



温州大學
WENZHOU UNIVERSITY

深度学习-目标检测

黄海广 副教授

2023年04月

- 01 目标检测概述**
- 02 目标检测算法**
- 03 YOLO算法**
- 04 Faster RCNN算法**

1.目标检测概述

3

01 目标检测概述

02 目标检测算法

03 YOLO算法

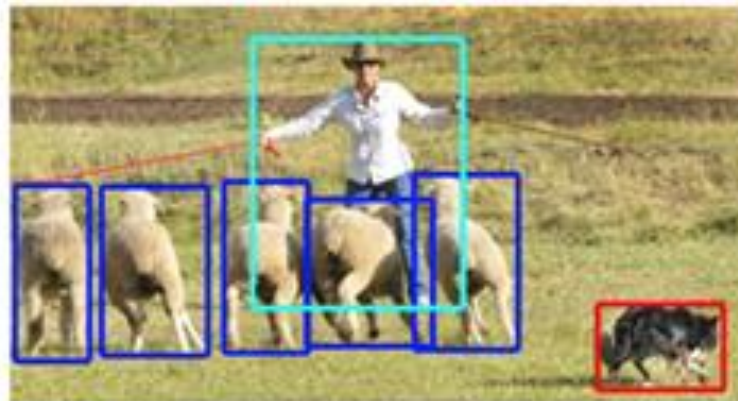
04 Faster RCNN算法

1.目标检测概述

4



分类 (Classification)
即是将图像结构化为某一类别的信息, 用事先确定好的类别(string)或实例ID来描述图片。这一任务是最简单、最基础的图像理解任务, 也是深度学习模型最先取得突破和实现大规模应用的任务。



检测 (Detection)
分类任务关心整体, 给出的是整张图片的内容描述, 而检测则关注特定的物体目标, 要求同时获得这一目标的类别信息和位置信息。



分割 (Segmentation)
分割包括语义分割 (semantic segmentation) 和实例分割 (instance segmentation), 前者是对前背景分离的拓展, 要求分离开具有不同语义的图像部分, 而后者是检测任务的拓展, 要求描述出目标的轮廓 (相比检测框更为精细)。

1. 目标检测概述

5

目标检测和识别

- 怎样检测和识别图像中物体，如汽车、牛等？



1. 目标检测概述

6

目标识别的应用



视频监控



智能汽车



场景搜索



基于内容的图像\视频检索



医学图像分析

1.目标检测概述

7

难点之一：如何鲁棒识别？



光照的影响



物体姿态的影响



背景混淆



遮挡



类内差异



视点的影响

1. 目标检测概述

8

类内差异 (intra-class variability)



Many face of Madonna

1. 目标检测概述

9

类间相似性 (inter-class similarity)



www.marykateandashley.com

双胞胎



news.bbc.co.uk/1/hi/english/in_depth/americas/2000/us_elections

父子

1.目标检测概述

10

学术和工业界主要将目标检测算法分成三类：

1.传统的目标检测框架

**2.基于深度学习的Two Stages目标检测框架
(准确度有优势)**

**3.基于深度学习的One Stage目标检测框架
(速度有优势)**

1.目标检测概述

11

1.传统的目标检测框架

- (1) 候选区域选择（采用不同尺寸、比例的滑动窗口对图像进行遍历）；
- (2) 对不同的候选区域进行特征提取（SIFT、HOG等）；
- (3) 使用分类器进行分类（SVM、Adaboost等）。

1.目标检测概述

12

2.基于深度学习的Two Stages目标检测框架 (准确度有优势)

