、TensorRT、OpenVINO 等），并对每种后端给出性能预期与兼容性说明。

回退与安全：支持模型签名校验、版本回滚与灰度发布策略（A/B 测试）。

## 2.5 检测结果输出

输出结构（每帧）：类别标签、边界框坐标（像素或归一化）、置信度、时间戳；可选：跟踪 ID、速度或其他附加属性。

事件与记录：支持规则化事件生成（如高置信度检测、特定类别出现、阈值触发），并将事件记录为结构化日志或消息。

接口与可视化：提供直接叠加可视化（边框、标签、置信度）用于调试；同时提供标准化上报接口（REST/gRPC/MQTT）以供后端消费。

置信度与不确定性：每条检测需带置信度；在需要时提供不确定性等级与后处理（NMS、平滑、历史融合）以提高稳定性。

## 2.6 系统管理与运维

配置管理：提供 UI 或 CLI 用于调整阈值、选择关注类别、调整推理分辨率与性能模式；支持配置版本管理与回滚。

日志与事件记录：记录检测日志、训练日志、运行时异常与事件，支持按时间/类别检索与导出关键帧或视频片段以便审查。

状态监控：实时展示数据流状态、帧率、推理延迟、算力/内存占用与服务健康；异常时支持告警与自动降级策略。

更新与升级：支持安全模型/代码更新机制（签名校验、回滚、渐进发布），并记录升级影响的在线指标。



## 2.7 性能验收与测试

基线验收指标：延迟/FPS、mAP（总体与关键类别）、召回率、资源占用与鲁棒性降幅（在夜间/雨雪场景）。

验收方法：使用标准数据集与场景化测试集进行离线评估；微调后在验证集给出训练曲线与最终报告；部署后通过在线日志验证实际性能。

## 2.8 风险、边界与合规

常见边界场景：极端光照、严重遮挡、异常摄像设备损坏或帧丢失、罕见类别样本不足。

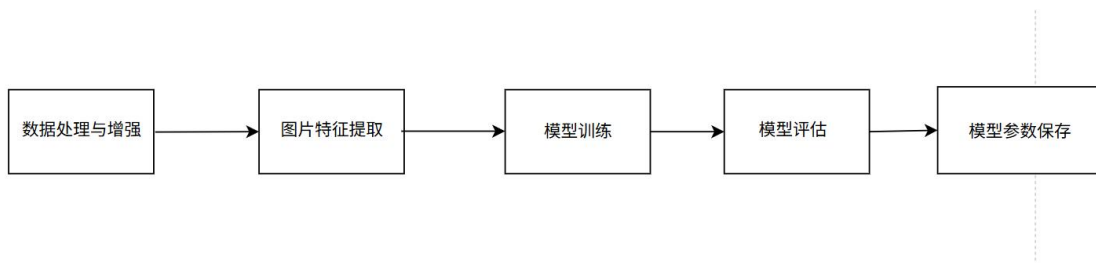
异常策略：当置信度不足或系统资源不足时输出“不确定”或降级结果，记录详尽日志并触发人工复核流程。

隐私与合规：对视频与日志实行访问控制、最小化保存策略与合规记录，以满足数据保护要求。

# 第三部分 需求分析

## 一、 用例分析

### 1. 训练模型用例



#### 1.1、数据处理与增强用例描述

ID	YOL0000
用例名称	数据处理与增强



父用例 ID	-
主要执行者	研究人员
前置条件	
事件流	通过裁剪、变色、加噪等方式处理输入图片，并将图片转化为模型需要的输入格式
可选事件流	无
异常事件流	未检测到图片
后置条件	无

## 1.2、图片特征提取用例描述

ID	YOL0001
用例名称	图片特征提取
父用例 ID	YOL0000
主要执行者	研究人员
前置条件	数据处理完成
事件流	通过骨干网络（ResNet、VGG 等网络）将输入图片输出为一个特征向量
可选事件流	无
异常事件流	
后置条件	无

