



人工智能与深度学习

4-4 更多例子

王中雷

经济学院、王亚南经济研究院、邹至庄经济研究院

厦门大学

(第一版)

内容摘要

1. MobileNet

2. YOLO

MobileNet

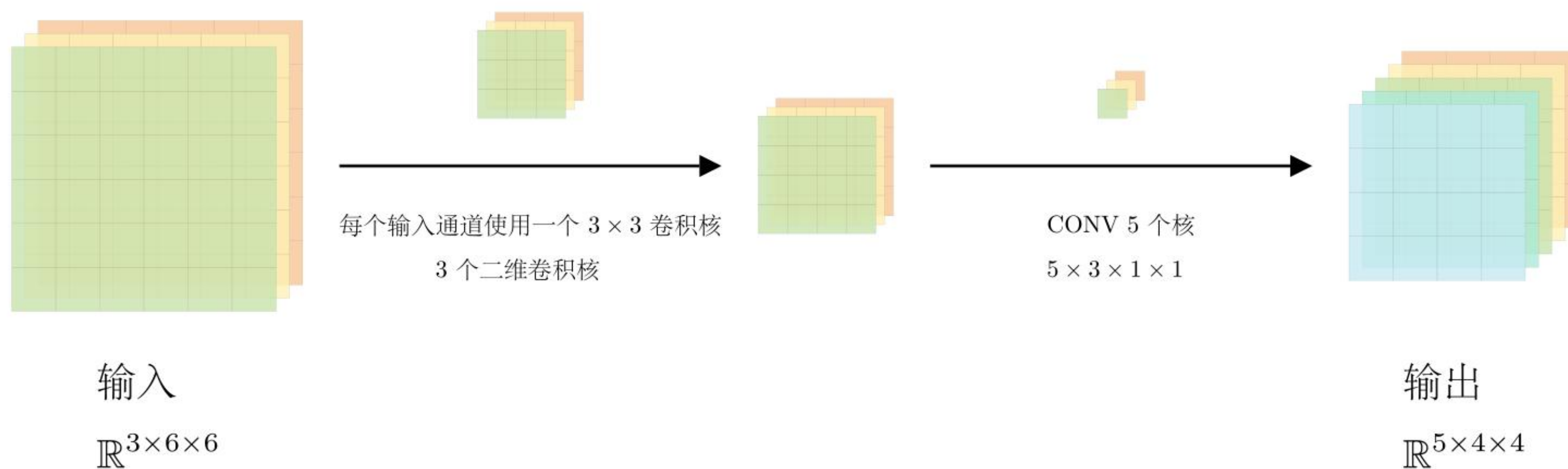
1. 一个计算效率较高，能够部署在手机等小型设备上的网络结构

- 基于一个“能够在小型计算设备上实现高效计算”的结构
- 目标：让卷积神经网络更小、更快，控制精度损失的同时降低参数量和计算量
- 核心结构：Depth-wise separable convolutions



Figure 1. MobileNet models can be applied to various recognition tasks for efficient on device intelligence.

直观理解



1. 只有 $3 \times 3 \times 3 + 3 \times 5 = 42$ 个模型参数

备注

1. 把传统的卷积操作拆成两个的子操作

- 更少的模型参数和更快的计算速度
- 能保留较为满意的性能

YOLO

1. 到目前为止，我们只讨论了分类问题
2. 在实际问题中，目标检测也很重要
3. 接下来，我们同时解决物体分类和位置识别两个任务

YOLO



[<https://www.superannotate.com/blog/yolo-object-detection>]