此类算法将检测问题分为两个阶段，

第一阶段生成大量可能含有目标的候选区域（Region Proposal），并附加大概的位置信息；

第二个阶段对其进行分类，选出包含目标的候选区域并对其位置进行修正（常使用R-CNN、Fast R-CNN、Faster R-CNN等算法）。

1.目标检测概述

13

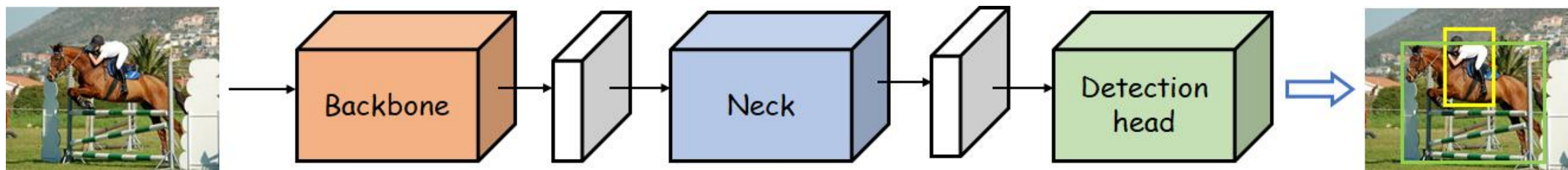
3.基于深度学习的One Stage目标检测框架 (速度有优势)

此类检测算法属于端到端（End-to-End），不需要生成大量候选区域的阶段，而是将问题转化为回归（Regression）问题处理，使用完整图像作为输入，直接在图像的多个位置上回归出该位置的目标边框及所属类别（常使用YOLO、SSD、CornerNet等算法）。

1. 目标检测概述

14

目标检测网络的常见结构



Backbone network，即**主干网络**，是目标检测网络最为核心的部分

Neck network，即**瓶颈网络**，Neck部分的主要作用就是将由backbone输出的特征进行整合。

Detection head，即**检测头**，这一部分的作用就就是若干卷积层进行预测，也有些工作里把head部分称为decoder（解码器）的。

2.目标检测算法

15

01 目标检测概述

02 目标检测算法

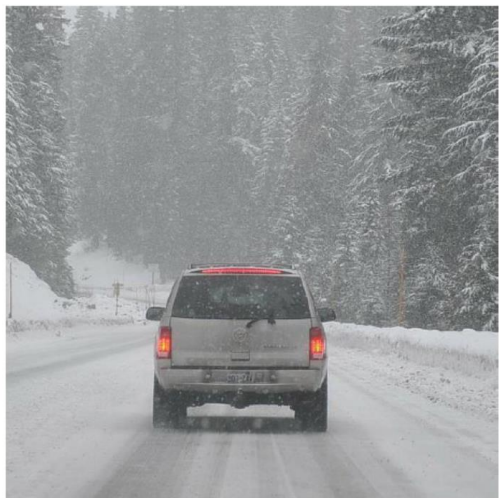
03 YOLO算法

04 Faster RCNN算法

2. 目标检测算法

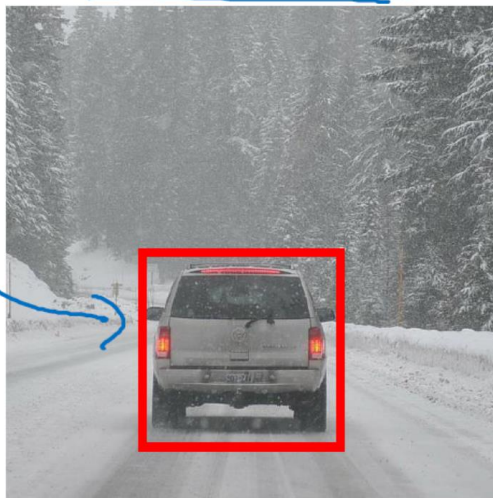
16

Image classification



"Car"

Classification with
localization



"Car"

