

# 第十届“泰迪杯” 数据挖掘挑战赛

## 优秀 作品

作品名称：农田害虫图像识别

荣获奖项：一等并获网宿创新奖

作品单位：山东大学

作品成员：刘士渤 苏倩仪 张建伟

指导老师：栾俊峰

封面为后期添加，原作品没有此页。

# 基于深度学习的农田害虫图像识别

## 摘要

由于农田害虫具有多样性和信息类型的复杂性，通过人工观察统计的传统害虫监测方式已经难以满足现代大规模农业生产对虫害预防工作的需要。使用虫情测报灯可帮助植保人员高效地进行虫情分析，提高测报工作效率和准确率，避免农药的滥用和误用，减少农产品的农药残留，改善农田生态环境。

本文就害虫识别定位问题，提出了创新性的解决方法：将 Faster R-CNN 与 Yolov5 以集成学习的思路相结合，并对错误数据进行了清洗，对数量稀少的样本进行了数据增强。最终取得了优于单一模型的效果。

## 关键字：

目标检测 集成学习 Faster R-CNN Yolov5

# 目录

摘要 .....	1
目录 .....	2
1. 引言 .....	3
1.1 赛题背景.....	3
1.2 问题重述.....	3
1.3 相关工作.....	3
1.4 解决流程.....	4
2. 数据预处理 .....	5
2.1 数据描述.....	5
2.2 数据清洗.....	6
2.3 数据增强.....	7
3. 基于双模型融合的图像检测分类 .....	10
3.1 模型选择.....	10
3.2 模型理论基础.....	10
3.3 模型融合.....	13
4. 实验 .....	14
4.1 实验设置 .....	14
4.2 实验结果 .....	15
5. 总结与分析 .....	22
5.1 优势 .....	22
5.2 不足 .....	22
5.3 后续改进 .....	22
6. 参考文献: .....	23

# 1. 引言

## 1.1 赛题背景

病虫害一直是农业生产中无法避免的问题，据中国统计年鉴，2016年，我国农业生产总值达 5.93 万亿元，占 GDP 的 8%，但由农业病害等灾害造成的直接损失达 0.503 万亿元，占农业生产总值的 8.48%。因此，对农作物进行准确的病害识别并推荐合适的防治措施，创造出能为植物看病的“医生”，可以挽救农作物的生命，减少农药使用量，保证农作物的产量。这就需要收集有害生物信息，并以此为基础进行虫情分析。由于农田害虫具有多样性和信息类型的复杂性，通过人工观察统计的传统害虫监测方式已经难以满足现代大规模农业生产对虫害预防工作的需要。近年来出现的虫情测报灯是虫情信息采集的智能设备，它专为农林业虫情测报而研制，利用光电技术实现自动诱虫、杀虫、分装等功能。可配备风速风向、环境温度湿度、光照等多种传感器接口，在需要时监测环境参数，并可通过 GPRS 上传数据。以监测环境与病虫害之间的关系，通过科技手段生成“数字农业解决方案”。随着中国智慧农业的高速发展，通过科技手段生成“数字农业解决方案”，使用虫情测报灯可帮助植保人员高效地进行虫情分析，提高测报工作效率和准确率，避免农药的滥用和误用，减少农产品的农药残留，改善农田生态环境。

## 1.2 问题重述

1.2.1 使用附件 1 提供图片信息和附件 2 中的标记信息，建立识别害虫位置和种类的模式。

1.2.2 应用问题 1 所建立的模型和算法识别附件 3 中所提及图片中害虫的种类，并确定害虫的位置，并把结果存放在“result2.csv”文件中。

1.2.3 对问题 2 得到的结果进行分析，对每个图像文件中不同种类的害虫数量进行统计，并把统计结果存放在“result3.csv”文件中。

## 1.3 相关工作

目标推荐。在此方面有大量的文献及方法，使用较广泛的方法包括基于超 grouping super-pixels（如：Selective Search<sup>[5]</sup>，CPMC<sup>[6]</sup>，MCG<sup>[7]</sup>）和基于滑动窗口的方法（如：objectness in windows<sup>[8]</sup>，EdgeBoxes<sup>[9]</sup>）。目标推荐方法也经常被独立于检测器而被外部模块使用（如：Selective Search 目标检测<sup>[5]</sup>，R-CNN<sup>[1]</sup>和 Fast R-CNN<sup>[2]</sup>）。

用于目标检测的深度网络。R-CNN<sup>[1]</sup>端到端地训练 CNNs，将候选区域分类成物体类别或背景。许多论文<sup>[10][11][12][13]</sup>已经提出了使用深度网络来预测对象边界框。

集成学习。其核心思想是通过训练若干个弱学习器，使用特定的结合策

略，以构建一个强学习器。构建方法可以分为两类：平行算法（如：bagging<sup>[14]</sup>）和顺序方法（如：boosting<sup>[15]</sup>）。

