第16章 目标检测算法

《人工智能算法》

清华大学出版社

2022年7月

提纲

- 引例
- 目标检测算法概述
- 卷积神经网络概述
- YOLO算法
- 总结

引例

- 为保证施工人员安全，需确保其正确佩戴各类安全护具
- 如何自动、快速、准确识别是否正确佩戴安全护具（如安全帽）

- 需大量人力资源
- 检测效率低
- 准确率受主观因素影响
- 无法保证监督的有效性

- 节省人力资源
- 每秒检测上百张图像
- 识别快速准确
- 可7×24小时实时监控

提纲

- 引例
- 目标检测算法概述
- 卷积神经网络概述
- YOLO算法
- 总结

目标检测算法概述 (1)

- **目标检测（Object Detection）**
  - 利用矩形边界框来确定图像中目标所在的位置及大小，识别目标所属的种类，并给出相应的置信度
- **应用场景**
  - 交通领域：交通违法事件检测
  - 医学领域：病变细胞检测
  - 工程领域：安全帽佩戴检测
  - 工业领域：绝缘子检测
  - 农业领域：农作物病虫害检测

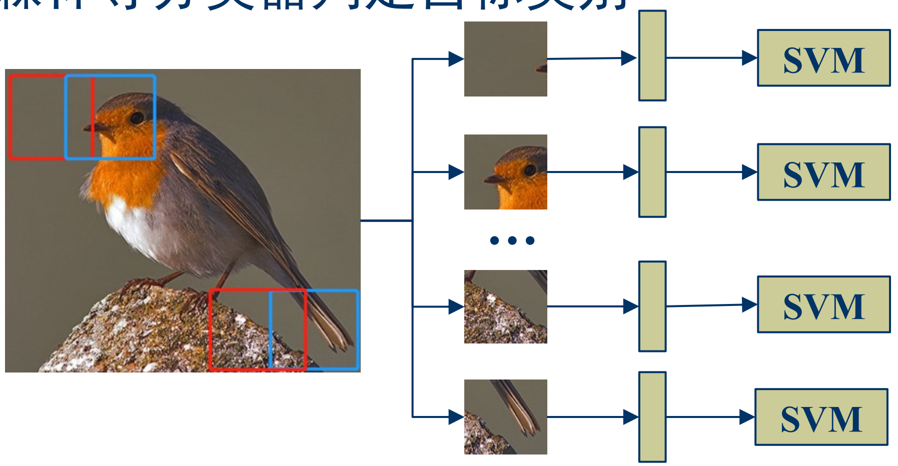
目标检测算法概述 (2)

- **目标检测算法分类**

- 目标检测算法
  - 传统的目标检测算法
HOG，DPM
  - 基于深度学习的目标检测算法
    - 一阶段目标检测算法
SSD，YOLO
    - 两阶段目标检测算法
R-CNN, Fast R-CNN, Faster R-CNN, Cascade R-CNN, Mask R-CNN

目标检测算法概述 (3)

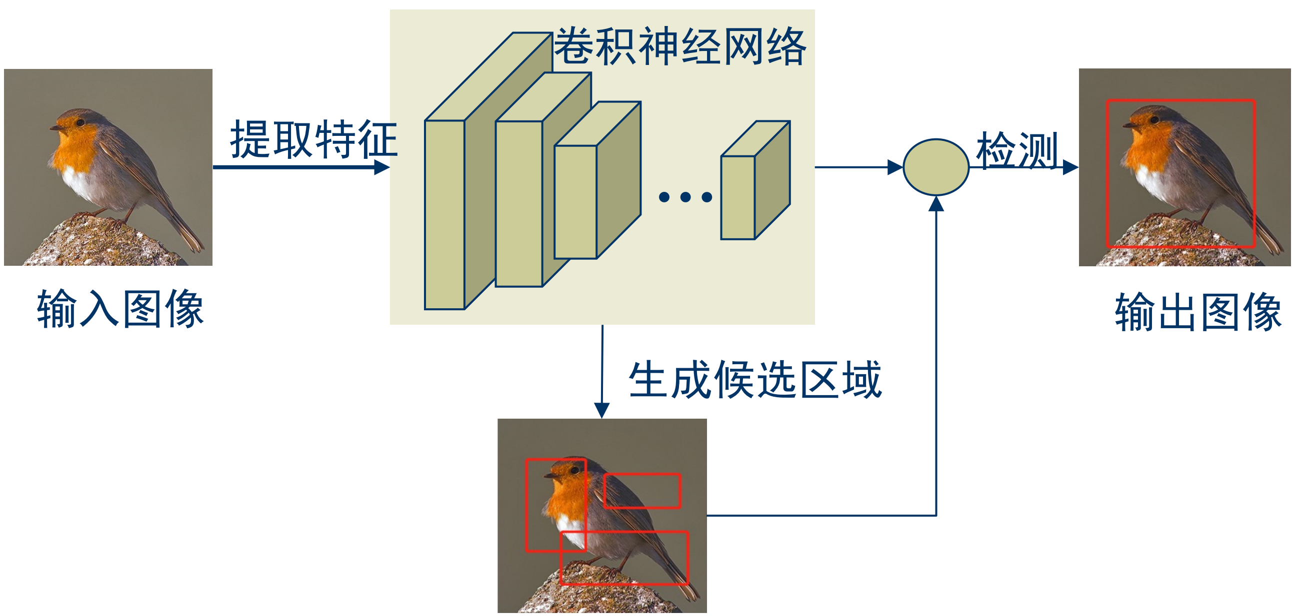
- **传统目标检测算法**
  - 采用滑动窗口的方式遍历整幅图像，选择候选区域
  - 逐个提取候选区域的特征
  - 利用支持向量机(SVM)、随机森林等分类器判定目标类别



- **优点**
  - 训练时间短、硬件要求较低
- **缺点**
  - 针对特定检测任务，需人工选择和设计不同特征，可移植性差
  - 特征提取和分类任务分离，无法完成端到端的训练

目标检测算法概述(4)