1 object

Detection

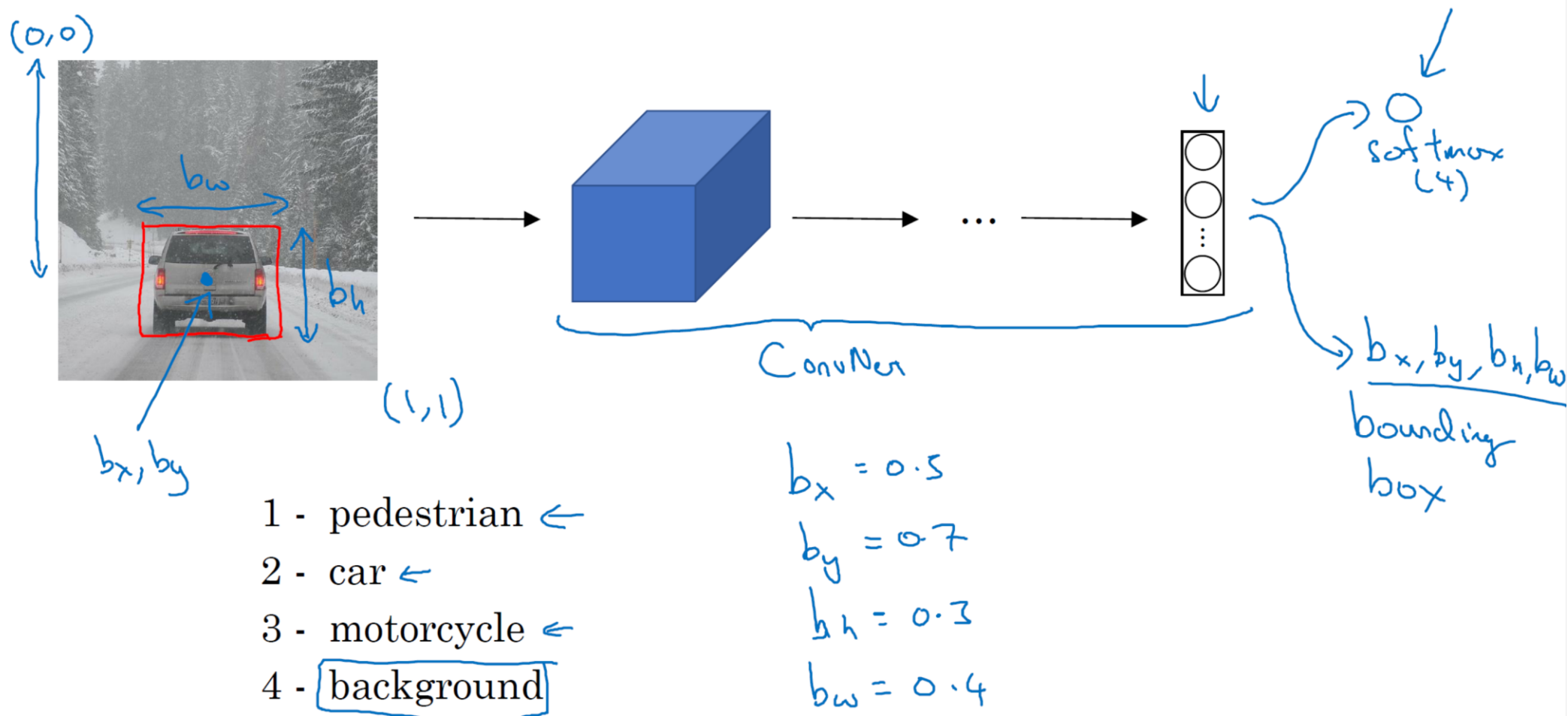


multiple
objects

2. 目标检测算法

17

Classification with localization



2. 目标检测算法

18

目标标签 y 的定义如下:

$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

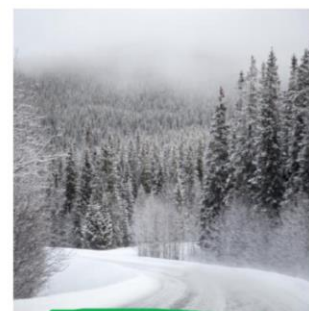
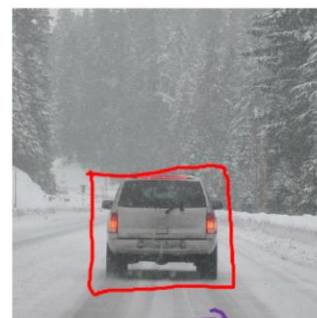
1 - pedestrian
 2 - car ←
 3 - motorcycle
 4 - background ←

$$L(\hat{y}, y) = \begin{cases} (\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2 + \dots + (\hat{y}_8 - y_8)^2 & \text{if } y_1 = 1 \\ (\hat{y}_1 - y_1)^2 & \text{if } y_1 = 0 \end{cases}$$

$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

is there any object?