### 1.4 解决流程

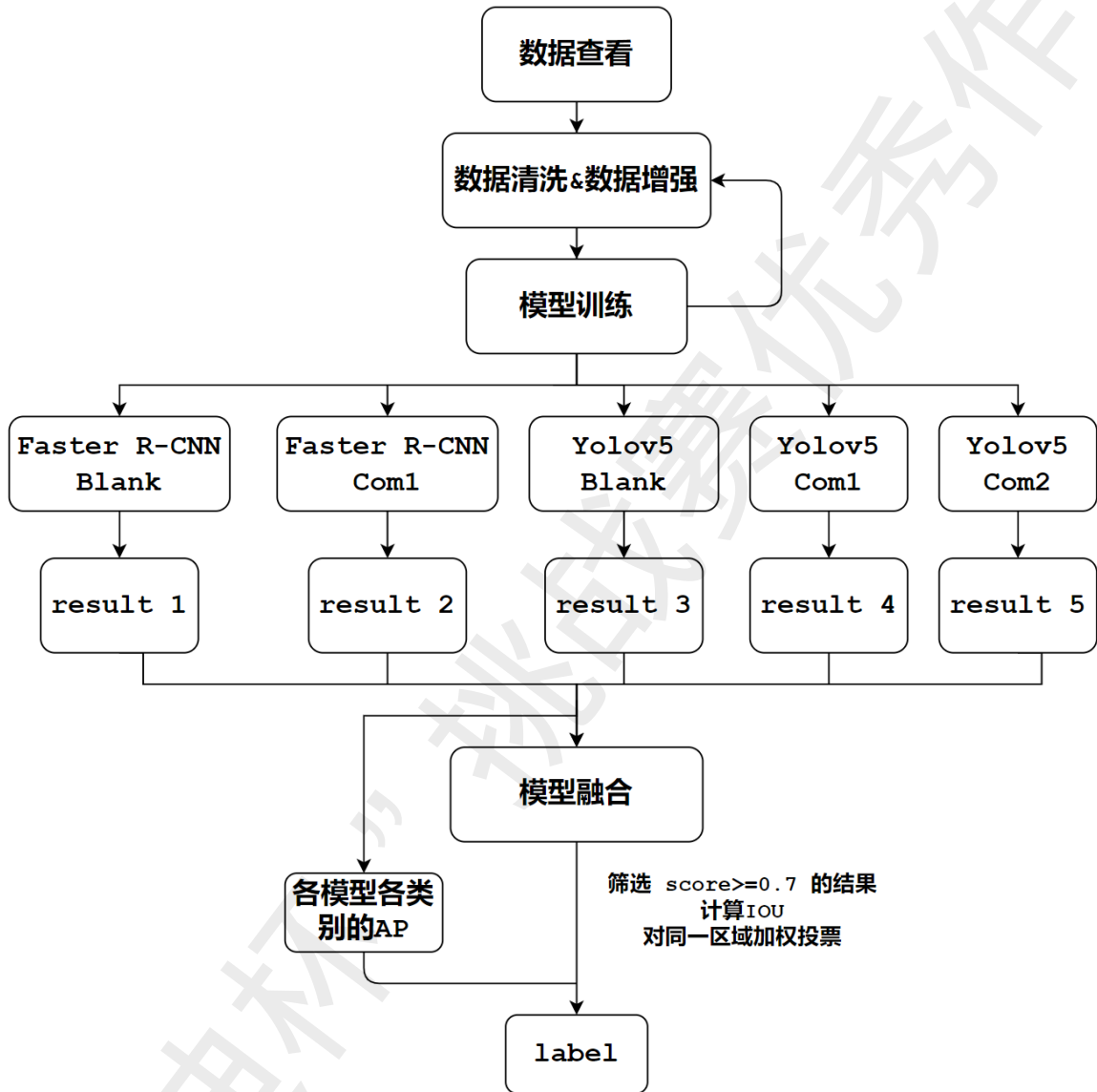


图 16: 流程图

## 2. 数据预处理

### 2.1 数据描述

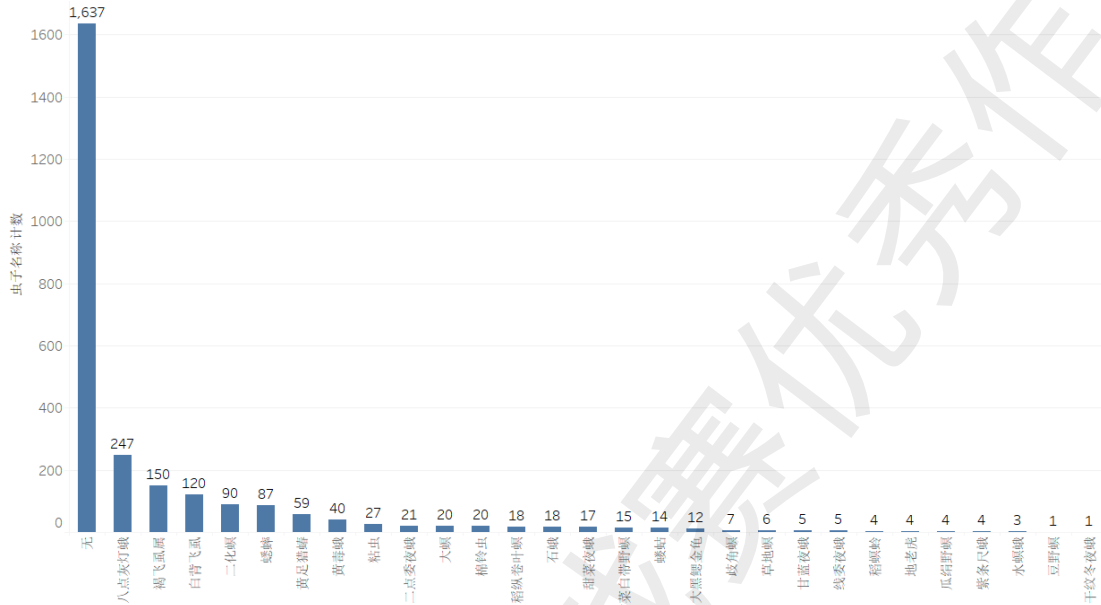


图 1: 训练集中的害虫种类及其数量统计

属性	数据类型
序号	整数
文件名	字符串
虫子编号	整数
虫子名称	字符串
中心点 x 坐标	整数
中心点 y 坐标	整数
左上角 x 坐标	整数
左上角 y 坐标	整数
右下角 x 坐标	整数
右下角 y 坐标	整数

表格 1: 图片虫子位置详情表的属性及其数据类型

在附件 1 中共有 3015 张图片。在附件 2 ‘图片虫子位置详情表.csv’ 中，每条记录包含 10 个相关属性，如表格 1 所示。其中，若虫子编号为 0 则表示没有害虫，非零数字与虫子名称一一对应。共有 28 个害虫种类。在附件 3 ‘无位置信息的图片汇总表.csv’ 中，说明了需要进行测试的 802 张图片。

通过分析附件 1、附件 2、附件 3，我们发现，在全部的 3015 张照片中，除去需要测试的 802 张图片，仅有 576 张图片包含被标记的害虫，剩余的 1637 张图片均没有包含标记害虫，有效的数据非常少。

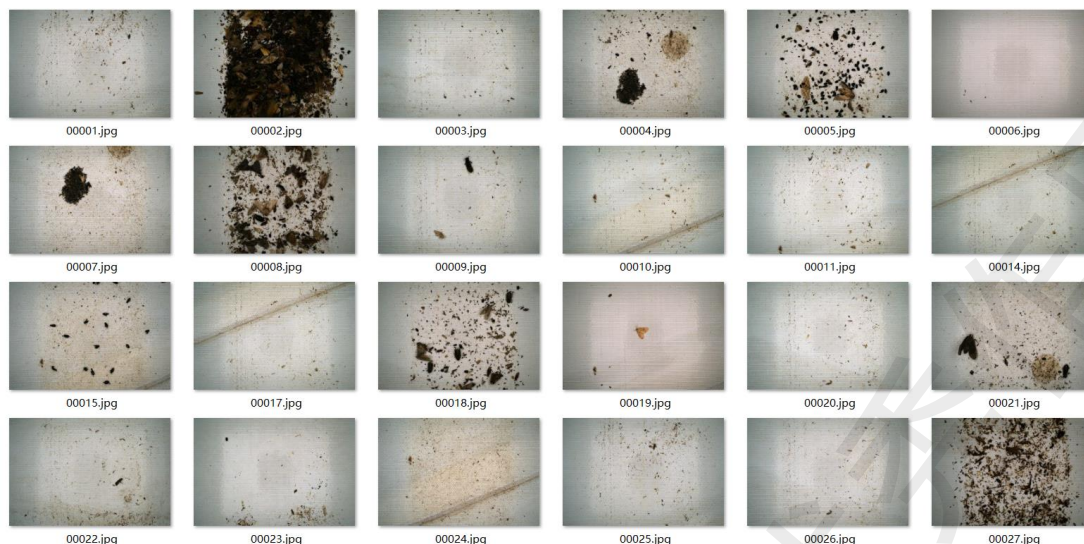


图 2：图片数据集部分展示

## 2.2 数据清洗

根据标记文件将图像进行切割，通过观察发现，标记中存在部分错误标签，我们对此进行了初步删除处理。



图 3：蟋蟀样本

图 3 是被标记为蟋蟀的样本，可以明显观察到(b)、(d)、(f)、(g)、(j)、(k)、(l)被正确标记，而(a)、(c)、(e)、(i)均被错误归类为蟋蟀，其中(a)、(c)、(e)十分杂乱，很大程度上会影响模型的训练效果。



## 2.3 数据增强

### 2.3.1 原始数据分布

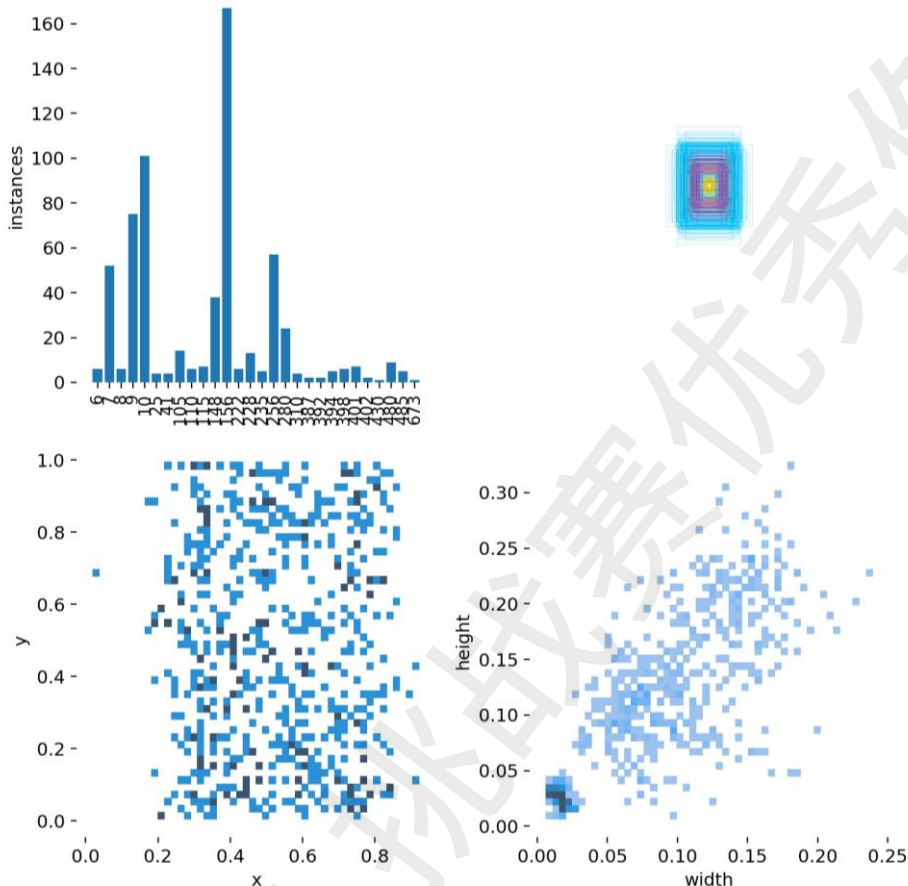


图 4: 数据增强前的数据分布情况。从左到右, 从上到下分别是: 害虫样例数分布、候选框叠加、害虫中心点坐标分布、候选框宽高分布

通过害虫样例数分布可以看出, 大部分害虫的样例都很少, 只有 7 类害虫样例数超过了 20, 其余 21 类害虫样例数均不足 20。这就证明了数据增强的必要性。

通过候选框宽高分布可以看出, 比较多的害虫比较小, 大小约为  $180 \times 160$  pixels。如果候选框设定较大的话, 可能不利于这一部分害虫的识别。