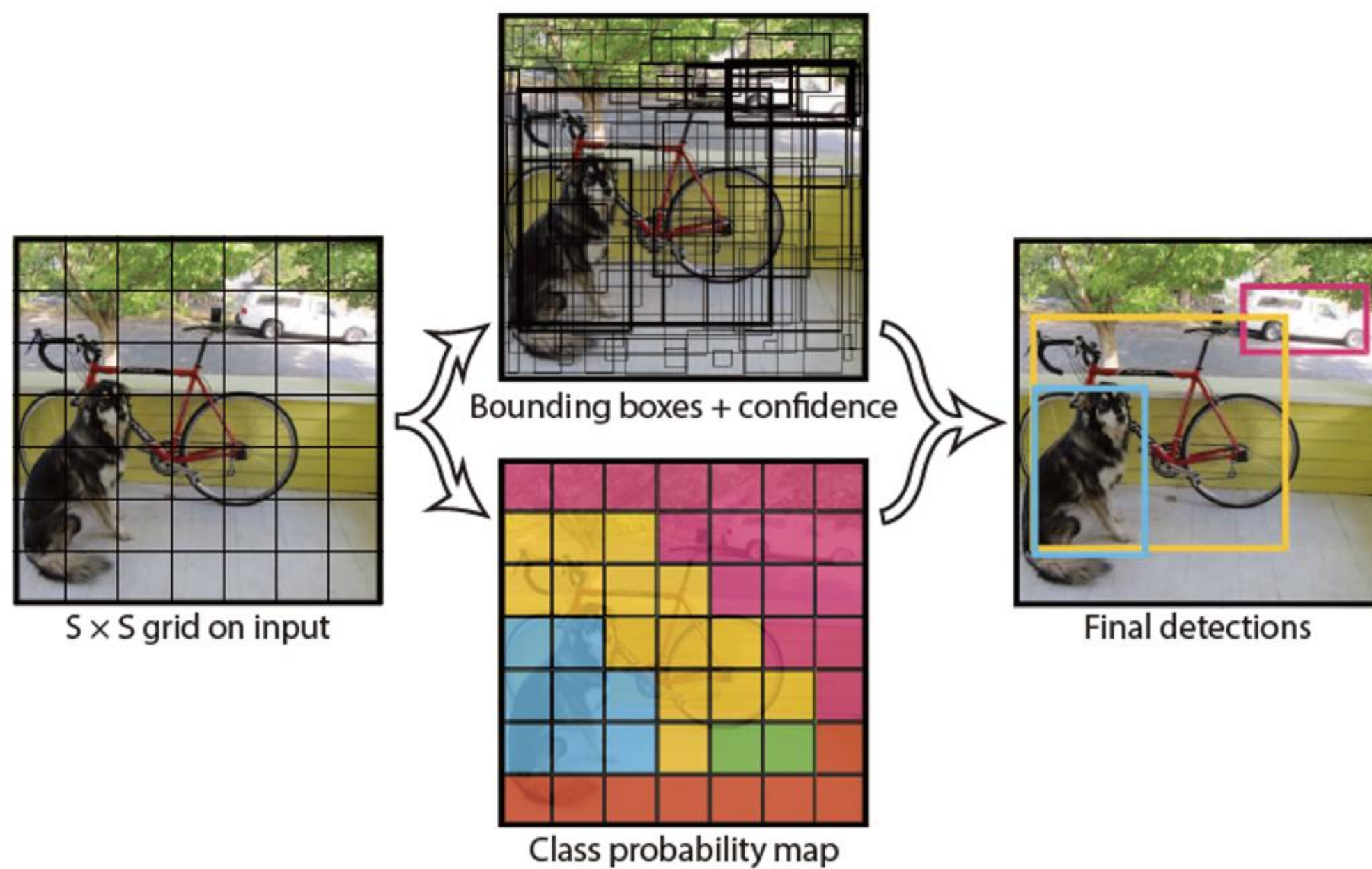
YOLO

1. 我们需要完成 C 类的分类问题，并且完成对应目标检测
2. YOLO (You Only Look Once) 为同时完成目标分类与定位提供了一种高效的统一框架
3. 我们只介绍 YOLO v1 (Redmon et al., 2016)
4. 更多介绍请参见 <https://www.bilibili.com/video/BV1JT411j7MR?p=2>

YOLO v1

1. 将图片划分为（不相交的） $S \times S$ 个网格单元
2. 对于网格单元，考虑 B 个边界框（bounding boxes）
3. 对于网格单元，我们基于 IoU 指标，选择一个具有“代表性”的边界框
4. 最小化一个特定的代价函数
5. 利用 non-maximum suppression 算法，完成物体的位置界定