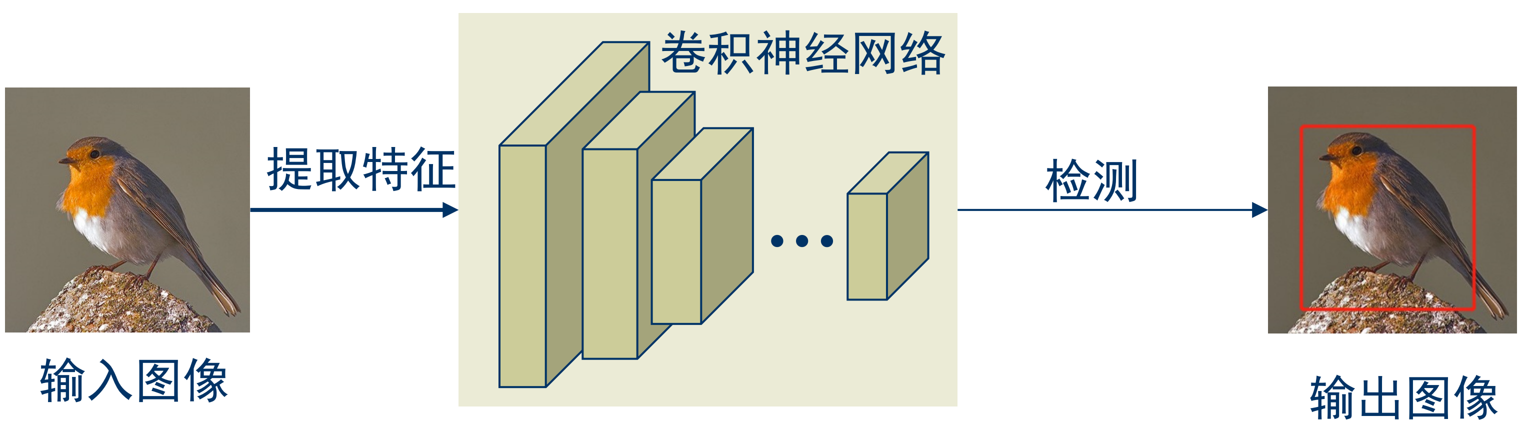
- **两阶段目标检测算法**
  - 第一阶段：对输入的图像提取候选区域特征信息
  - 第二阶段：利用卷积神经网络对候选区域进行分类和位置精修
  - 优点：基于候选区域做二次修正，相较于一阶段目标检测算法精度更高



目标检测算法概述(5)

- **一阶段目标检测算法**
  - 不需要产生候选区域阶段
  - 可直接计算目标物体的类别和检测框坐标
- **优点**

  一阶段目标检测算法不需要产生候选区域阶段，直接对图像进行计算得到检测结果，相较于两阶段目标检测算法，检测速度更快



提纲

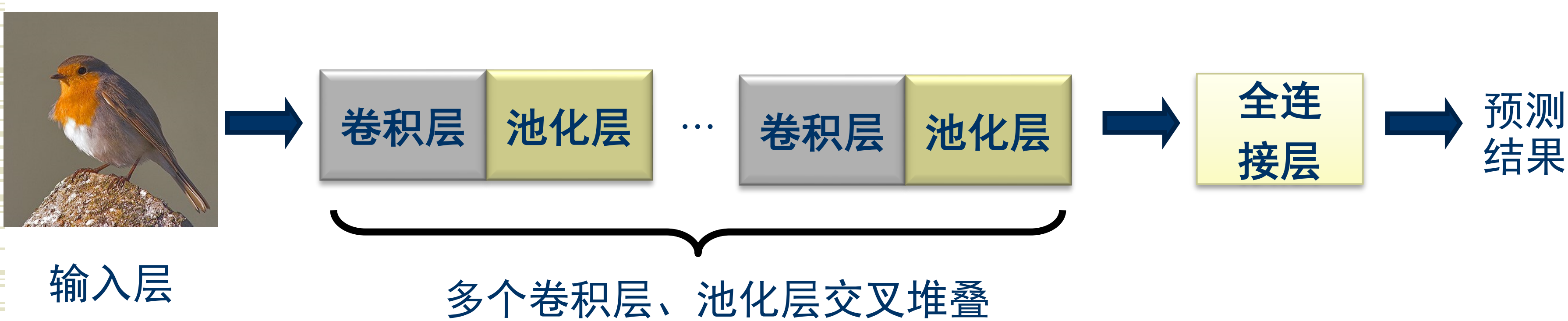
- 引例
- 目标检测算法概述
- 卷积神经网络概述
- YOLO算法
- 总结

卷积神经网络概述 (1)

- **卷积神经网络（Convolutional Neural Network, CNN）**

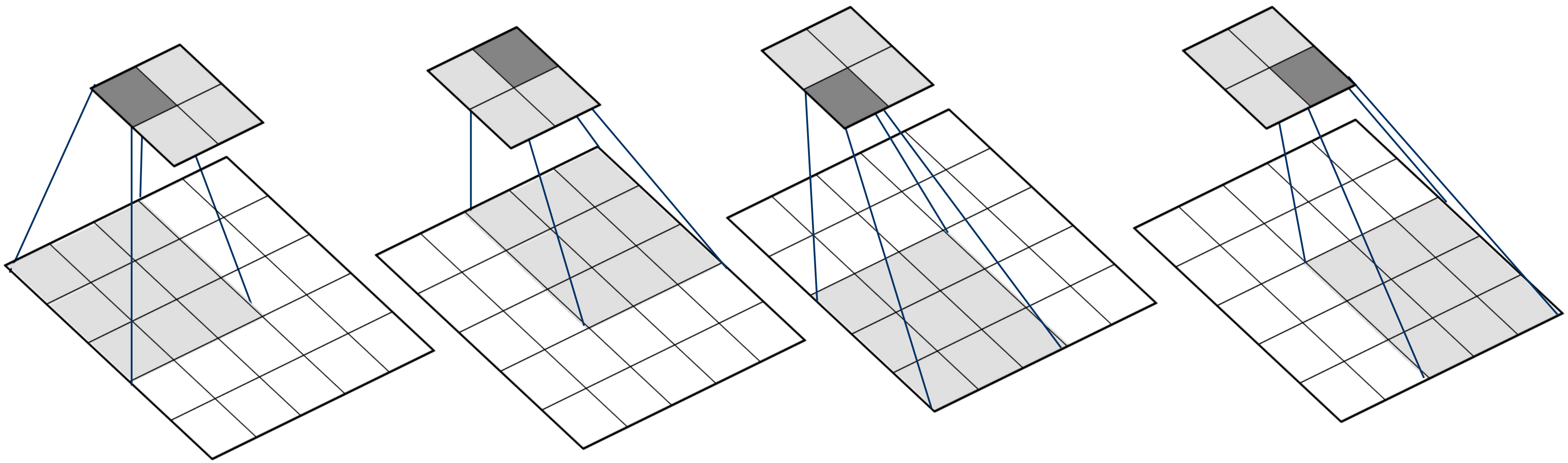
  卷积神经网络的层级结构主要包含输入层、卷积层、池化层、全连接层等，通过层级的组合可构建不同的卷积神经网络

- **优点**
  - 与全连接神经网络相比，层与层之间稀疏的局部连接减少了参数数量
  - 共享的卷积核参数有助于捕捉图像的局部特征



卷积神经网络概述 (2)