Need to output b_x, b_y, b_h, b_w , class label (1-4)



$$(x, y) \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

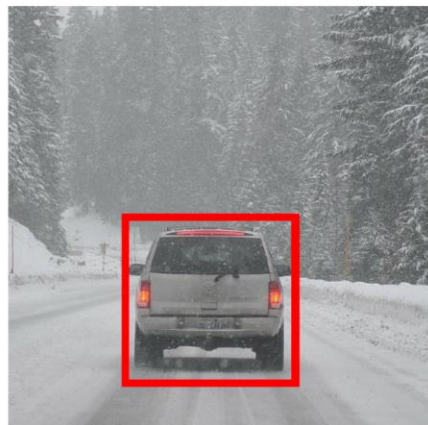
$$\begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}$$

← "don't care"

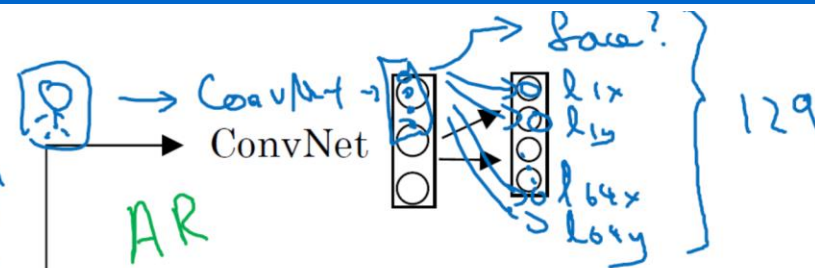
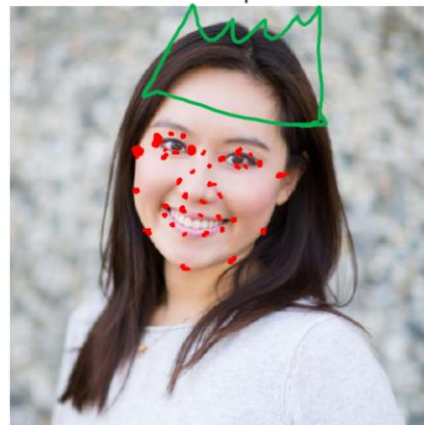
2.目标检测算法

19

Landmark detection



b_x, b_y, b_h, b_w



$\left. \begin{matrix} l_{1x}, l_{1y}, \\ l_{2x}, l_{2y}, \\ l_{3x}, l_{3y}, \\ l_{4x}, l_{4y}, \\ \vdots \\ l_{64x}, l_{64y} \end{matrix} \right\} x, y$

$\left. \begin{matrix} l_{1x}, l_{1y}, \\ \vdots \\ l_{32x}, l_{32y} \end{matrix} \right\}$

备注：图中的模特是吴恩达老师的夫人**Carol Reiley**

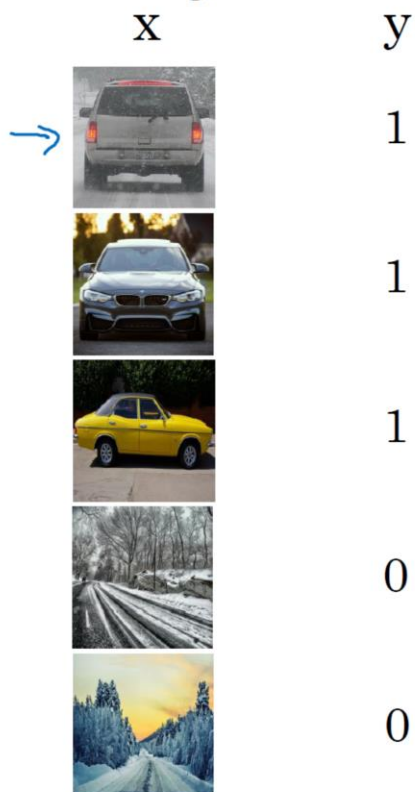
2. 目标检测算法

20

汽车检测案例



Training set:

