

# 虫害检测模型构建

下面按模块给出清晰的方案、理由与实现细节（可直接拿去做实验）。是否采取交叉取决于数据集大小。

## 1. 系统总体架构（一句话）

用强学习能力的大模型（DINO-DETR 或 SAM2，经 ConvNeXt/ViT 微调并用 LoRA 适配虫害）作为 Teacher，通过多路蒸馏（logits + feature + attention）训练轻量 Student（YOLOv11 + MobileNet + 轻量注意力模块），再对 Student 做稀疏训练→结构化剪枝→量化与加速（ONNX/TensorRT）后部署到嵌入式设备。

## 2. 数据与前处理（关键点）

- 数据类型：RGB 图片（含不同作物/光照/粘虫纸/诱捕装置），标注 bbox（必要时 mask）。
- 数据增强：Mosaic、MixUp、Multi-scale random crop、Color jitter、CutMix（注意小目标不可过度裁剪）。
- 采样策略：为少数类/小目标做过采样或使用 class-balanced sampler。
- 数据划分：训练/验证/测试（按场景划分，避免同一地点分散在 train/test），保留未标注大量图片用于半/无监督训练或伪标注。

## 3. Teacher 端（高性能，大模型）

- 备选：**DINO-DETR**（实时化 DETR 变体）或 **SAM2**（若需 segmentation 能力）。
- Backbone：ConvNeXt 或 ViT（视 Teacher 是否需全局 mask；SAM2 本身以 ViT 更合适）。
- 微调策略：
  - 使用 LoRA 在 Transformer 或 attention 层做参数高效适配（只训练 adapter），节省显存与时间。
  - 在虫害数据上做少量 epoch 精细微调，并使用强数据增强 + 半监督伪标注扩大数据。
- 输出：类别 logits、每层/每尺度的 feature maps、attention map（或 SAM 的 mask tokens）供蒸馏用。

## 4. Student 端（轻量，可部署）

- 架构：**YOLOv11n head + MobileNetV4 (或 MobileNetV#) backbone**。
- 轻量注意力：在 backbone 的关键 stage 插入 **ECA / SE / CBAM-lite**（推荐 ECA，因为几乎零计算额外）。
- 可选 hybrid head：在 neck 中增加一层小型 transformer-like cross-attention block（1 层、低维度），仅在蒸馏实验验证其收益；如果延迟显著增加则删除。

## 5. 蒸馏设计（检测专用）

采用 **多级蒸馏**，覆盖输出与中间层：

- Logits 蒸馏（分类）**：Soft targets（温度 T），学生用 KL divergence 学习 teacher 的类别分布。

- 公式:  $L_{\text{logit}} = \text{KL}(\text{softmax}(z_T/T), \text{softmax}(z_S/T))$
- **Bounding-box 蒸馏 (回归)**: 用 teacher 的回归目标或更稳定的 SIoU / GIoU 指导 student 的边界回归 ( $L1 + \text{IoU loss}$ )。
- **Feature 蒸馏**: 在 FPN/neck 的对应尺度上做 L2 或 通道注意力加权 MSE。建议使用 **attention-guided feature distillation**: 先对 teacher 的 feature 做空间注意力 / saliency map (对小目标区域放大权重), 再将 student 的对应 feature 与之对齐。
- **Attention 蒸馏 (可选)**: 直接让 student 学 teacher 的注意力热图 (例如 Transformer 的 QK attention map, 或 SAM 的 mask 的激活图)。
- **总损失**:
$$L = \alpha L_{\text{det\_gt}} + \beta L_{\text{logit}} + \gamma L_{\text{feat}} + \delta L_{\text{attn}} + \epsilon L_{\text{bbox}}$$
$$L = \alpha L_{\text{det\_gt}} + \beta L_{\text{logit}} + \gamma L_{\text{feat}} + \delta L_{\text{attn}} + \epsilon L_{\text{bbox}}$$
  - 推荐初始权重:  $\alpha=1.0$  (确保 task loss 仍重要),  $\beta=1.0$ ,  $\gamma=2.0$  (特征蒸馏优先),  $\delta=0.5$  (若使用 attention),  $\epsilon=2.0$  (若 bbox 非常关键)。这些超参需在验证集上调优。