## 1.3、模型训练用例描述

ID	YOL0002
用例名称	模型训练
父用例 ID	YOL0001
主要执行者	研究人员
前置条件	图片特征提取完成
事件流	先通过分类器预训练训练骨干网络权重，再将权重加载到分类头之中。在检测头训练中，我们输入图片/视频流，在一系列处理后得到一个特征向量，再将特征向量输入分类头，得到对应的输出，并与目标输出计算训练损失与训练平均精度，通过梯度回传调整模型参数。
可选事件流	无
异常事件流	
后置条件	无



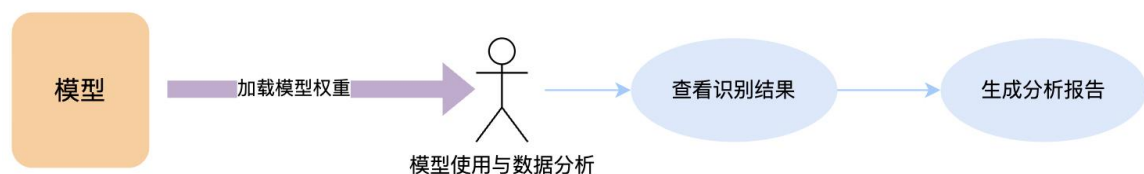
## 1.4、模型评估用例描述

ID	YOL0003
用例名称	模型评估
父用例 ID	YOL0002
主要执行者	研究人员
前置条件	模型训练完成
事件流	将模型转为评估模式，使用我们训练好的模型进行测试并输出损失与平均精度均值，记录结果并评估模型性能
可选事件流	无
异常事件流	
后置条件	无

## 1.5、模型参数保存用例描述

ID	YOL0004
用例名称	模型参数保存
父用例 ID	YOL0003
主要执行者	研究人员
前置条件	模型训练与评估成功
事件流	将模型参数保存到指定路径
可选事件流	无
异常事件流	模型保存失败
后置条件	无

## 2、调用模型用例





## 2.1、导入测试视频数据用例描述

ID	YOL0005
用例名称	调用预训练模型
父用例 ID	-
主要执行者	数据分析师
前置条件	
事件流	通过 Web Service 或本地路径加载模型权重
可选事件流	无
异常事件流	模型加载失败
后置条件	无

## 2.2、模型使用与数据分析用例描述

ID	YOL0006
用例名称	调用预训练模型
父用例 ID	YOL0005
主要执行者	数据分析师
前置条件	模型权重预加载完成
事件流	输入需要检测与分析的图片或视频流，并调用模型进行目标检测，输出检测结果并保存到对应路径
可选事件流	无
异常事件流	未检测到目标物体
后置条件	无

## 2.3、查看识别结果用例描述

ID	YOL0007
用例名称	查看识别结果
父用例 ID	YOL0006
主要执行者	数据分析师
前置条件	模型检测正常进行
事件流	从对应路径获得检测结果并展示