### 2.3.2 图像旋转, 翻转

由于数据集给出的图像中, 被标记的样本数量十分稀少, 为了获得更多的数据集, 旋转, 翻转是过渡到图像分析的关键步骤, 因此我们对标记十分稀少的图像进行旋转, 翻转等操作, 以增加它们在总样本中的比例。

利用 python 的 Opencv-Python 库将原图像分别随机生成旋转、翻转后的图片, 并对添加噪声等可以较为明显的增强数据多样性, 如图 5 所示。



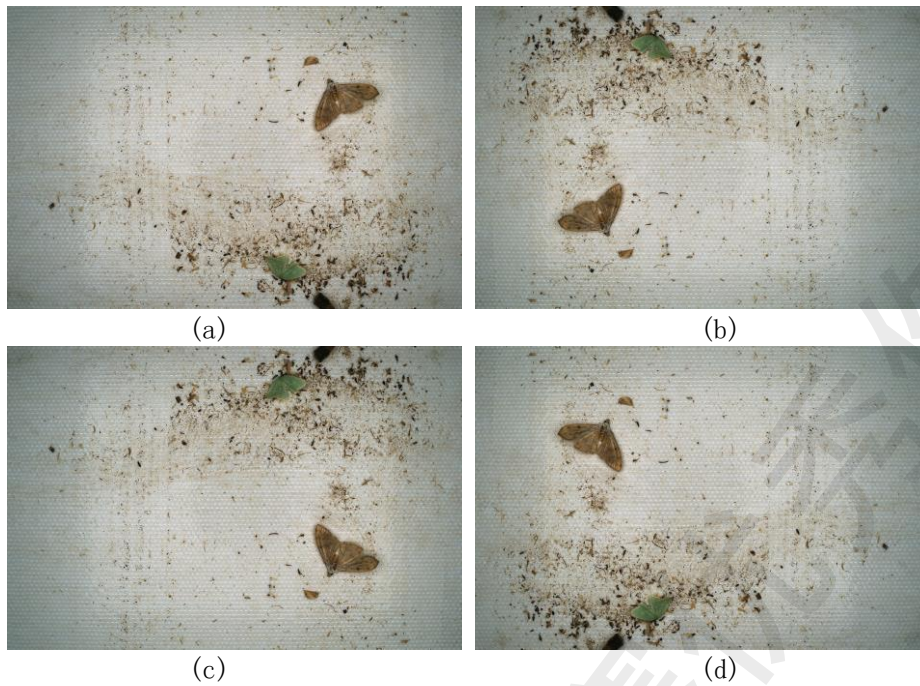


图 5: (a)是原图, 将(a)旋转 180° 得到(b), 上下翻转得到(c), 左右翻转得到(d)

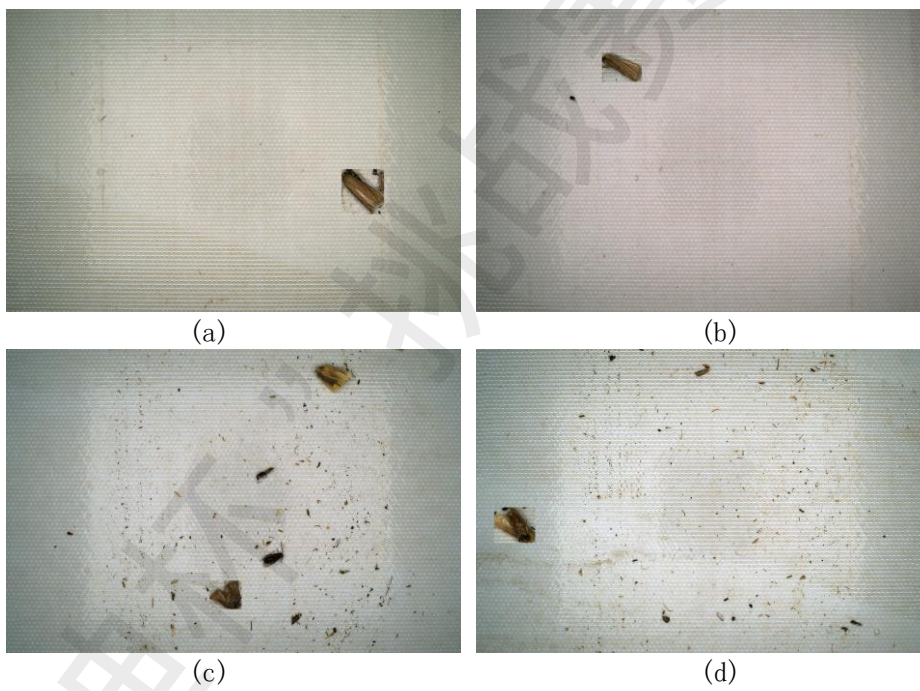


图 6: 合成图片

### 2.3.3 图像裁剪与合成

由于有大量无害虫标注的图片, 为了使模型能更好的区分害虫和背景, 我们将使用标记的害虫图像与背景合成作为输入, 以使我们的数据更好的区分害虫和背景。如图 6 所示。

### 2.3.4 数据增强后的数据分布

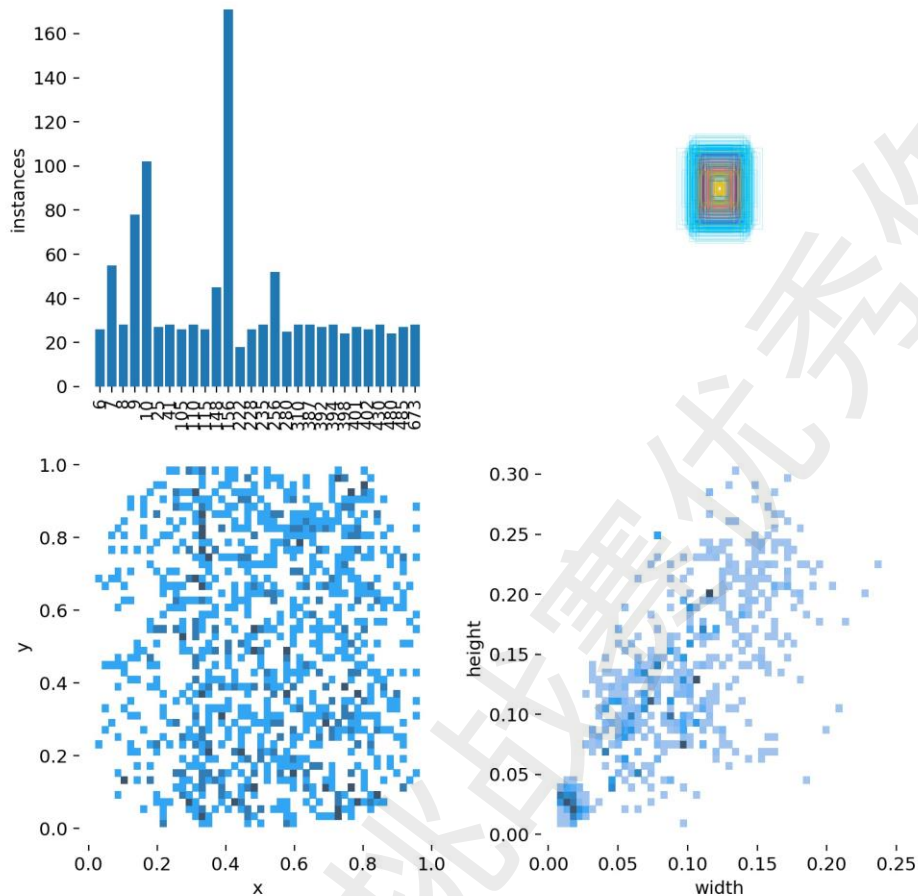


图 7：数据增强后的数据分布情况。从左到右，从上到下分别是：害虫样例数分布、候选框叠加、害虫中心点坐标分布、候选框宽高分布

通过害虫样例数分布可以看出，经过数据增强，大部分害虫的样例都被补齐超过了 20。

## 3. 基于双模型融合的图片检测分类

### 3.1 模型选择

YOLOv5 在视觉人工智能视觉领域有着广泛应用。Faster R-CNN 是典型的 two-stage 检测算法，在 ImageNet 检测任务、ImageNet 定位任务、COCO 检测任务、COCO 分割任务中表现出色，虽然速度较慢，但能够有效提高目标检测精度。