## 6. 训练细节与技巧

- 优化器: AdamW 或 SGD+momentum (Student 用 SGD 有时更稳)。LR schedule 用 Cosine annealing + warmup。
- Batch size: 能用多大的就用多大 (蒸馏时 teacher 推理也要并行), 若显存受限可用梯度累积。
- 预训练: Student 初始化用 ImageNet 或 COCO 预训练权重 (同系列更好)。
- 半监督扩展: 用 teacher 生成伪标签 (高置信度 threshold), 并用 SoftTeacher 式的软标签策略加入到 student 训练中以提升泛化。
- 小目标策略: 保留高分辨率输入 (例如  $1024 \times 1024$  训练/验证时随机裁剪为  $640/800$ ), 多尺度训练。

## 7. 压缩与部署 (训练后到上线)

流水线建议:

1. **稀疏化训练 (可选)**: 在学生训练阶段加入 L1 正则或剪枝感知正则, 使一部分权重趋于 0。
2. **结构化剪枝**: 基于通道重要性 (BN gamma 值或基于梯度/activation) 进行通道/层剪枝, 然后对剪枝后模型做短时微调 (few epochs)。
3. **量化感知训练 (QAT) 或后训练量化 (PTQ)**: 优先用 QAT (整合到微调中), 否则使用 INT8 PTQ 并校准。
4. **导出与加速**: ONNX -> TensorRT / OpenVINO / CoreML。对不同平台做特定优化 (张量融合、内核替换)。
5. **延迟/内存测试**: 在目标硬件 (Jetson Nano/Orin、Raspberry Pi + NPU、ARM with NPU) 实测 FPS、内存占用、功耗。

## 8. 评估指标与门槛

- 常规: mAP@0.5、mAP@0.5:0.95、Recall、Precision、F1。
- 小目标特别指标: 分别统计不同 bbox 尺度 (small/medium/large) 上的 mAP 与 recall。
- 部署指标: 平均推理延迟、p50/p99 latency、内存占用、能耗 (如果可测)。
- 业务阈值: 例如召回  $\geq 90\%$  (若更关心不漏检), 或 FPS  $\geq 10$  在 Jetson Nano 上。

## 9. 消融实验建议 (必须做)

- 不同蒸馏分量 ablation: 只 logits / 只 feature / logits+feature / +attention, 比较 mAP 与 recall。
- Student attention vs 无 attention: 加入 ECA/SE 带来的增益与延迟开销。
- 剪枝前后对比: 不同剪枝率 (30%/50%/70%) 对精度与速度的影响。
- 量化效果: FP16 vs INT8 (PTQ/QAT) 对精度的影响。
- Teacher 对比: DINO-DETR vs SAM2 (哪个 teacher 更能提升 student), 以及 ConvNeXt vs ViT 作为 teacher backbone 的差别。

## 10. 风险点与缓解策略

---

- Teacher 与 Student 架构差异大 → 蒸馏难度上升。缓解: 使用中间层对齐 (projection heads) 将 teacher feature 映射到 student 的维度再做蒸馏。
- Student 太小学不到 teacher 特征 → 重点蒸馏关键尺度层、用更强的 feature weight。
- 伪标注噪声 → 只用高置信度或引入校准机制 (e.g. soft teacher, confidence smoothing)。

## 11. 实施计划 (简单里程碑)

---

1. 数据清洗与增强 pipeline + 基线训练 (YOLOv11+MobileNet) — 1-2 周。
2. Teacher 微调 (DINO-DETR 或 SAM2 + LoRA) 并输出特征/attention — 2-3 周。
3. 蒸馏训练 Student (多次尝试超参) — 2-4 周。
4. 剪枝+量化+部署优化 — 1-2 周。
5. 消融实验与性能评估 — 2 周。

## 12. 最后 — 可选增强点 (未来方向)

---

- 引入 semi/self-supervised pretraining (MAE, DINOv2) 在无标注田间图像上先预训练 teacher/backbone。
- 在 student head 加轻量性的 DETR-like query head (tiny transformer) 并配合蒸馏测试长期依赖的收益。
- 用任务感知的损失改进 (SIoU/Inner-SIoU) 专门针对小目标回归性能。