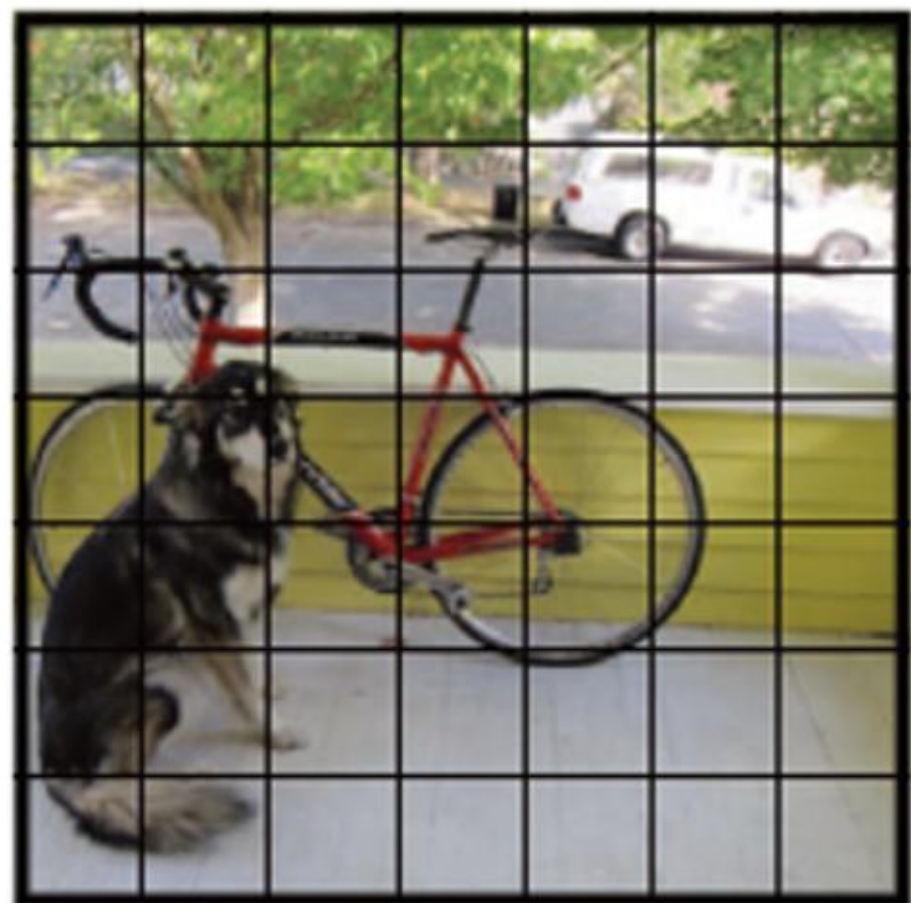
YOLO v1



[Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016), You Only Look Once: Unified, Real-Time Object Detection, CVPR, 779–788]

YOLO v1

1. 将图片划分为（不相交的） $S \times S$ 个网格单元



$S \times S$ grid on input

YOLO v1

1. 对于每个网格单元，考虑 $B = 2$ 个（备选）边界框

- 每个网格单元预测 B 个边界框，并利用 IoU 最大的边界框预测器指定为该物体的负责预测器
- 每个边界框由六个元素组成
 - ▷ x, y : 边界框相对于对应网格单元大小的中心坐标
 - ▷ w, h : 边界框相对于整个图片大小的宽和高
 - ▷ *confidence*: 量边界框中存在物体的可能性以及边界框定位准确程度的置信度
 - ▷ 分类得分: 每个网格单元还预测 C 个条件类别概率