因此，我们选择 Yolov5 和 Faster R-CNN 作为我们的模型基础。

### 3.2 模型理论基础

#### 3.2.1 Faster R-CNN

目标检测的发展很大程度上归功于候选区域方法和基于区域的卷积神经网络 (R-CNNs)<sup>[1]</sup>。Fast R-CNN<sup>[2]</sup>在忽略候选区域的时间开销的情况下，使用非常深的网络实现了接近实时的速度。Faster R-CNN<sup>[3]</sup>使用深度卷积神经网络计算候选区域，使得候选区域的计算几乎是无成本的，因为检测网络的计算量是给定的。为此引入了一个完全卷积网络——区域建议网络 (RPN)，通过在测试时共享卷积，使得计算方案的边际成本很小，并且可以针对生成检测建议的任务进行端到端的训练。

Faster R-CNN 由两个模块组成。第一个模块是产生候选框的深度全卷积神经网络，第二个模块是使用候选框的 Fast R-CNN 检测器。如图 8 所示，Faster R-CNN 是一个单个的、统一的目标检测网络，RPN 模块充当这个网络的“注意力”。

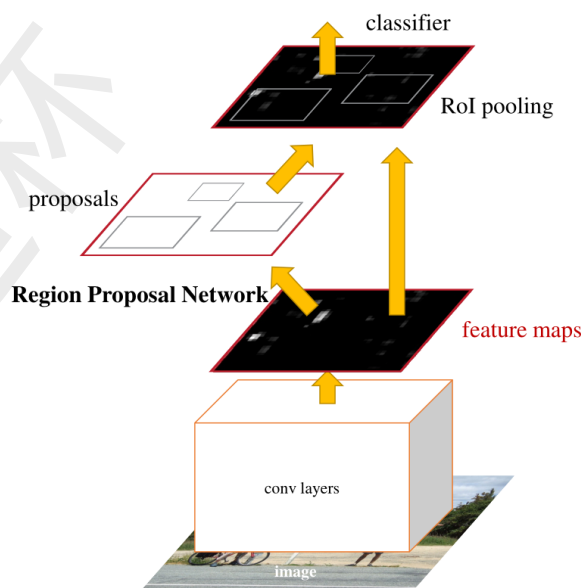


图 8: Faster R-CNN 结构



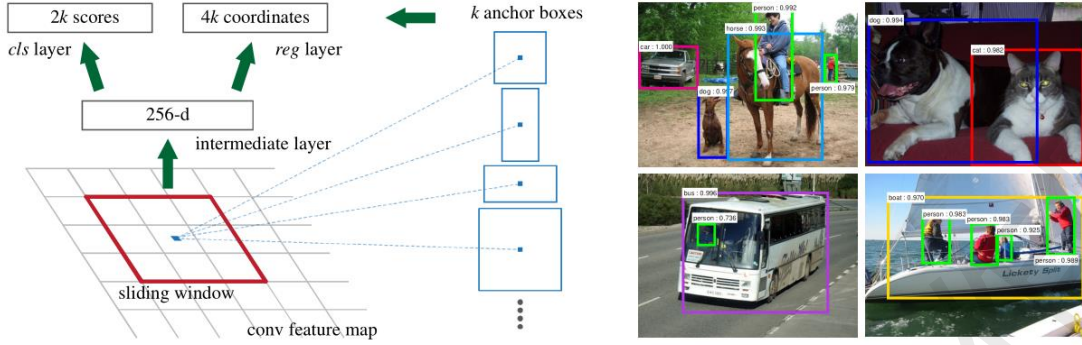


图 9：左图：RPN。右图：PASCAL VOC 2007 测试中使用 RPN 的检测示例，可以检测各种比例和宽高比的物体。

RPN 是一个全卷积网络，以任意大小的图像作为输入，输出一组候选框，每个候选框都有一个目标得分。为了生成候选框，在最后共享的卷积层输出的卷积特征图上滑动一个小网络。这个网络将输入卷积特征图的  $3 \times 3$  空间窗口作为输入。每个滑动窗口被映射到一个低维特征，再被输入到两个全连接层（一个边界框回归层 reg 和一个边界框分类层 cls），如图 9（左）所示。因为小网络以滑动窗口方式运行，所以所有空间位置共享全连接层。这种架构通过一个  $3 \times 3$  卷积层、和两个  $1 \times 1$  卷积层（分别用于 reg 和 cls）实现。

为了训练 RPN，为每个 anchor 分配一个二元分类标签（是目标或不是目标）。给这两种 anchor 分配一个正样本标签：具有与真实边界框的重叠最高 IOU 的 anchor，或者具有与真实边界框的重叠超过  $0.7 \text{IOU}$  的 anchor。单个真实边界框可以为多个 anchor 分配正标签。通常第二个条件足以确定正样本；但我们仍然采用第一个条件，因为在极少数情况下，第二个条件可能找不到正样本。对于所有的真实边界框，如果一个 anchor 的 IOU 低于  $0.3$ ，我们给非正样本的 anchor 分配一个负标签。既不是正样本标签也不是负样本标签的 anchor 对训练目标函数没有作用。根据这些定义，最小化多任务损失的目标函数，定义图像的损失函数为：

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i L_{reg}(t_i, t_i^*)$$

其中， $i$  是小批量数据中 anchor 的索引， $p_i$  是 anchor 为目标的预测概率。若 anchor  $i$  为正样本，则  $p_i^*$  真实标签为 1，否则为 0。 $t_i$  是表示预测边界框的 4 个参数化坐标组成的向量， $t_i^*$  是与正标签 anchor 相关联的真实边界框坐标组成的向量。分类损失  $L_{cls}$  是两个类别上（是目标或不是目标）的对数损失。回归损失  $L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$ ，其中  $R$  是  $smoothL_1$ 。

对于边界框回归，采用<sup>[1]</sup>中的 4 个坐标参数化：

$$\begin{aligned} t_x &= (x - x_a) / w_a, t_y = (y - y_a) / h_a, \\ t_w &= \log(w / w_a), t_h = \log(h / h_a), \\ t_x^* &= (x^* - x_a) / w_a, t_y^* = (y^* - y_a) / h_a, \\ t_w^* &= \log(w^* / w_a), t_h^* = \log(h^* / h_a) \end{aligned}$$

其中， $x, y, w, h$  表示边界框的中心坐标及其宽和高。 $x, x_a, x^*$  分别表示预测边界框、anchor 框和真实边界框。 $y, w, h$  类似。

使用四步训练算法，通过交替优化使得 RPN 和 Fast R-CNN 共享卷积层。首先，使用 ImageNet 预训练模型对 RPN 进行初始化，并针对任务进行端对端的微调。然后，Fast R-CNN 使用 RPN 生成的候选框训练一个由 ImageNet 预训练模型进行初始化的检测网络。此时两个网络不共享卷积层。接着，使用检测网络来初始化 RPN 训练，固定共享卷积层，只对 RPN 特有的层进行微调。此时两个网络共享卷积层。最后，固定共享卷积层，对 Fast R-CNN 特有的层进行微调。因此，两个网络共享相同的卷积层，并形成统一的网络。类似的交替训练可以运行更多的迭代次数。