- **卷积层**
  - 对输入的图像进行特征提取，生成特征图
  - 每个卷积层包含一个或多个卷积核，以滑动窗口的形式在输入图像或特征图上进行卷积操作

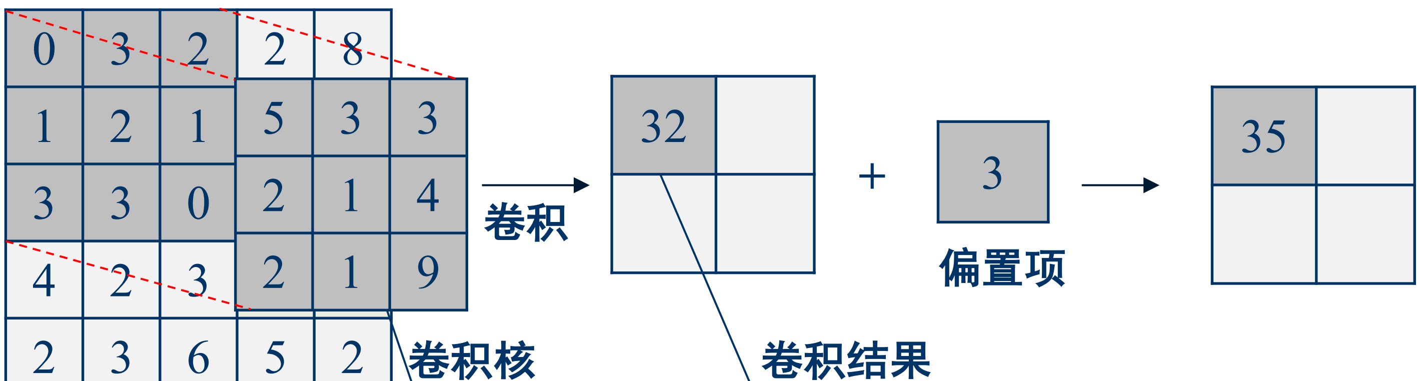


卷积核大小为3×3，步长为2，填充值为0

卷积神经网络概述 (3)

- **卷积计算**

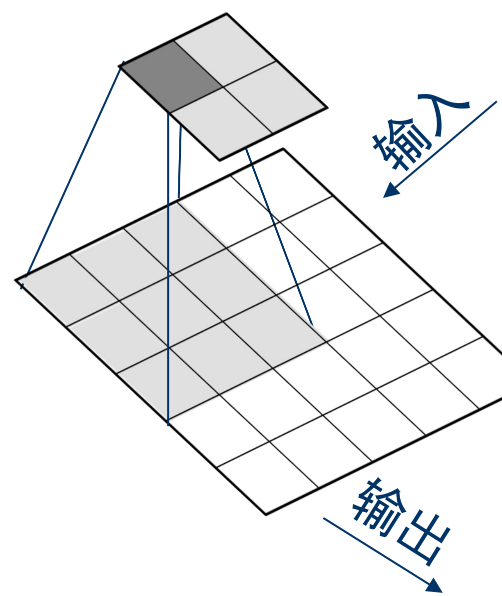
卷积核上的参数和窗口对应区域内的像素值进行乘法求和运算并加上偏置项，得到输出特征图对应位置的特征值



0×5+3×3+2×3+1×2+2×1+1×4+3×2+3×1+0×9 +3 = 35

卷积神经网络概述 (4)

- **卷积层输出特征图尺寸计算公式**
  - 输入特征图的尺寸$W_1 \times H_1 \times C_1$
  - 卷积核超参数：
    (1) 卷积核大小F
    (2) 卷积核个数K
    (3) 填充值P
    (4) 步长S
  - 输出特征图大小$W_2 \times H_2 \times C_2$：
    (1) $W_2 = \frac{W_1 - F + 2P}{S} + 1$
    (2) $H_2 = \frac{H_1 - F + 2P}{S} + 1$
    (3) $C_2 = K$

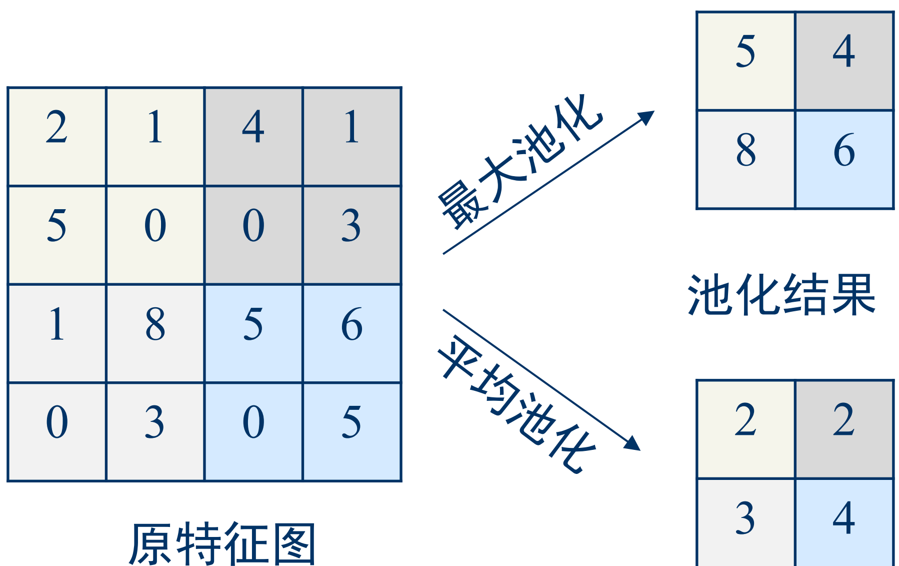


$W_1 = 5$
$H_1 = 5$
$C_1 = 1$

$$W_2 = \frac{5-3+0}{2} + 1 = 2$$
$$H_2 = \frac{5-3+0}{2} + 1 = 2$$
$$C_2 = 1$$

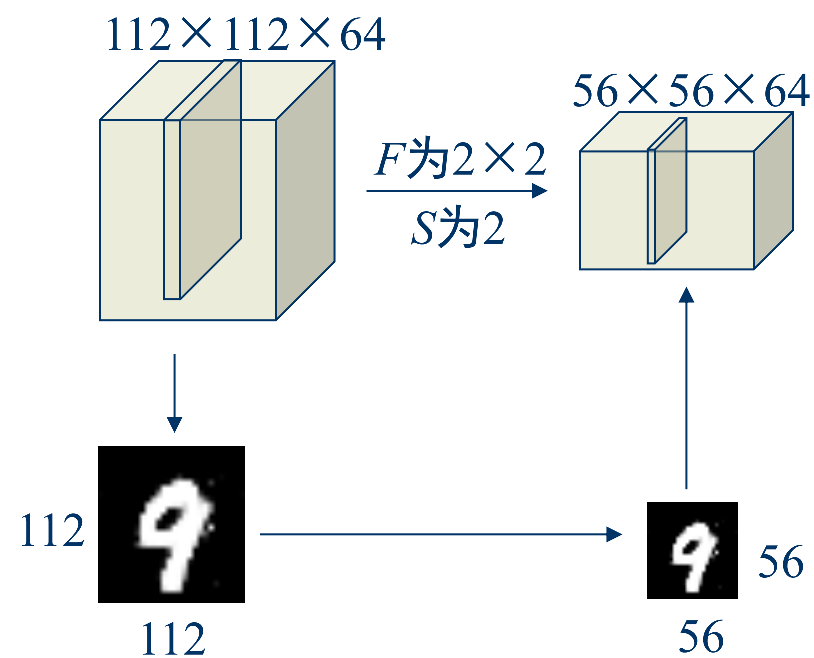
卷积神经网络概述 (5)