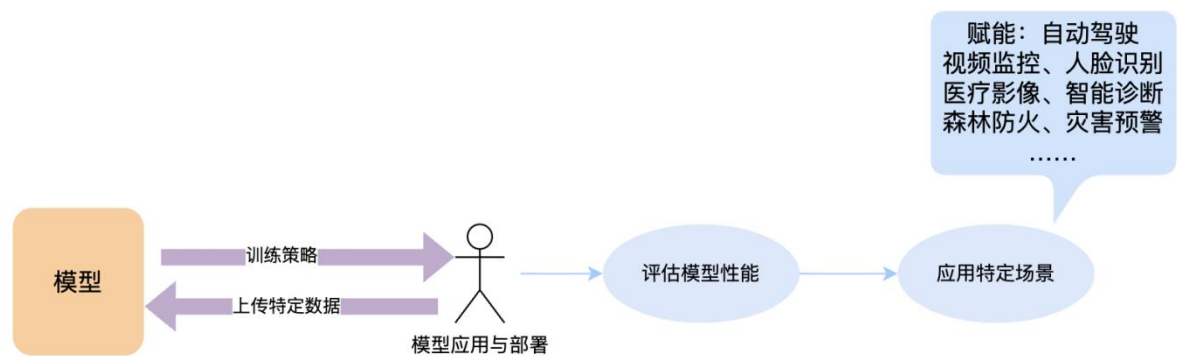


可选事件流	无
异常事件流	无可用路径
后置条件	无

## 2.4、生成分析报告用例描述

ID	YOL0008
用例名称	生成分析报告
父用例 ID	YOL0007
主要执行者	数据分析师
前置条件	得到正确的识别结果
事件流	系统自动或手动输出各类评估指标，导出 PDF/Excel 测试结果
可选事件流	无
异常事件流	
后置条件	无

## 3、模型应用与部署用例



### 3.1、数据上传

ID	YOL0009
用例名称	数据上传



父用例 ID	-
主要执行者	开发者
前置条件	
事件流	将私有数据集上传到模型训练接口中
可选事件流	YOLO 模型版本
异常事件流	
后置条件	无

### 3.2、模型二次训练

ID	YOL0010
用例名称	模型二次训练
父用例 ID	YOL0009
主要执行者	开发者
前置条件	新的训练数据已上传
事件流	加载预训练的模型，根据模型说明文档用指定数据集训练模型
可选事件流	无
异常事件流	
后置条件	无

### 3.3、评估模型性能

ID	YOL0011
用例名称	评估模型性能
父用例 ID	YOL0010
主要执行者	开发者
前置条件	模型训练完毕
事件流	查看模型识别结果，计算各项评价指标并与现有模型对比
可选事件流	无
异常事件流	
后置条件	无



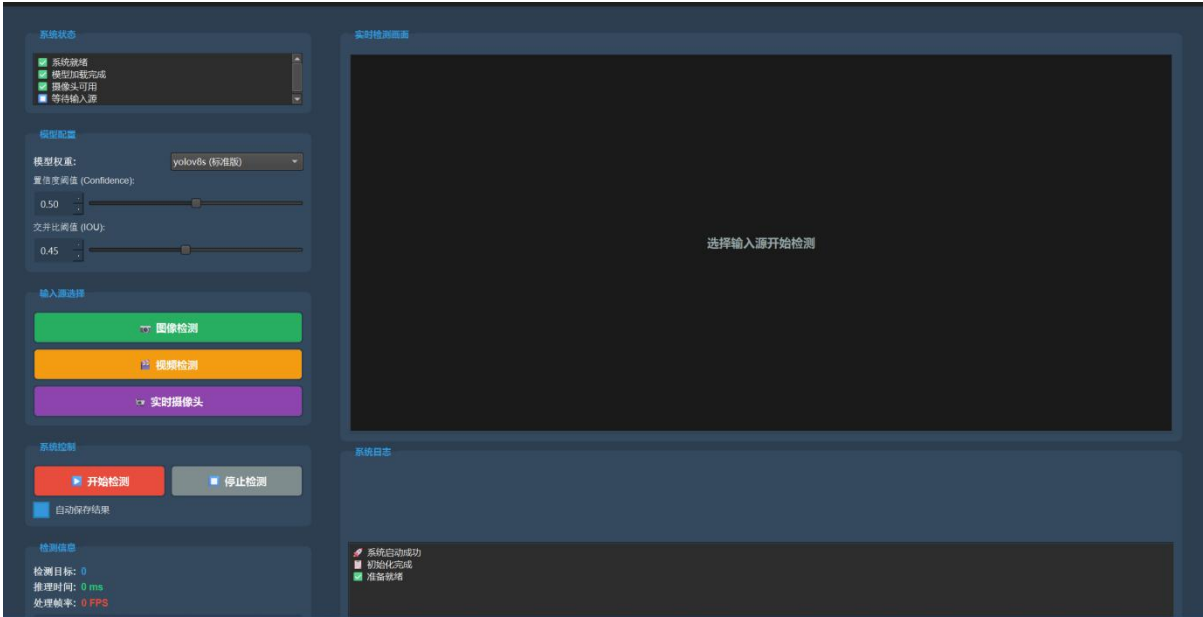


### 3. 4、模型部署到特定场景

ID	YOL0012
用例名称	模型部署到特定场景
父用例 ID	YOL0010、YOL0011
主要执行者	开发者
前置条件	模型已训练完毕且获得最优的表现结果
事件流	将训练好的模型部署到用户指定的使用场景下
可选事件流	无
异常事件流	
后置条件	无

