
《YOLO 模型》 分析与评估报告 V1.0

版本/ 状态	作者	参与 者	日期	备注

目 录

目 录.....	3
1. 文档介绍.....	3
1.1 文档目的.....	3
1.2 文档范围.....	3
1.3 读者对象.....	3
1.4 参考文献.....	4
1.5 术语与缩写解释.....	4
2. 模型的算法与技术分析.....	4
2.1 YOLOv1 模型的算法与技术分析.....	4
2.1.1 YOLOv1 模型的算法分析.....	4
2.2 YOLOv3 模型的算法与技术分析.....	5
2.3 SwinYOLO 模型的算法与技术分析.....	6
3. 模型的训练数据集与增强处理说明.....	7
3.1 模型的训练数据集说明.....	7
3.1.1 VOC 数据集说明.....	7
3.1.2 COCO 数据集说明.....	7
3.2 模型的训练增强处理说明.....	7
4. 模型训练的常规参数分析与评估.....	7
4.1 YOLOv1 的参数分析与评估.....	7
4.1.1 参数量与占比.....	7
4.1.2 架构要点.....	8
4.1.3 训练与资源.....	8
4.2 YOLOv3 的参数分析与评估.....	8
4.2.1 参数量与占比.....	8
4.2.2 架构要点.....	8
4.2.3 训练与资源.....	8
4.3 SwinYOLO 的参数分析与评估.....	9
4.3.1 参数量与占比.....	9
4.3.2 架构要点.....	9
4.3.3 训练与资源.....	9
5. 模型的商用结论与优势.....	9
5.1 总体商业结论（面向上线决策）.....	9
5.2 场景—模型选型对照.....	9
5.3 交付 KPI 与 SLA 建议（建议阈值）.....	10
5.4 部署与性能优化建议（可直接落地）.....	10
5.5 成本（TCO）与资源规划.....	10
5.6 风险与缓解措施.....	11
5.7 运行监控与 A/B 策略.....	11
5.8 交付物清单.....	11

1. 文档介绍

1.1 文档目的

本文档给软件测试部门对《YOLO 代码复现框架》项目上线前的测试维护工作提供整体测试方案。

1.2 文档范围

本文档仅对《YOLO 代码复现框架》项目有效

1.3 读者对象

《YOLO 代码复现框架》项目项目相关负责人
《YOLO 代码复现框架》项目软件测试部门相关测试人员

1.4 参考文献

《系统详细设计文档》
《系统概要设计文档》
《用户操作手册》

1.5 术语与缩写解释

缩写、术语	解释
置信度	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。
边界框	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
非极大值抑制	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
推理速度	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

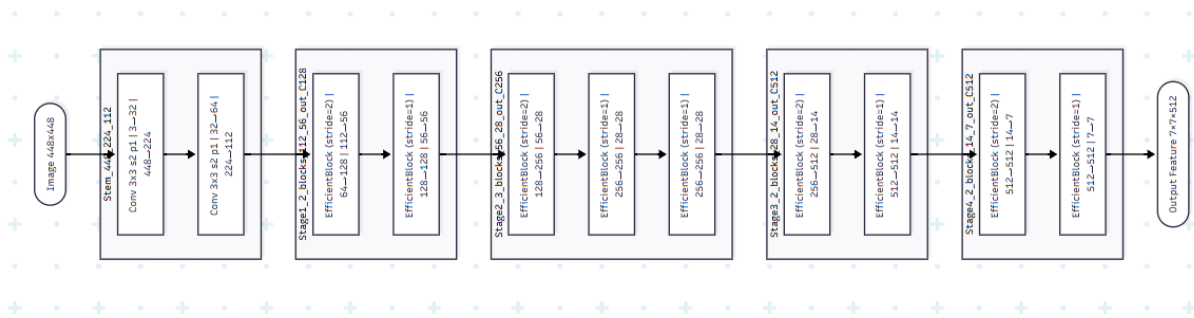
	系统的实时性
平均精度均值	目标检测模型性能评估的重要指标，衡量模型在不同类别上的检测准确率，mAP 越高表示模型检测效果越好
交并比	衡量两个边界框重叠程度的指标，计算公式为重叠面积除以并集面积。IoU 广泛用于训练损失计算、NMS 阈值判断和 mAP 评估