YOLO v1

1. IoU 是 “Intersection over Union” 的缩写
2. 在训练过程中，我们已知物体的位置和标签，我们可以选择一个具有较大 IoU 的边界框
3. 并且在这个基础上，针对相应目标，对边界框进行微调

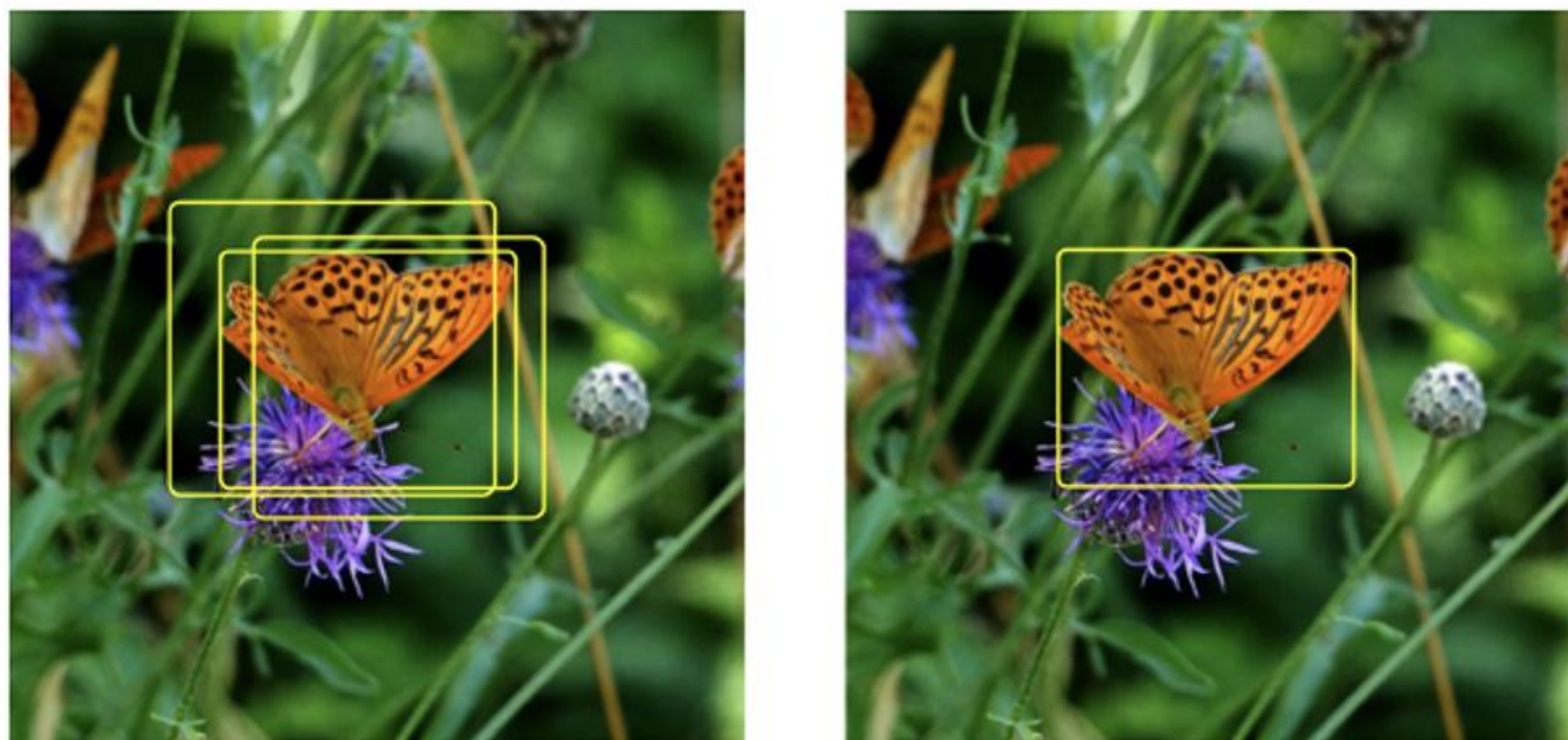
YOLO v1

1. 基于 $\lambda_{\text{coord}} = 5$ 以及 $\lambda_{\text{noobj}} = 0.5$ 的代价函数为

$$\begin{aligned} & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \quad (3) \end{aligned}$$

YOLO v1

1. 对于每个类别，首先删除类别得分低于阈值的候选框
2. 在然后选取得分最高的候选框，并删除与其 IoU 超过阈值的其余候选框，
3. 对剩余候选框重复上述过程，直至处理完毕



[<https://www.oreilly.com/library/view/practical-machine-learning/9781098102357/ch04.html>]

课程总结

1. MobileNet 回答的是：当计算资源受限时，怎么把卷积做得更轻、更适合部署
2. YOLO 回答的是：怎么把类别判断和位置预测，统一到一个高效框架里