## 二、界面风格

主页面：





## 第四部分 验收标准

### 一、功能范围定义

#	产品	模块	组件		角色	接入
1	YOLO 代码复现框架	模型架构实现	YOLOv1 基础网络	复现论文中的 Backbone + Detection Head，输出 7x7x30 张量	系统管理员	Web
2			YOLOv2-v13 改进模块	实现 Anchor 机制、多尺度预测、CSP 结构、注意力机制等	系统	-
3		数据预处理	数据加载与增强	支持 COCO/VOC 格式，实现随机裁剪、翻转、色彩抖动等		
4		训练流程	损失函数实现	实现 YOLO 损失（坐标、置信度、分类损失）		
5			训练调度与优化器	支持 SGD/Adam，学习率衰减、warmup 策略		
6		评估与验证	mAP 计算模块	实现 COCO 评估标准，输出 AP@0.5、AP@0.5:0.95 等		
7			可视化模块	绘制损失曲线、PR 曲线、检测结果可视化		
8		部署与应用	模型导出与转换	支持导出为 ONNX、TorchScript 格式	车载系统	API/边缘部署
9			推理 Demo	实现交通检测/农业作物检测等应用场景 Demo		



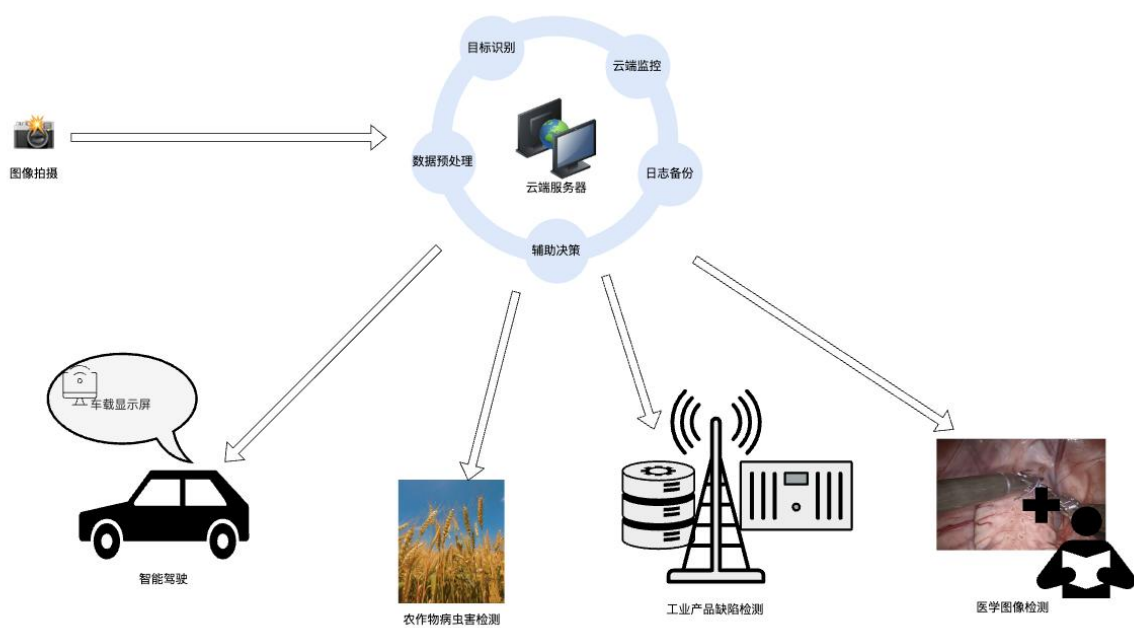
## 二、性能指标定义

#	产品	模块	组件	规格/型号	性能级别
1	YOLO 代码复现框架	模型架构实现	YOLOv1 基础网络	复现精度达到论文报告的 $mAP@0.5:0.95 \geq 60\%$	A
2			YOLOv2-v13 改进模块	各版本 mAP 与速度指标达到官方报告的 90%以上	A
3		数据预处理	数据加载与增强	支持多线程加载, 吞吐量 $\geq 1000$ img/s	B
4		训练流程	损失函数实现	训练收敛性良好, 无梯度爆炸/消失	A
5			训练调度与优化器	训练时间在单卡 GPU 上 $\leq 24$ 小时 (COCO 数据集)	B
6		评估与验证	mAP 计算模块	评估速度 $\geq 100$ img/s, 结果与官方评估工具一致	A