- **池化层**
  - 对输入特征图进行下采样，将子区域内的特征值压缩成一个能表示该区域的特征值，整合邻域特征，减少参数和计算量
  - 常见的池化操作包括最大池化和平均池化



卷积神经网络概述 (6)

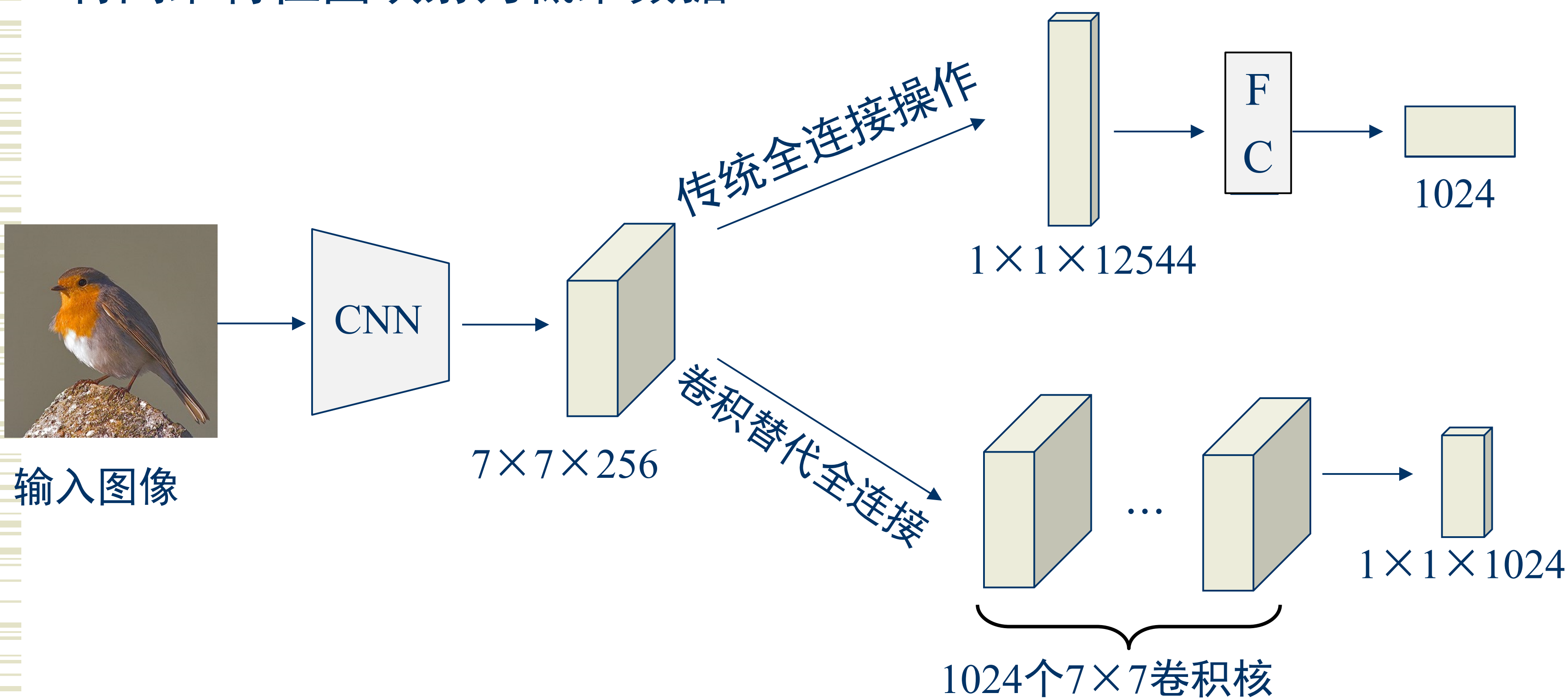
- **池化层输出特征图尺寸计算公式**
  - 操作在特征图上的每个通道单独进行，因此仅减小了特征图尺寸，特征图的通道数没有发生改变
  - 输入特征图的尺寸$W_1 \times H_1 \times C_1$
  - 池化层超参数：
    (1) 卷积核大小F
    (2) 步长S
  - 输出特征图大小$W_2 \times H_2 \times C_2$：
    (1) $W_2 = \frac{W_1 - F}{S} + 1$
    (2) $H_2 = \frac{H_1 - F}{S} + 1$
    (3) $C_2 = C_1$



卷积神经网络概述 (7)