### 3.2.2 Yolov5s

Yolov5 是一种单阶段目标检测算法，它在 Yolov4 的基础上添加了一些新的改进思路，使速度与精度都得到了极大的性能提升。

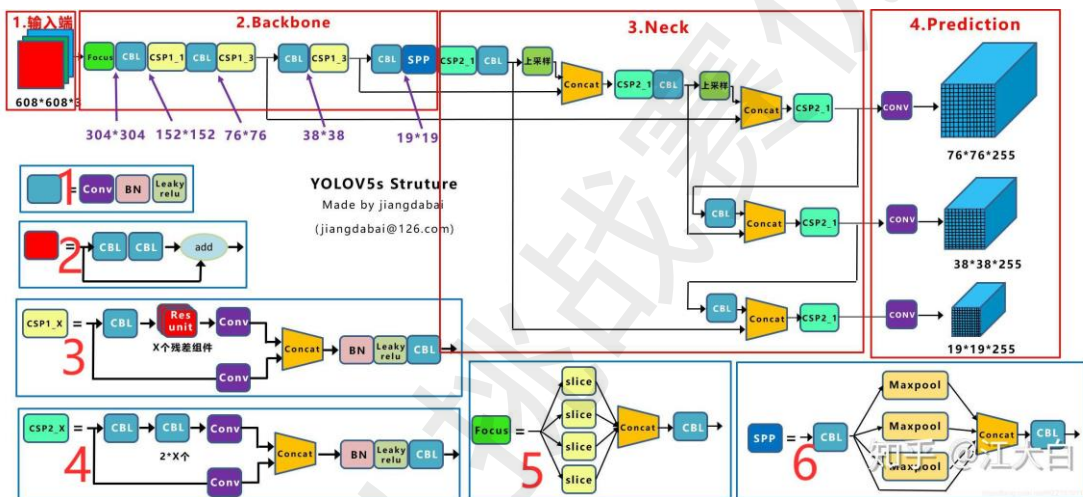


图 10: Yolov5 目标检测算法的整体框图。

如图 10 所示，对于一个目标检测算法而言，通常可以将其划分为 4 个通用的模块，具体包括：输入端、基准网络、Neck 网络与 Head 输出端，对应于上图中的 4 个红色模块。

Yolov5 基础组件主要有六部分：CBL-CBL 模块由 Conv+BN+Leaky\_relu 激活函数组成，如上图中的模块 1 所示。Res unit-借鉴 ResNet 网络中的残差结构，用来构建深层网络，CBM 是残差模块中的子模块，如上图中的模块 2 所示。CSP1\_X-借鉴 CSPNet 网络结构，该模块由 CBL 模块、Res unit 模块以及卷积层、Concat 组成而成，如上图中的模块 3 所示。CSP2\_X-借鉴 CSPNet 网络结构，该模块由卷积层和 X 个 Res unit 模块 Concat 组成而成，如上图中的模块 4 所示。Focus-如上图中的模块 5 所示，Focus 结构首先将多个 slice 结果 Concat 起来，然后将其送入 CBL 模块中。SPP-采用 1x1、5x5、9x9 和 13x13 的最大池化方式，进行多尺度特征融合，如上图中的模块 6 所示。

### 3.3 模型融合

由于使用 Faster R-CNN 模型和 Yolov5 模型的效果不理想，对此我们提出了基于 Faster R-CNN 模型和 Yolov5 模型的双模型融合图像分类模型，其基本思路为：

1. 根据不同模型在验证集上不同类别的表现赋予相应的模型权重。
2. 将图像放入五模型中分别进行检测，得到各个模型的结果
3. 对预测 IOU 较大的 bounding boxes 进行分析，若预测类别不同，结合模型在类别权重和预测置信度对不同模型预测结果赋予不同的“票数”，选择票数最多的类为最终结果，并取投票正确的结果的边框平均值为最终结果。

## 4. 实验

### 4.1 实验设置

#### 4.1.1 Faster R-CNN

	数据集设置	参数设置
对照组	使用 576 张有 label 的照片，训练集和验证集比例为 1: 1	rpn_negative_overlap=0.3 rpn_positive_overlap=0.7 rpn_fg_fraction=0.5
实验组	使用 576 张有 label 的照片，以及数据增强 <sup>1</sup> 后的 672 张图片，训练集和验证集比例为 1: 1	rpn_train_nms_thresh=0.7 rpn_test_nms_thresh=0.7 iter=10000, batch_size=256

表格 2: Faster R-CNN 实验设置

实验组和对照组旨在对比得出数据增强<sup>1</sup>对模型训练效果的影响。

#### 4.1.2 Yolov5s

	数据集设置	超参设置	运行设置
对照组	使用 576 张有 label 的照片，根据梯度划分训练集和验证集 <sup>2</sup>	flipup=0 fliplr=0	epoch=100 batch_size=32 img_size=640
实验组 1	使用 576 张有 label 的照片，以及数据增强 <sup>1</sup> 后的 672 张图片，根据梯度划分训练集和验证集 <sup>2</sup>	mosaic=1 mixup=0 copy_paste=0	
实验组 2	使用 576 张有 label 的照片，根据梯度划分训练集和验证集 <sup>2</sup>	flipup=0.5 fliplr=0.5 mosaic=0.5 mixup=0.5 copy_paste=0.5	

表格 3: Yolov5s 实验设置

对照组和实验组 1 旨在对比得出数据增强<sup>1</sup>对模型训练效果的影响；对照组和实验组 2 旨在对比得出通过调整超参的数据增强对模型训练效果的影响。

\*注 1: 这里的数据增强指论文的 2.3 部分

\*注 2: 根据梯度划分是指: 对于样本数  $\in [1, 10]$  的害虫, 验证集和训练集使用相同的照片; 对于样本数  $\in (10, 20]$  的害虫, 以 1:1 的比例划分验证集和训练集; 对于样本数  $\in (20, 50]$  的害虫, 以 3:7 的比例划分验证集和训练集; 对于样本数  $> 50$  的害虫, 以 1:4 的比例划分训练集和验证集。这样划分的目的在于防止样本少的害虫的验证集过少。



## 4.2 实验结果

### 4.2.1 各类害虫指标

虫子编号/ 虫子名称	模型	组别	Precis ion	Recal 1	AP	虫子编号/ 虫子名称	模型	组别	Precis ion	Reca 11	AP
6/ 大螟	faster R-CNN	对照组	<b>0.761</b>	<b>0.8</b>	<b>0.784</b>	7/ 二化螟	faster R-CNN	对照组	<b>0.469</b>	<b>0.855</b>	<b>0.705</b>
		实验组	0.785	0.55	0.513			实验组	0.464	0.733	0.532
	Yolov5	对照组	0	0	0		对照组	0.393	0.511	0.335	
		实验组 1	0.8	0.2	0.16		实验组 1	0.733	0.489	0.388	
		实验组 2	0	0	0		实验组 2	0.338	0.533	0.288	
	faster R-CNN+ Yolov5	-	1	0.45	0.45		faster R-CNN+ Yolov5	-	0.549	0.833	0.688
8/ 稻纵卷 叶螟	faster R-CNN	对照组	<b>0.243</b>	<b>1</b>	<b>0.915</b>	9/ 白背飞 虱	faster R-CNN	对照组	0	0	0
		实验组	0.928	0.722	0.714			实验组	0	0	0
	Yolov5	对照组	0	0	0		对照组	0	0	0	
		实验组 1	0.818	0.5	0.429		实验组 1	<b>1</b>	<b>0.008</b>	<b>0.008</b>	
		实验组 2	0	0	0		实验组 2	0	0	0	
	faster R-CNN+ Yolov5	-	0.381	0.889	0.412		faster R-CNN+ Yolov5	-	0	0	0
10/ 褐飞虱 属	faster R-CNN	对照组	0.319	0.1	0.046	25/ 地老虎	faster R-CNN	对照组	0	0	0
		实验组	0.25	0.006	0.003			实验组	<b>0.25</b>	<b>1</b>	<b>1</b>
	Yolov5	对照组	<b>0.706</b>	<b>0.08</b>	<b>0.068</b>		对照组	0	0	0	
		实验组 1	0.75	0.1	0.087		实验组 1	0.75	0.1	0.087	
		实验组 2	0	0	0		实验组 2	0	0	0	
	faster R-CNN+ Yolov5	-	0.833	0.033	0.031		faster R-CNN+ Yolov5	-	0.6	0.75	0.65
41/ 蝼蛄	faster R-CNN	对照组	0.857	0.428	0.398	105/ 粘虫	faster R-CNN	对照组	<b>0.625</b>	<b>0.555</b>	<b>0.494</b>
		实验组	0.857	0.428	0.429			实验组	1	0.370	0.370
	Yolov5	对照组	0	0	0		对照组	0	0	0	
		实验组 1	0.8	0.571	0.533		实验组 1	0.467	0.259	0.207	