2. 模型的算法与技术分析

2.1 YOLOv1 模型的算法与技术分析

2.1.1 YOLOv1 模型的算法分析

1.backbone 网络结构



考虑到我们的训练方式和原文有所出入，我们采用了简化版的 **darknet** 网络作为主干网络，与原网络相比我们的网络会更加简洁高效，并且不会因为网络过于复杂导致过拟合的发生。

2.检测头结构

检测头结构我们保留了原文的做法，在 **backbone** 后面连接了两个全连接层，用于输出检测结果

3.优化器的选择

考虑到局部最优等问题，我们使用 **Adam** 作为训练的优化器

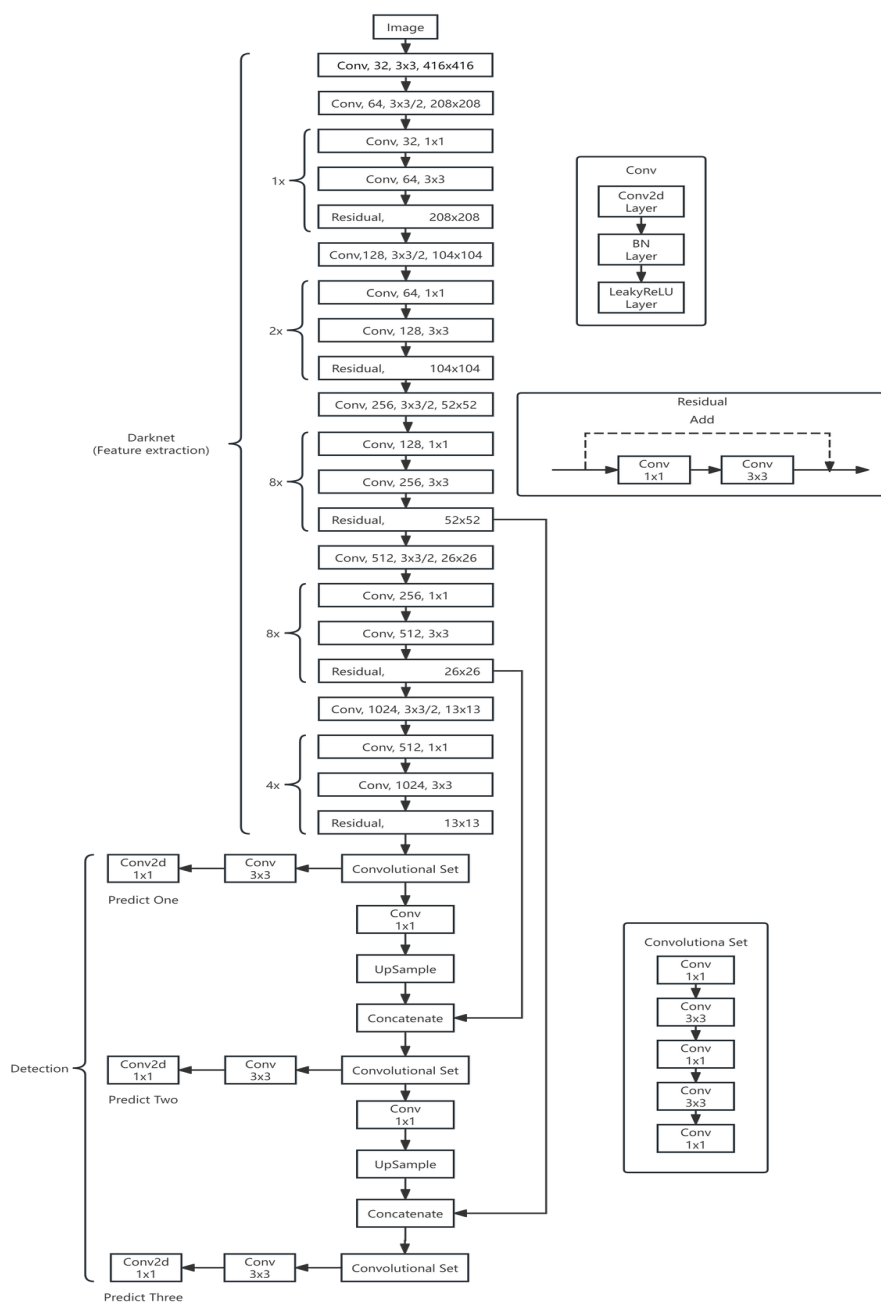
2.2 YOLOv3 模型的算法与技术分析

1.整体网络结构

YOLOv3 我们完全参照的原文的网络结构：**Darknet53+FPN**（作为检测头），具体网络结构如图

2.优化器的选择

考虑到局部最优等问题，我们使用 **Adam** 作为训练的优化器



2.3 SwinYOLO 模型的算法与技术分析

1.整体网络结构

我们选择 Swin-Large 官方权重作为骨干网络并在 backbone 后接上了一个 YOLOv1 的检测头

2. 优化器的选择

考虑到局部最优等问题，我们使用 Adam 作为训练的优化器

3. 模型的训练数据集与增强处理说明

3.1 模型的训练数据集说明

本项目主要利用了 COCO2017 与 VOC2007+2012 数据集作为训练数据集

3.1.1 VOC 数据集说明

我们的VOC数据集主要包括了2007与2012两年的VOC数据集，由于官方下载路径不可用，我们从 Kaggle 上下载了数据集，并保存在本地，Kaggle 上数据集的链接为

<https://www.kaggle.com/datasets/vijayabhaskar96/pascal-voc-2007-and-2012>

3.1.2 COCO 数据集说明

我们的 COCO 数据集选择的官方发布的 COCO2017 数据集

3.2 模型的训练增强处理说明

模型的数据增强方式主要包括：

1. 随机裁剪
2. 色彩变幻
3. 旋转变换

4. 模型训练的常规参数分析与评估

4.1 YOLOv1 的参数分析与评估

4.1.1 参数量与占比

Backbone 2.00M ($\approx 57.5\%$)

检测头 1.48M ($\approx 42.5\%$)

总计 3.48M。

4.1.2 架构要点