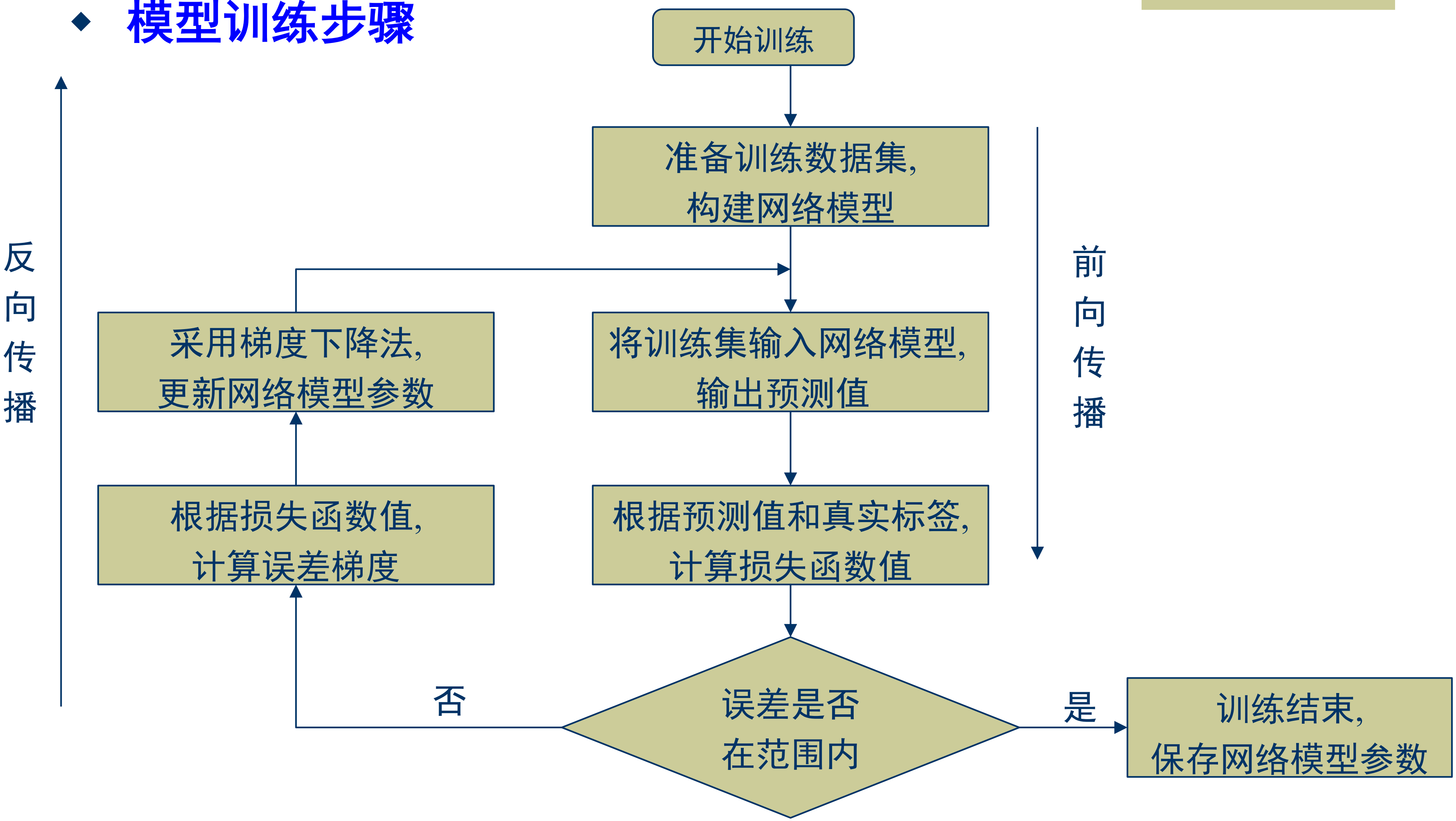
- **全连接层**

将高维特征图映射为低维数据



卷积神经网络概述 (8)

- **模型训练步骤**

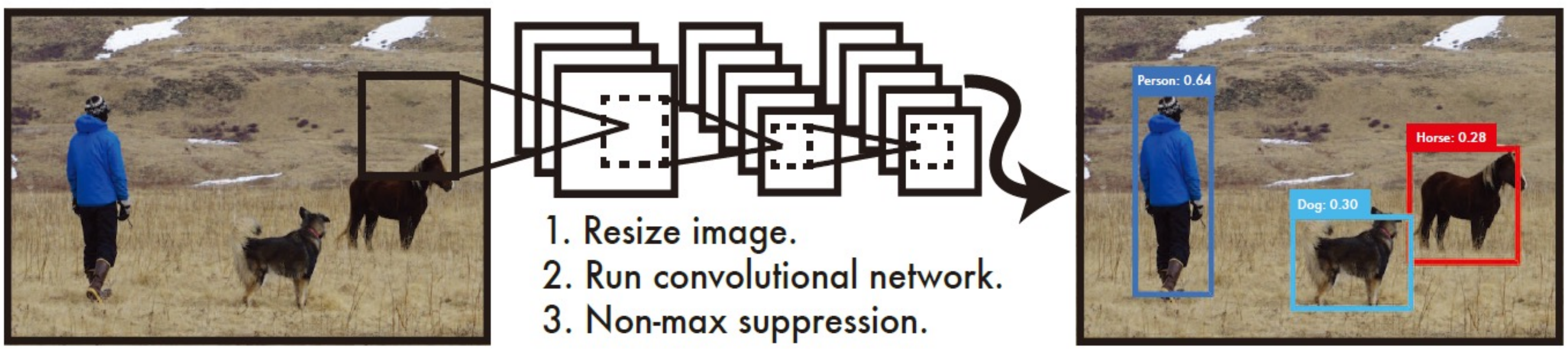


提纲

- 引例
- 目标检测算法概述
- 卷积神经网络
- YOLO算法
- 总结

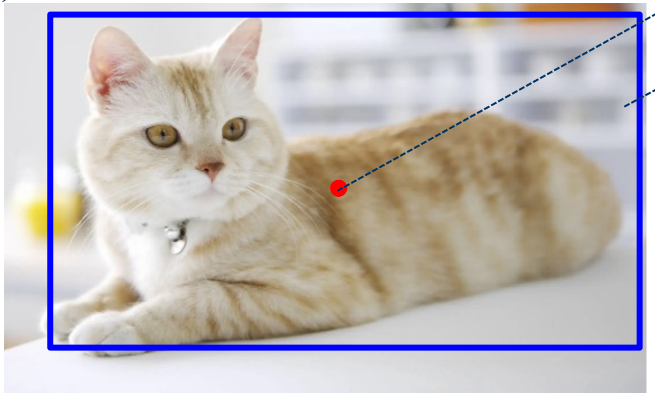
YOLO算法概述

- 由Joseph Redmon和Ali Farhadi等于2015年提出
- 与R-CNN等二阶段算法不同
- 仅基于单个CNN，并将特征提取和检测框定位两个步骤相结合
- 可直接从完整的图像中预测检测框、得到分类置信度



YOLO算法训练 (1)

- **图像预处理**
  - **标注检测框**
    - ✓ 训练数据集包括原始图像和标注信息两个部分
    - ✓ 标注信息包含任意多个检测框的位置信息、置信度及类别信息



训练数据集

$(\hat{x}, \hat{y})$：中心坐标

$(\hat{w}, \hat{h})$：宽度和高度

$\hat{p}$：类别信息，表示是否包含某类物体

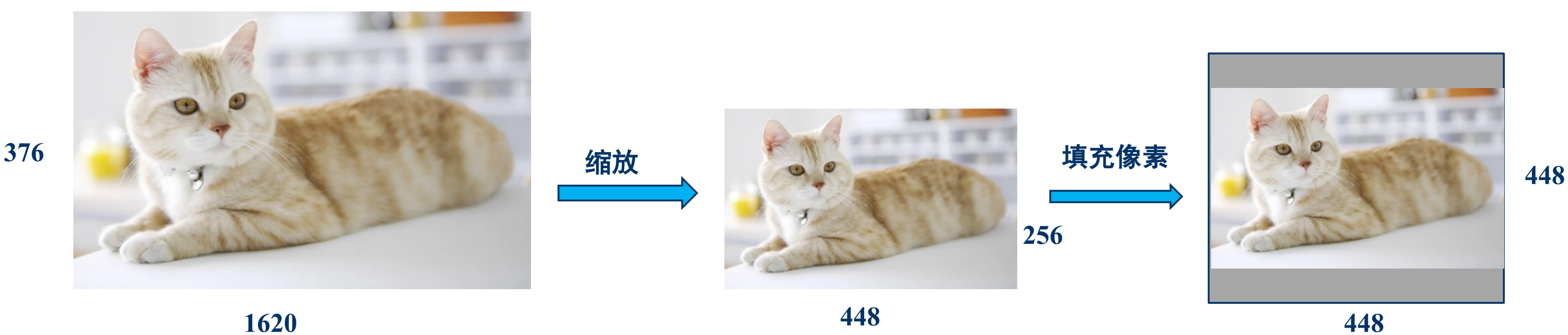
$\widehat{Conf}$：标注检测框的置信度为1，表示框中有物体且位置准确（可由Labelme自动生成）

$(\hat{x}, \hat{y}, \hat{w}, \hat{h}, \widehat{Conf}, \hat{p})$：一张原始图像的标注信息由该六元组描述

例如，左图检测框的标注信息为(850, 150, 1300, 300, 1, cat)