		实验组 2	0	0	0			实验组 2	0	0	0	
	faster R-CNN+ Yolov5	-	<b>0.889</b>	<b>0.5</b>	<b>0.571</b>			faster R-CNN+ Yolov5	-	0.75	0.222 222	0.222 222
	faster R-CNN	对照组	0	0	0			faster R-CNN	对照组	1	0.058	0.058
		实验组	<b>0.333</b>	<b>0.333</b>	<b>0.333</b>				实验组	<b>0.236</b>	<b>0.764</b>	<b>0.428</b>
110/ 草地螟		对照组	0	0	0	115/ 甜菜夜蛾			对照组	0	0	0
	Yolov5	实验组 1	0.333	0.333	0.111			Yolov5	实验组 1	0.444	0.235	0.156
		实验组 2	0	0	0				实验组 2	0	0	0
	faster R-CNN+ Yolov5	-	1	0.333	0.333			faster R-CNN+ Yolov5	-	0.409	0.529	0.285
148/ 黄足猎蝽	faster R-CNN	对照组	<b>0.511</b>	<b>0.728</b>	<b>0.709</b>	156/ 八点灰灯蛾		faster R-CNN	对照组	0.75	0.975	0.951
		实验组	0.636	0.355	0.320				实验组	0.736	0.943	0.912
		对照组	0.426	0.339	0.234				对照组	0.690	0.939	0.880
	Yolov5	实验组 1	0.711	0.458	0.358			Yolov5	实验组 1	0.797	0.907	0.734
		实验组 2	1	0.034	0.034				实验组 2	0.648	0.911	0.876
	faster R-CNN+ Yolov5	-	0.698	0.729	0.654			faster R-CNN+ Yolov5	-	<b>0.706</b>	<b>0.976</b>	<b>1.103</b>
222/ 棉铃虫	faster R-CNN	对照组	0.370	1	0.963	228/ 二点委夜蛾		faster R-CNN	对照组	0.281	0.857	0.424
		实验组	0.512	1	0.958				实验组	<b>0.368</b>	<b>0.666</b>	<b>0.432</b>
		对照组	0	0	0				对照组	0	0	0
	Yolov5	实验组 1	0.2	0.05	0.017			Yolov5	实验组 1	0.474	0.429	0.303
		实验组 2	0	0	0				实验组 2	0	0	0
	faster R-CNN+ Yolov5	-	<b>0.467</b>	<b>1</b>	<b>1.032</b>			faster R-CNN+ Yolov5	-	0.462	0.571	0.395
235/ 甘蓝夜蛾	faster R-CNN	对照组	0	0	0	256/ 蟋蟀		faster R-CNN	对照组	<b>0.561</b>	<b>0.862</b>	<b>0.721</b>
		实验组	<b>1</b>	<b>0.8</b>	<b>0.8</b>				实验组	0.285	0.643	0.527
		对照组	0	0	0				对照组	0.560	0.540	0.450
	Yolov5	实验组 1	0.231	0.6	0.2			Yolov5	实验组 1	0.615	0.459	0.315
		实验组 2	0	0	0				实验组 2	0.338	0.287	0.169
	faster	-	0.8	0.8	0.72			faster	-	0.462	0.827	0.641

	R-CNN+ Yolov5					R-CNN+ Yolov5				
280/ 黄毒蛾	faster R-CNN	对照组	0.639	0.975	0.937	faster R-CNN	对照组	0	0	0
		实验组	<b>0.612</b>	<b>0.95</b>	<b>0.893</b>		实验组	0.4	0.5	0.500
	Yolov5	对照组	0.763	0.725	0.700	Yolov5	对照组	0	0	0
		实验组 1	0.8	0.6	0.545		实验组 1	0.667	0.5	0.417
		实验组 2	0.857	0.6	0.565		实验组 2	0	0	0
	faster R-CNN+ Yolov5	-	<b>0.661</b>	<b>0.95</b>	<b>0.849</b>	faster R-CNN+ Yolov5	-	<b>0.667</b>	<b>0.5</b>	<b>0.5</b>
387/ 紫条尺 蛾	faster R-CNN	对照组	0	0	0	faster R-CNN	对照组	0	0	0
		实验组	1	0.25	0.250		实验组	0.75	1	1
	Yolov5	对照组	0	0	0	Yolov5	对照组	0	0	0
		实验组 1	1	0.25	0.25		实验组 1	0.333	0.333	0.167
		实验组 2	0	0	0		实验组 2	0	0	0
	faster R-CNN+ Yolov5	-	<b>1</b>	<b>0.25</b>	<b>0.25</b>	faster R-CNN+ Yolov5	-	0.667	0.667	0.667
394/ 线委夜 蛾	faster R-CNN	对照组	0	0	0	faster R-CNN	对照组	0	0	0
		实验组	1	1	1		实验组	<b>0.8</b>	<b>0.266</b>	<b>0.267</b>
	Yolov5	对照组	0	0	0	Yolov5	对照组	0	0	0
		实验组 1	1	0.6	0.6		实验组 1	0.6	0.2	0.173
		实验组 2	0	0	0		实验组 2	0	0	0
	faster R-CNN+ Yolov5	-	1	0.6	0.6	faster R-CNN+ Yolov5	-	1	0.133	0.133
401/ 歧角螟	faster R-CNN	对照组	0	0	0	faster R-CNN	对照组	0	0	0
		实验组	<b>0.545</b>	<b>0.857</b>	<b>0.816</b>		实验组	0.5	0.5	0.5
	Yolov5	对照组	0	0	0	Yolov5	对照组	0	0	0
		实验组 1	0.6	0.429	0.314		实验组 1	1	0.5	0.5
		实验组 2	0	0	0		实验组 2	0	0	0
	faster R-CNN+ Yolov5	-	0.667	0.286	0.286	faster R-CNN+ Yolov5	-	<b>1</b>	<b>0.5</b>	<b>0.5</b>

430/ 豆野螟	faster R-CNN	对照组	0	0	0	480/ 石蛾	faster R-CNN	对照组	<b>0.578</b>	<b>0.611</b>	<b>0.553</b>
		实验组	1	1	1			实验组	0.8	0.444	0.438
	Yolov5	对照组	0	0	0			对照组	0	0	0
		实验组 1	1	1	1			实验组 1	0.545	0.333	0.237
		实验组 2	0	0	0			实验组 2	0	0	0
		faster R-CNN+ Yolov5	-	1	1		1		faster R-CNN+ Yolov5	-	1
485/ 大黑鳃金龟	faster R-CNN	对照组	0.857	0.5	0.448	673/ 干纹冬夜蛾	faster R-CNN	对照组	0	0	0
		实验组	<b>0.421</b>	<b>0.666</b>	<b>0.623</b>			实验组	<b>1</b>	<b>1</b>	<b>1</b>
	Yolov5	对照组	0	0	0			对照组	0	0	0
		实验组 1	0.75	0.5	0.427			实验组 1	0	0	0
		实验组 2	0	0	0			实验组 2	0	0	0
		faster R-CNN+ Yolov5	-	0.667 7	0.5		0.432		faster R-CNN+ Yolov5	-	0

表格 4: 6 个模型在 28 种害虫类别上的 precision、recall、AP。

模型	组别	Precision	Recall	mAP
Faster R-CNN	对照组	0.55803	0.22628	0.31547
	实验组	0.54163	0.20821	0.59138
Yolov5	对照组	0.60407	0.14533	0.09528
	实验组 1	0.70912	0.16980	0.31127
	实验组 2	0.7539	0.11981	0.06899
Faster R-CNN +Yolov5	-	<b>0.62128</b>	<b>0.24398</b>	<b>0.49466</b>

表格 5: 6 个模型整体的 precision、recall、mAP。

从表格 4 可以看出总体上 Faster R-CNN 总体上相较于 Yolov5 要好一些, 但是 Yolov5 在标记样本数量较多的类别上表现较好, 数据增强后, 标记数目较少的类别能被更好的识别, 所以使用模型融合以结合不同模型的优势。

从表格 5 可以看出, 虽然 Yolov5 在 precision 表现不错, 但是 recall 过低, 原因在于它只预测了标记较多的类别, 没有很好的识别标记较少的类别; 融合模型在 Precision 和 Recall 上都有不错的表现。

## 4.2.2 YOLOv5s

混淆矩阵的理想情况是所有的类别都被 100% 正确识别、没有 False Positive 和 False Negative，即对角线上全部为 1，其余地方都是 0。