7			可视化模块	图像检测推理速度 $\geq 30$ FPS (1080p 图像)	A
8		部署与应用	模型导出与转换	导出模型可在 TensorRT/OpenVINO 上推理, 速度提升 $\geq 2$	B
9			推理 Demo	Demo 界面友好, 支持实时摄像头/视频流检测	B

**说明:** 级别 (A: 表示非常重要必须达到的技术性能要求, B: 表示重要推荐达到的技术性能要求, C: 表示非重要可以弱化的技术性能要求.)

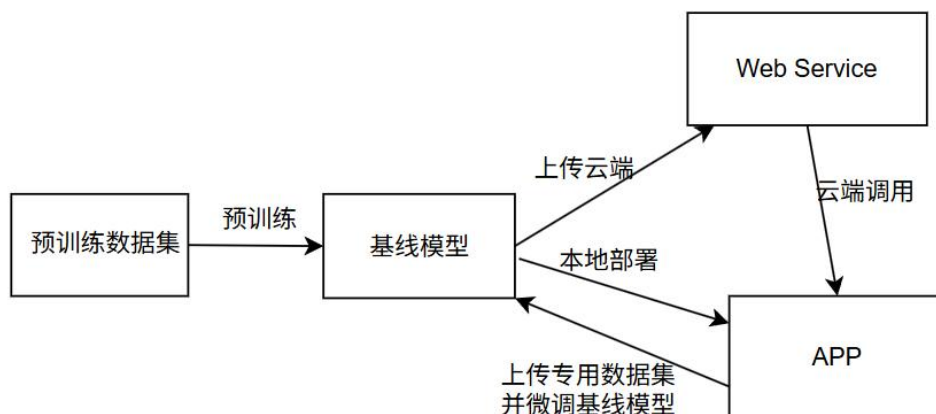
## 第五部分 环境和部署要求

### 一、网络部署图





## 二、应用部署图



## 三、运行环境说明

### 1、服务器

- 1) 操作系统: Ubuntu 20.04 LTS
- 2) 硬件要求:
  - GPU: NVIDIA RTX 3090 / A100 / Tesla T4
  - CPU: Intel Xeon 或 AMD EPYC, 8 核及以上
  - 内存: 32GB 以上
  - 硬盘: 1TB SSD 或更高, 需支持高速读写用于视频数据处理
- 3) 深度学习框架:
  - Python 3.10
  - PyTorch  $\geq 2.0$
  - YOLOv8 (Ultralytics)
  - OpenCV、NumPy
- 4) 本地开发环境:



---

操作系统: Windows 10 / macOS / Ubuntu / Windows 11

IDE: VS Code / PyCharm

模拟工具: CARLA / LGSVL 自动驾驶仿真器

## 2、客户机器环境

- 1) Pentium III 或以上微处理器 (CPU) ;
- 2) 操作系统: Windows/MacOS/Ubuntu;
- 3) 512MB 以上内存, 建议使用 1024MB 内存;
- 4) WEB: 要求 IE7 以上版本, 最好 IE8 以上版本。