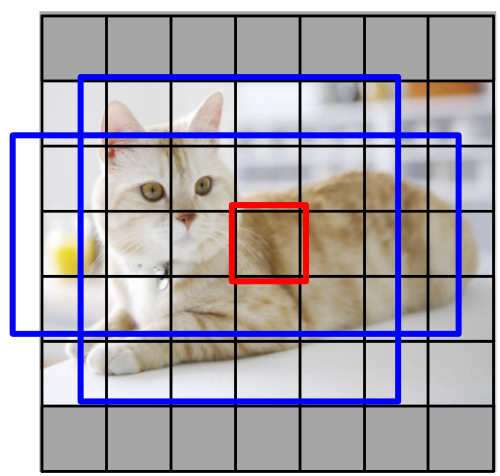
YOLO算法训练 (2)

- **图像缩放**
  - ✓ 基于YOLO的网络结构，需将输入图像的长宽像素缩放为固定值448×448
  - ✓ 先将图片中最长的边缩放到448像素
  - ✓ 再对短边的空白位置补上灰色



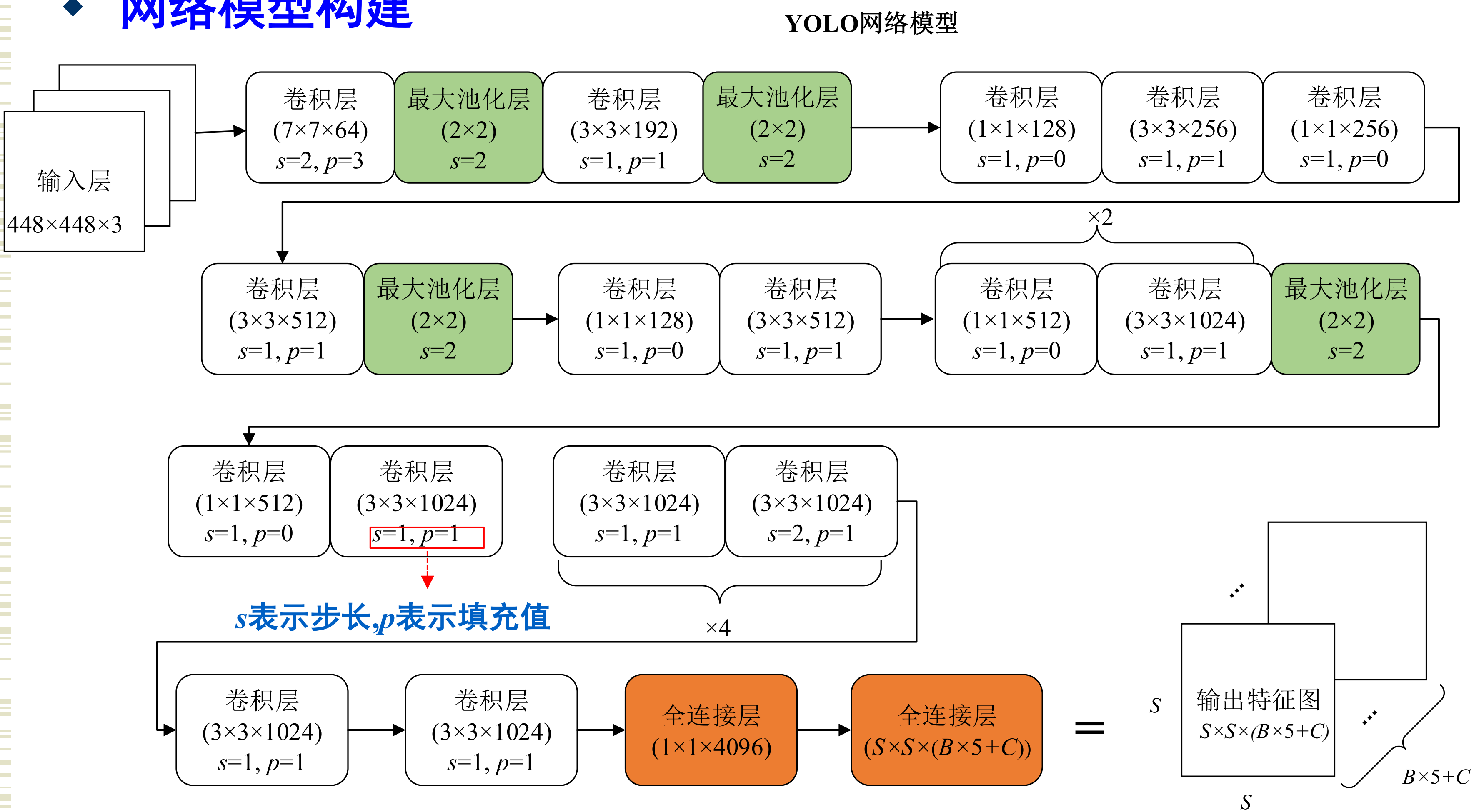
YOLO算法训练步骤(3)

- **设置网格及网格中检测框个数**
  - ✓ 把经缩放的图像划分为$S\times S$个网格（Grid）
  - ✓ 每个网格中设置B个检测框
  - ✓ 例如：将S和B分别设置为7和2，即将图像划分为7×7个网格，每个网格有2个检测框



YOLO算法训练 (4)

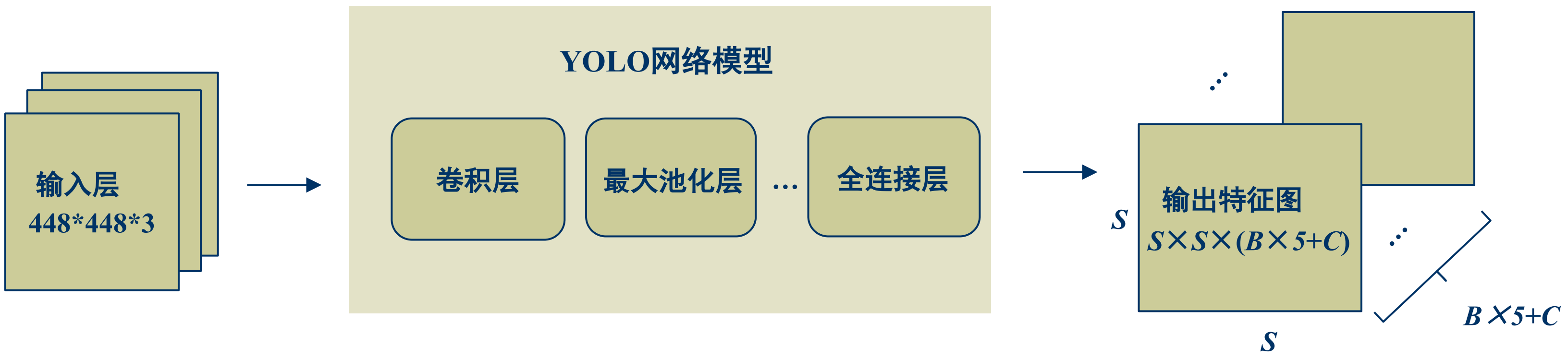
- **网络模型构建**



YOLO算法训练 (5)