如图 11 所示，对照组的对角线上大都是 0 和很小的值，大的数值集中在底部的一行，即 False Negative，说明模型没能识别出本应该识别出的害虫。实验组 1 的较大值集中在了对角线上，底部的一行相较于对照组有了很大的减少，说明数据增强<sup>1</sup>取得了不错的效果。实验组 2 与对照组差别不大：对角线上大都是 0 和很小的值，较大值集中在了底部的一行，说明通过调整超参的数据增强效果平庸。

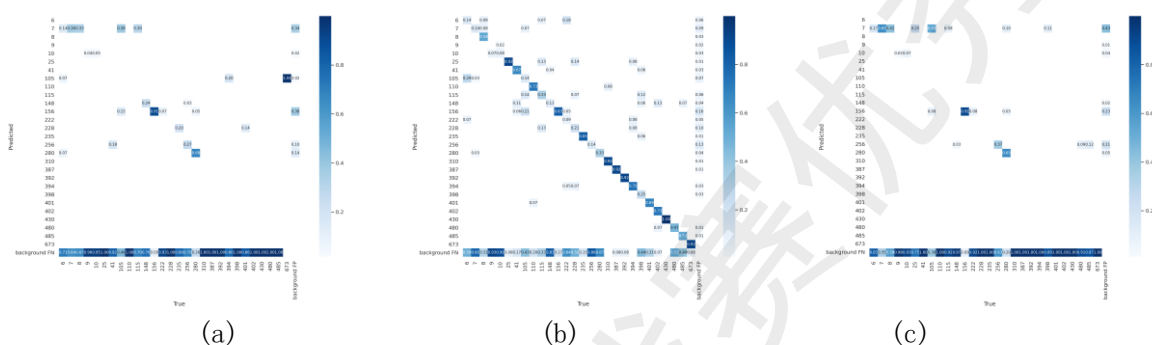


图 11: confusion\_matrix。 (a) 是对照组， (b) 是实验组 1， (c) 是实验组 2。

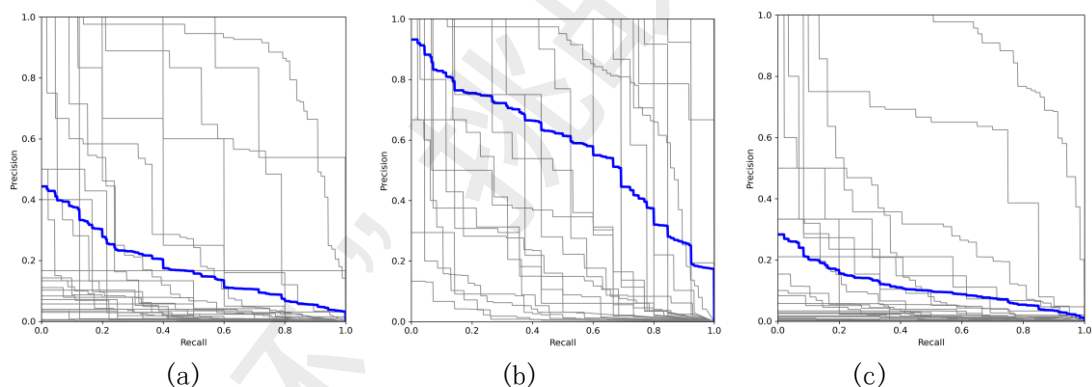


图 12: PR\_curve。 (a) 是对照组， (b) 是实验组 1， (c) 是实验组 2。

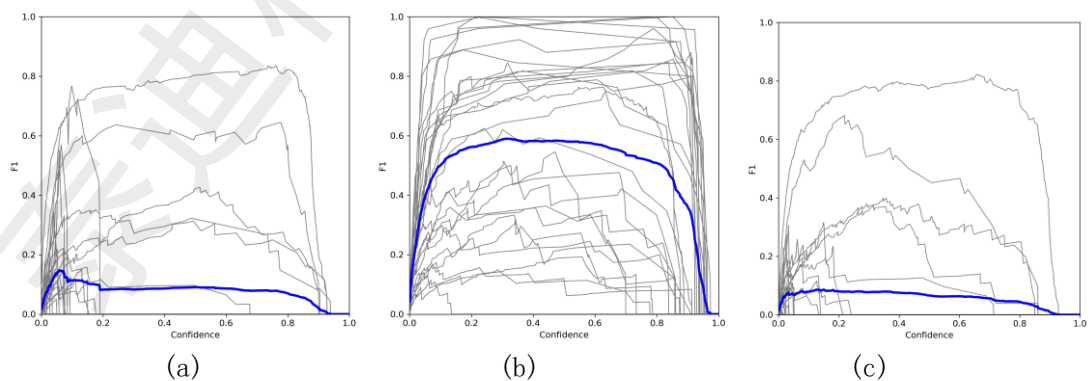
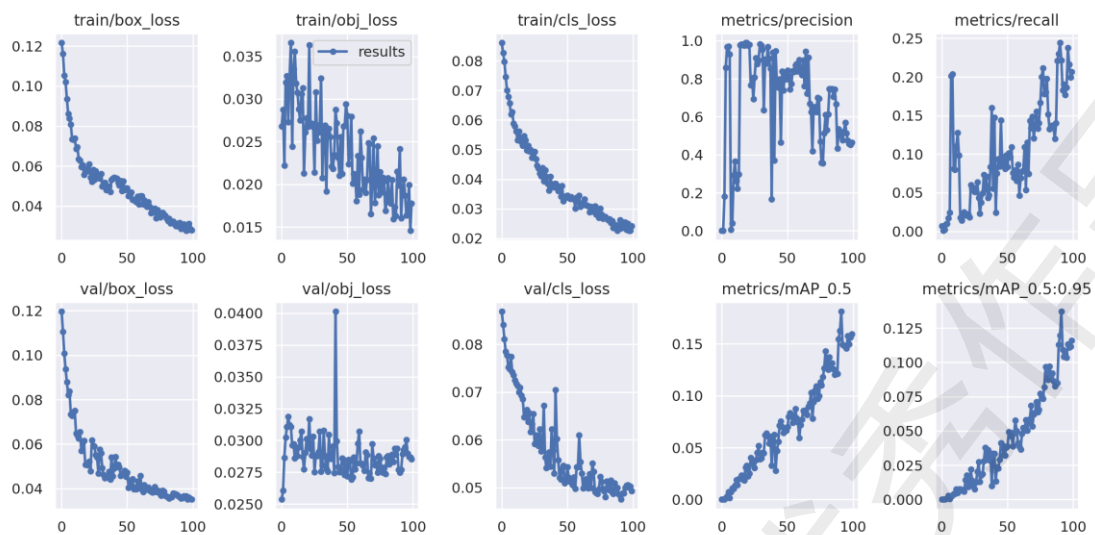
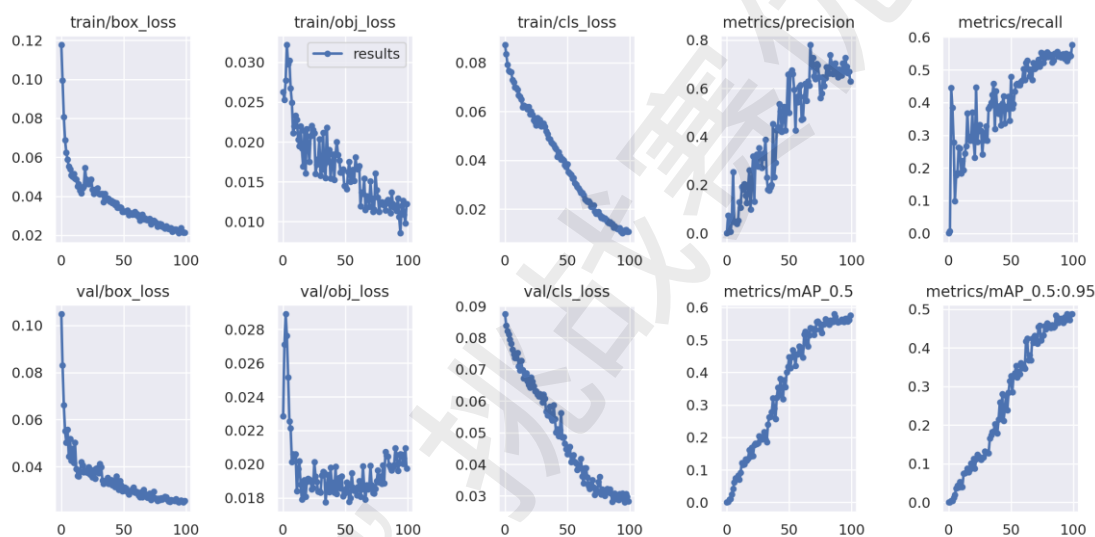