1×1 降维 \rightarrow DW 3×3 \rightarrow PW 1×1 的高效块，在 $7 \times 7 \times 512$ 输出处与 YOLOv1 网格天然对齐。

以较少参数获得足够感受野，但小目标表征能力有限（仅单尺度 7×7 ）。

4.1.3 训练与资源

轻显存、易训；12GB 显卡下 AMP 可用较大 batch（32 - 64）。

建议 SGD($lr \approx 0.01 @ eff_bs=64$, $wd=5e-4$) 或 AdamW($lr \approx 0.002 - 0.003$); 3 - 5ep warmup + cosine, 总 200 - 300ep; EMA 有益。

4.2 YOLOv3 的参数分析与评估

4.2.1 参数量与占比

Backbone 40.58M ($\approx 66.7\%$)

FPN+检测头 20.27M ($\approx 33.3\%$)

总计 60.85M。

4.2.2 架构要点

DarkNet 53 残差堆叠稳定可靠；FPN 三尺度（13/26/52）显著提升小/中目标召回。

输出层随类别线性增长（每尺度 $3 \times (5+C)$ ）。

4.2.3 训练与资源

12 - 24GB 显卡下 AMP 单卡 batch 通常 16 - 32; 273 - 300ep 基线训练。

SGD(lr \approx 0.01@eff_bs=64, wd=5e-4); Mosaic+MixUp、多尺度训练、AutoAnchor; 小 batch 用 SyncBN 或冻结 BN; EMA 建议开启。

4.3 SwinYOLO 的参数分析与评估

4.3.1 参数量与占比

Backbone 195.56M (\approx 98.2%)

检测头 3.55M (\approx 1.8%)

总计 199.12M。

4.3.2 架构要点

层级化 Transformer, 窗口注意力; 极强表征力, 对高分辨率与大数据收益明显。

输出适配到 512 通道再接 YOLO 栈, 头部很轻。

4.3.3 训练与资源

对显存/算力要求最高; 24 - 40GB 显卡通常需 AMP+梯度检查点+累积(单卡有效 batch 32 - 64)。

AdamW(lr 1e-4 \sim 3e-4, wd \approx 0.05), 长 warmup(\geq 10ep) + cosine; DropPath \approx 0.2、Grad Clip 1.0; 尽量使用 ImageNet 预训练。

5. 模型的商用结论与优势

5.1 总体商业结论(面向上线决策)