- **前向传播**

将训练图像输入YOLO网络进行前向传播，最终将输入的448×448×3的特征图，转化为$S \times S \times (B \times 5+C)$的输出特征图$\boldsymbol{O}$

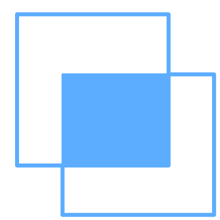


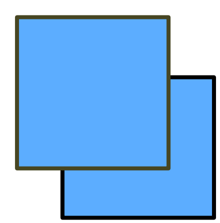
YOLO算法训练 (6)

- **非极大值抑制**

  - 根据预测检测框的坐标值、置信度及类别概率，利用非极大值抑制（Non-Maximum Suppression, NMS）方法进行筛选，得到包含目标物体的检测框
  - IoU(Intersection over Union)：两个检测框的交集面积与并集面积的比值称为交互比，用于度量两个检测框的交叠程度

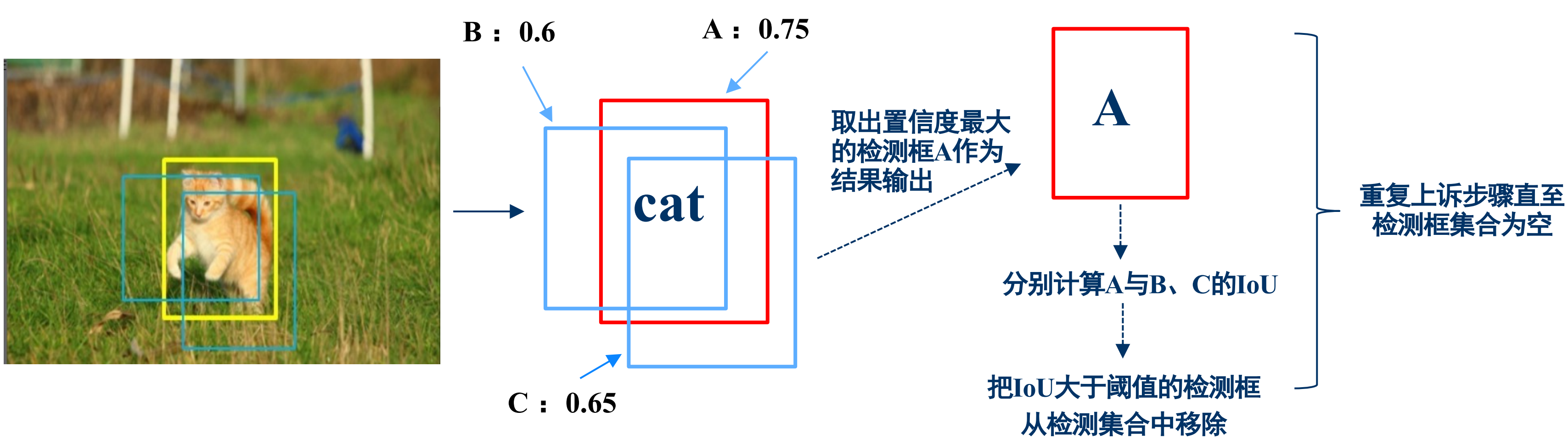
$$\mathrm{IoU} = \frac{\text{交集面积}}{\text{并集面积}}$$





YOLO算法训练 (7)

- **非极大值抑制过程**
  - 从检测框集合中取出最大置信度对应的检测框A并作为结果输出
  - 逐一计算其余检测框与检测框A的IoU，将IoU大于给定阈值（阈值一般设为0.5）的检测框从检测框集合中移除
  - 重复上述2个步骤直至检测框集合为空



YOLO算法损失函数 (1)

- **计算损失函数**

- 目标检测算法中的损失误差，通常包括检测框定位损失、检测框目标损失和分类损失
- 检测框定位损失反映了检测框位置的误差

YOLO算法的损失函数： $L = \boxed{l_{xy}} + l_{wh} + l_{obj} + l_{cls}$

中心坐标的定位损失项： $l_{xy} = \boxed{\lambda_{coord}} \sum_{i=1}^{S^2} \sum_{j=1}^{B} \boxed{1_{ij}^{obj}} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]$

该损失项的系数，越大表示该误差项对损失函数的影响越大，YOLO算法设$\lambda_{coord} = 5$

第i个网格存在目标物体且第j个检测框负责预测该目标物体

YOLO算法损失函数 (2)

- **损失函数—计算检测框定位损失**

检测框定位损失反映了检测框位置的误差

$$L = l_{xy} + \boxed{l_{wh}} + l_{obj} + l_{cls}$$

高与宽的定位损失项：

$$l_{wh} = \boxed{\lambda_{coord}} \sum_{i=1}^{S^2} \sum_{j=1}^{B} \boxed{1_{ij}^{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \right]$$

该损失项的系数，越大表示该误差项对损失函数的影响越大，YOLO算法设 $\lambda_{coord} = 5$

第i个单元格存在目标且第j个检测框负责预测该目标物体

YOLO算法损失函数 (3)

- **损失函数—计算检测框目标损失**

检测框目标损失反映了检测框中是否包含目标的置信度损失

$$L = l_{xy} + l_{wh} + \boxed{l_{obj}} + l_{cls}$$

$$l_{obj} = \sum_{i=1}^{S^2}\sum_{j=1}^{B} 1_{ij}^{\boldsymbol{obj}}\left(Conf_i - \widehat{Conf_i}\right)^2 + \boxed{\lambda_{\boldsymbol{noobj}}}\sum_{i=1}^{S^2}\sum_{j=1}^{B} \boxed{1_{ij}^{\boldsymbol{noobj}}}\left(Conf_i - \widehat{Conf_i}\right)^2$$

该损失项的系数，YOLO算法设 $\lambda_{noobj} = 0.5$

第i个单元格不存在目标物体且第j个检测框负责预测不存在的目标

YOLO算法损失函数 (4)

- **损失函数—计算分类损失**

分类损失反映了检测框中的目标物体分类是否准确

$$L = l_{xy} + l_{wh} + l_{obj} + \boxed{l_{cls}}$$

$$l_{cls} = \sum_{i=1}^{S^2} \boxed{1_i^{obj}} \sum_{c \in C} \left(\boxed{p_i(c)} - \boxed{\hat{p}_i(c)}\right)^2$$

表示第i个网格中是否存在物体

YOLO预测的第i个网格所包含目标物体属于类别c的概率

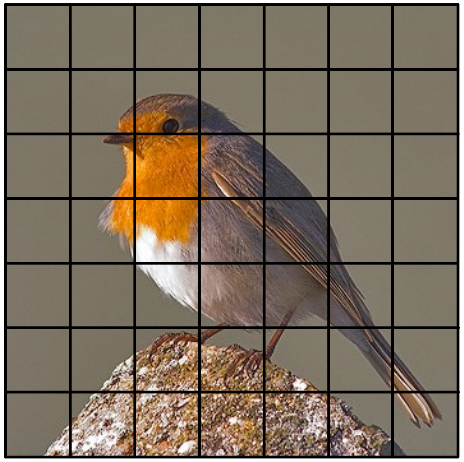
表示第i个网格所包含目标物体是否属于类别c

YOLO算法示例 (1)