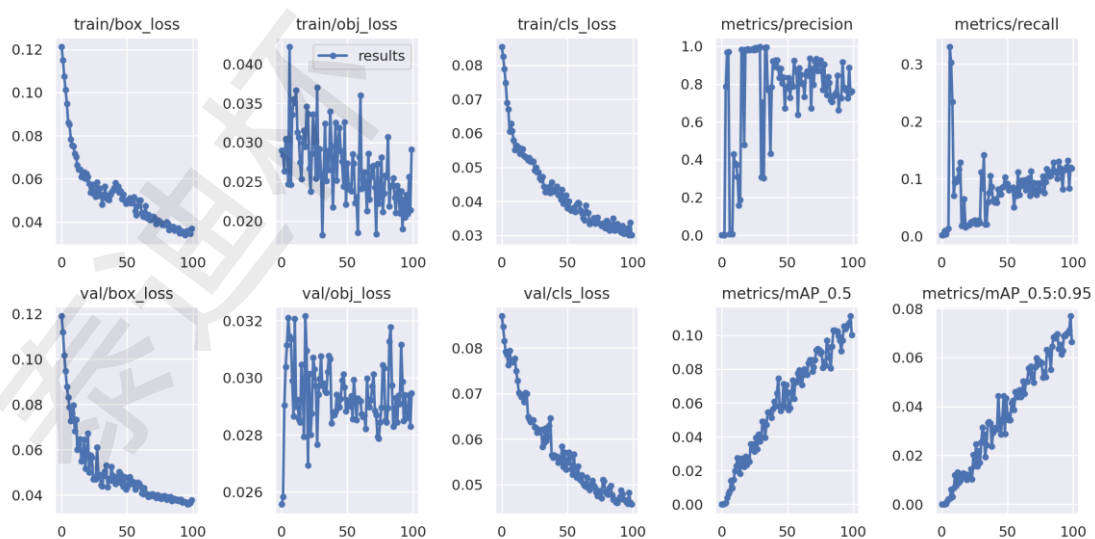
图 13: F1\_curve。 (a) 是对照组， (b) 是实验组 1， (c) 是实验组 2。



(a)



(b)



(c)

图 14: detial。(a)是对照组，(b)是实验组 1，(c)是实验组 2。

对于 PR 曲线，平衡点(BEP)是  $P=R$  时的取值，平衡点越大，模型越好，反映在图上就是直线  $y=x$  与 PR 曲线的交点越靠近右上角，模型越好。如图 12 所示，BEP: 实验组 1 > 对照组 > 实验组 2。

$F1=2PR/(P+R)$ ，从另一个角度展现了 Precision 和 Recall 的关系，从图 13 可以看出，实验组 1 优于对照组优于实验组 2。

图 14 的折线图展示了训练和验证过程中的各类 loss: box\_loss、object\_loss、class\_loss，以及 precision、recall、mAP、mAP\_0.5:0.95。其中各类 loss 总体都呈下降趋势且最终趋于稳定，但 object\_loss 有较大幅度的震荡。mAP 总体呈上升趋势，实验组 1 的 mAP 在 epoch=100 时趋于稳定，而对照组和实验组 2 的 mAP 在 epoch=100 时还有继续上升的趋势，可能再多训练一些 epoch 会有更大的 mAP；但就 epoch=100 而言，mAP: 实验组 1 > 对照组 > 实验组 2。

综合各类指标我们可以得出结论，在 yolov5 的三组实验中，实验组 1 优于对照组优于实验组 2。

### 4.2.3 部分结果展示

Yolov5:

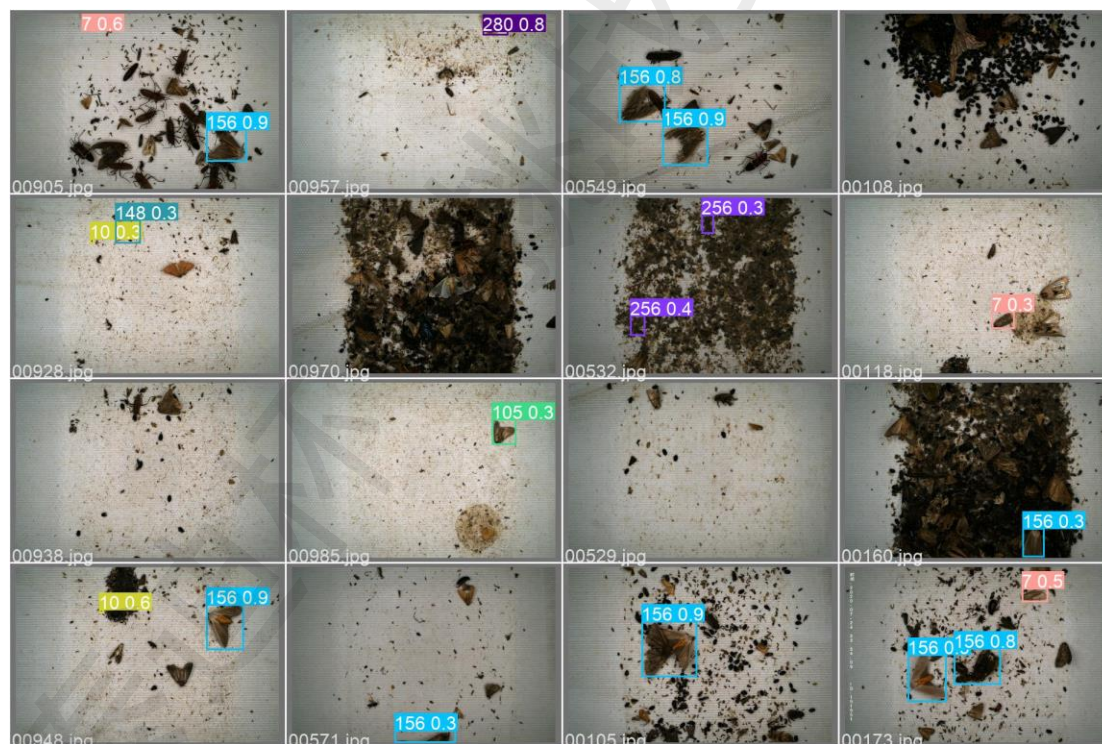


图 15: Yolov5 部分结果



## 5. 总结与分析

### 5.1 优势

- 本文选择了两种神经网络结构进行融合：FasterRCNN+Yolov5，结合了 Two Stage 和 One Stage 模型的优点。
- 对错误标注的数据进行了清洗。
- 对样本稀少的类别进行了数据增强。

### 5.2 不足

- 由于较多类别样本非常少，使得这些类别难以被模型识别。而我们的数据增强存在局限性，仅限于抠图和拼接。
- 由于缺乏昆虫识别相关的知识和背景，只依靠一些简单的观察和带噪学习算法剔除部分错误数据，并不能保证所有训练数据的标签正确。

### 5.3 后续改进

- 在本次任务中，很多图片是从相同背景得到的，识别出背景中会对实验结果造成影响的污渍或非害虫，可能有效提高模型预测的准确率。
- 采用更科学的方法进行数据增强和模型融合。

## 6. 参考文献:

- [1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
- [2] R. Girshick, "Fast R-CNN," in IEEE International Conference on Computer Vision (ICCV), 2015.
- [3] S Ren, K He, R Girshick, J Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." in IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017.
- [4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," arXiv:1512.03385, 2015.
- [5] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," International Journal of Computer Vision (IJCV), 2013.
- [6] J. Carreira and C. Sminchisescu, "CPMC: Automatic object segmentation using constrained parametric min-cuts," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2012.
- [7] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
- [8] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2012.
- [9] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in European Conference on Computer Vision (ECCV), 2014.
- [10] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in International Conference on Learning Representations (ICLR), 2014.
- [11] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in Neural Information Processing Systems (NIPS), 2013.
- [12] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
- [13] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov, "Scalable, high-quality object detection," arXiv:1412.1441 (v1), 2015.
- [14] L. Breiman, "Bagging Predictors." Machine Learning (1996).
- [15] Y Freund, Robert E. Schapire, "Experiment With a New Boosting Algorithm." Morgan Kaufmann, (1996).