YOLOv1 (3.48M, 13.3MB): 低成本实时/边缘优选, 推理延迟低, 易部署。

YOLOv3 (60.85M, 232.1MB): 通用生产基线, 多尺度稳健, 小目标更优。

SwinYOLO (199.12M, 759.6MB): 高精度/高分辨率/大数据场景, 需更高资源。

建议: 以“满足业务阈值的最小模型”为原则, 兼顾精度、时延与成本。

5.2 场景—模型选型对照

场景	业务特征	推荐模型	主要优势	主要约束/部署
边缘实时检测 (单/少路)	时延敏感、成本敏感、场景可控	YOLOv1	体积小、推理快、易量化	TensorRT FP16/INT8; NMS 引擎内融合; batch=1 流水线
多场景通用生产	目标尺度差异大、鲁棒性高	YOLOv3	FPN 三尺度、成熟稳定	Mosaic+MixUp; AutoAnchor; SyncBN; 多尺度训练
高精度复杂场景	高分辨率/远距离密集, 精度优先	SwinYOLO	Transformer 表征力强	预训练 +AdamW+长 warmup; AMP+检查点+累积

5.3 交付 KPI 与 SLA 建议 (建议阈值)

准确率 (VOC): $mAP@0.5 \geq 0.80$ (YOLOv1)、 ≥ 0.85 (YOLOv3)、 ≥ 0.88 (SwinYOLO)。

准确率 (COCO): $mAP@0.5:0.95 \geq 0.33$ (YOLOv1)、 ≥ 0.38 (YOLOv3)、 ≥ 0.43 (SwinYOLO)。

推理时延 (FP16/TensorRT, 单帧): YOLOv1 $\leq 20ms$; YOLOv3 $\leq 35ms$; SwinYOLO $\leq 60ms$ 。

吞吐建议 (1080p, 多路): Orin-NX: YOLOv1 2-6路、YOLOv3 1-3路; RTX3060/4060: YOLOv1 8-16路、YOLOv3 4-8路、SwinYOLO 1-2路。

上线验收: 以业务阈值为先 (漏/误检率、端到端时延、稳定运行 ≥ 7 天)。

5.4 部署与性能优化建议 (可直接落地)

推理链路: PyTorch \rightarrow ONNX (opset 13/17) \rightarrow TensorRT (FP16/INT8); NMS 合并引擎内。

量化: 优先 FP16; INT8 需 $\geq 2K$ 高质量校准集并覆盖各场景与光照。

并发: 多流队列+单引擎多上下文; batch=1; 流水线 Decode \rightarrow Preproc \rightarrow Infer \rightarrow NMS \rightarrow Postproc。

内存: 固定分辨率与显存预分配; 减少 host device 拷贝; 合并算子, 使用页锁定内存。

YOLOv1: 小目标不足 \rightarrow 提高输入分辨率/加一层金字塔; NMS IoU=0.5-0.6 控误检。

YOLOv3: 启用 Mosaic+MixUp、AutoAnchor; 小 batch 下用 SyncBN 或冻结 BN。

SwinYOLO: 使用预训练; AdamW+长 warmup+DropPath; AMP+梯度检查点+累积。

5.5 成本 (TCO) 与资源规划

训练成本: SwinYOLO 最高; YOLOv3 中等; YOLOv1 最低。

推理成本: 单位 QPS 成本 YOLOv1 最优; YOLOv3 适中; SwinYOLO 最高。

策略: 以“满足业务阈值的最小模型”为原则; 采用“轻量在线+高精度离线复核”分层架构。

5.6 风险与缓解措施

小目标/密集漏检：提高输入分辨率；调整 anchor/正负样本阈值；增强数据覆盖。

数据域偏移：颜色抖动/天气增强；跨域验证集 A/B；周期性再训练与增量学习。

时延超标：FP16/INT8；NMS 合并；降输入尺度；算子融合与流水线；必要时“轻→重”两段式。

训练不稳（Swin）：预训练；更长 warmup/更低 LR；GradScaler+梯度裁剪；检查点+累积。

5.7 运行监控与 A/B 策略

指标：mAP、召回率、漏/误检、端到端时延、GPU/CPU 利用率、Crash 率。

A/B：灰度 5 - 10%流量 \geq 7 天；劣化自动回滚；问题样本回流用于再训练。

5.8 交付物清单

训练产物：模型权重（.pt/.onnx/.engine）、配置、类别映射、（如 INT8）校准集。

文档：部署说明、接口协议、SLA 与验收报告、变更记录。

工程：导出脚本、TensorRT 构建脚本、监控/日志脚本、A/B 配置模板。