- **示例**

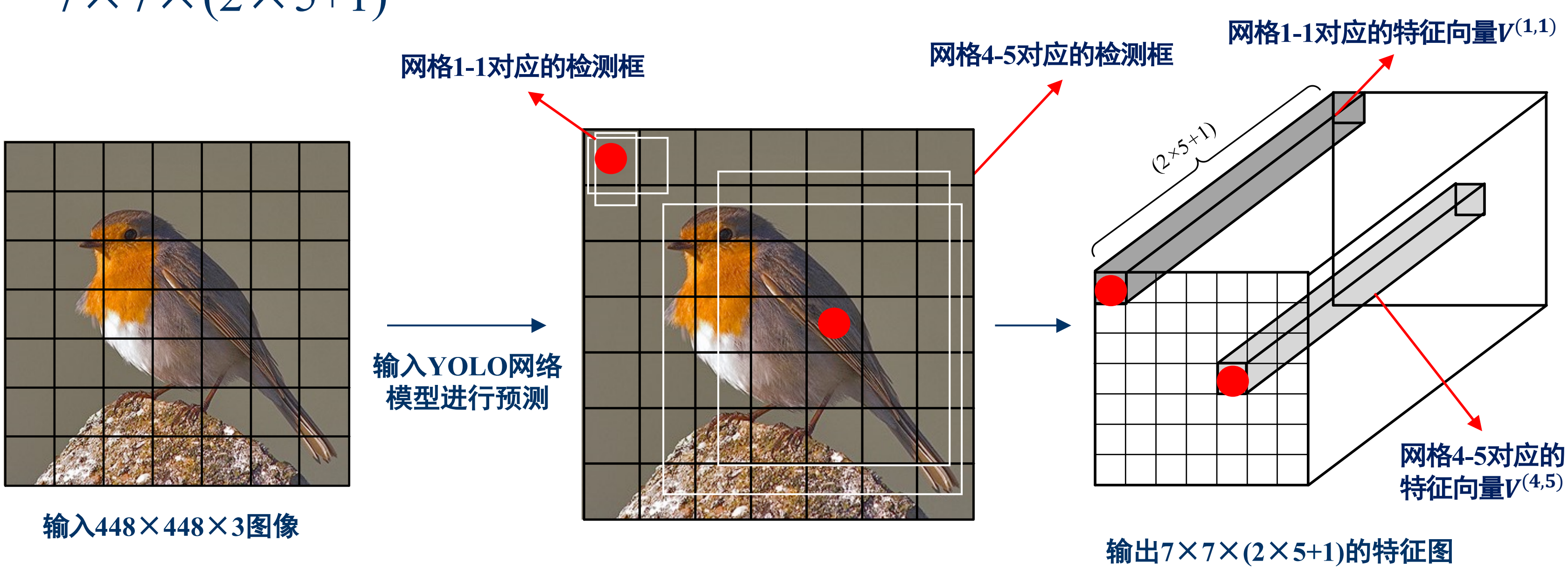
以基于YOLO算法预测图像中的鸟类为例，将输入图像缩放448×448×3，设S=7，即划分为(7×7)个网格

S为7，将图像划分为7×7的网格



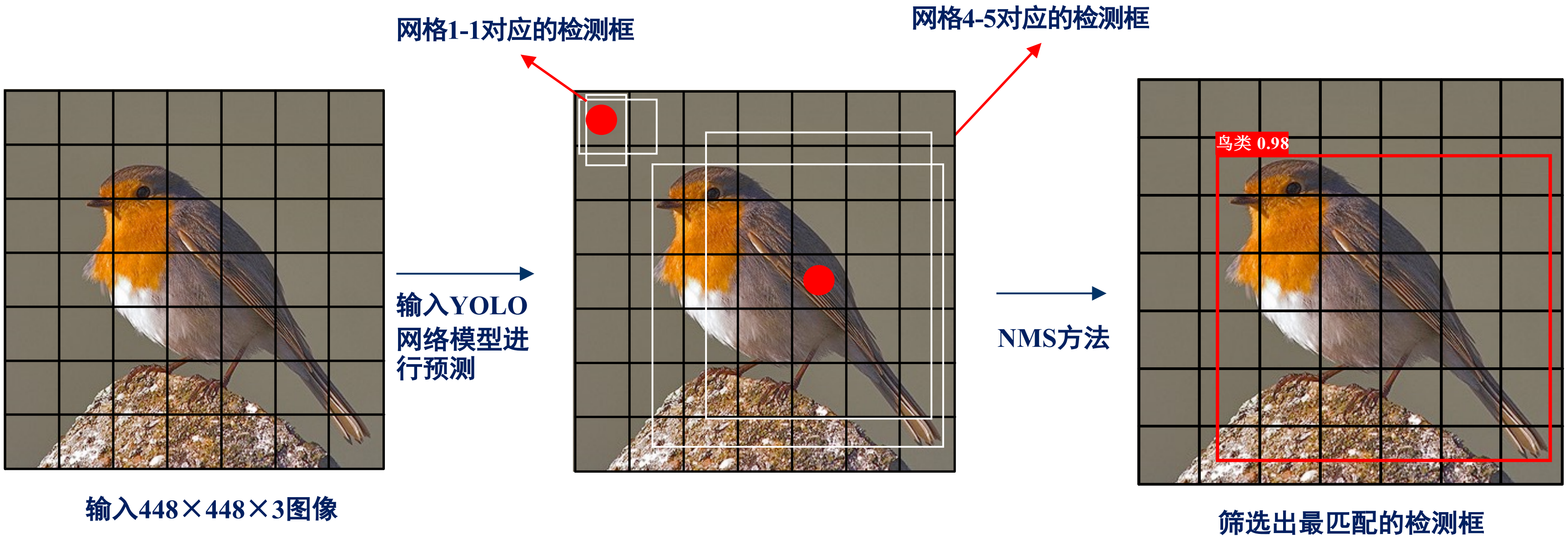
YOLO算法示例 (2)

- 将处理后的图像输入经过训练的YOLO网络模型中进行前向传播
- 每个网格采用2个检测框检测目标物体（B=2）
- 由于目标物体仅有鸟类，因此C=1，最终输出特征图的维度为 $7\times7\times(2\times5+1)$



YOLO算法示例 (3)

- 输出特征图共预测出7×7×2 (98)个检测框及相应置信度
- 将98个检测框通过NMS方法进行筛选
- 最终得到网格4-5所对应的检测框为最终结果



提纲

- 引例
- 目标检测算法概述
- 卷积神经网络
- YOLO算法
- 总结

总结

- 目标检测的基本思想、应用场景和常见方法
- CNN的基本操作、模型结构与训练步骤
- 目标检测的重要算法实例：
  - ✓ 一阶段目标检测方法YOLO的基本思想、模型结构与训练步骤
  - ✓ 基于YOLO算法预测图像中的鸟类

结语

谢谢！