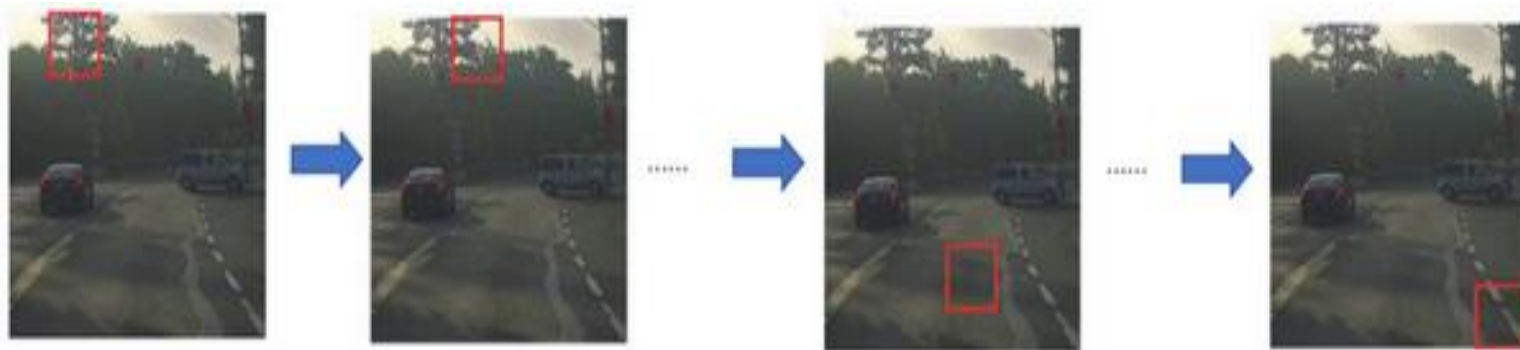


→ ConvNet → y

2. 目标检测算法

21

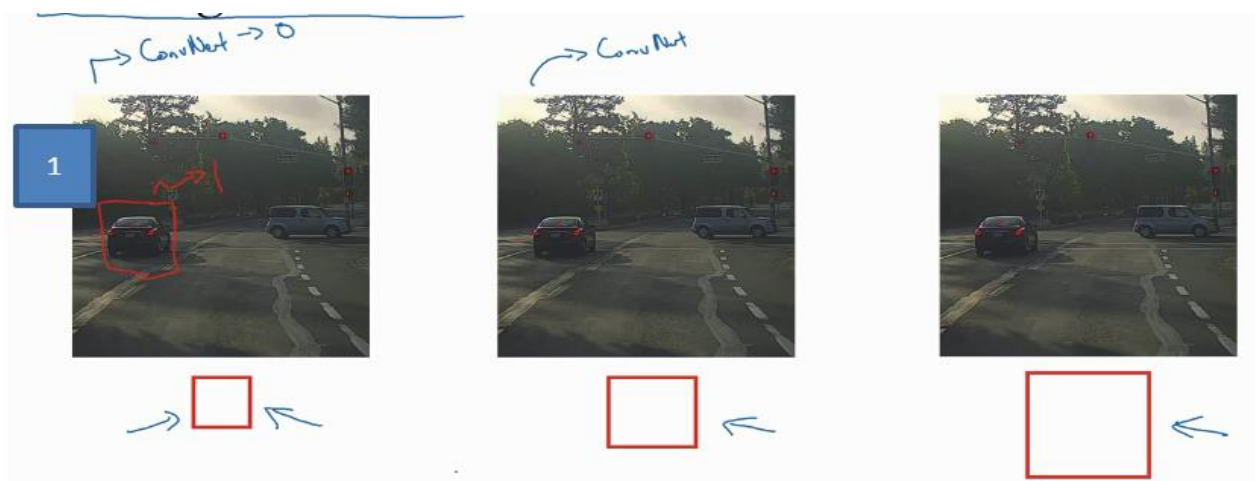
滑动窗口检测



2. 目标检测算法

22

滑动窗口检测

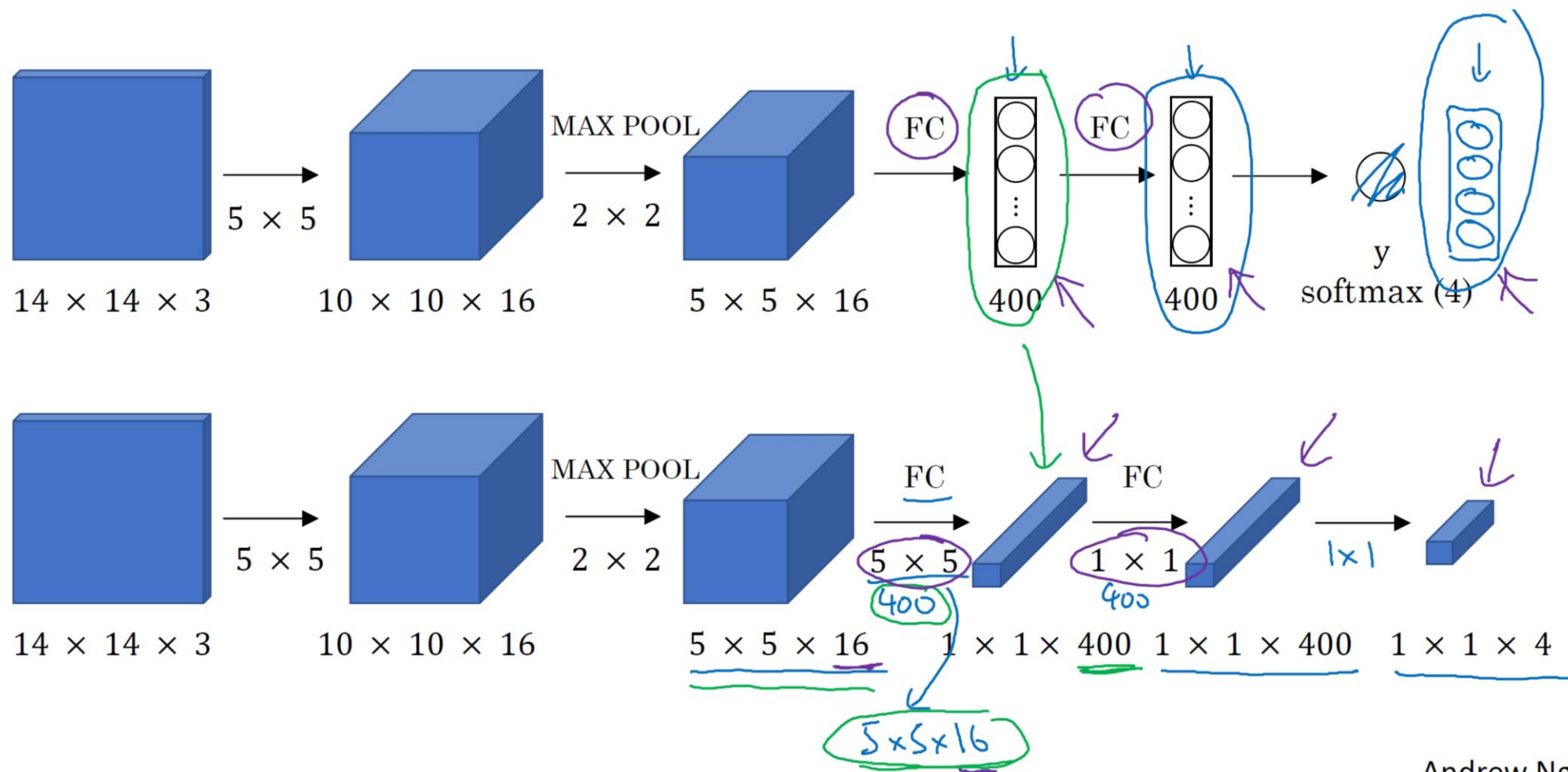


滑动窗口目标检测算法也有很明显的缺点，就是计算成本，因为你在图片中剪切出太多小方块，卷积网络要一个个地处理。如果你选用的步幅很大，显然会减少输入卷积网络的窗口个数，但是粗糙间隔尺寸可能会影响性能。反之，如果采用小粒度或小步幅，传递给卷积网络的小窗口会特别多，这意味着超高的计算成本。

2. 目标检测算法

23

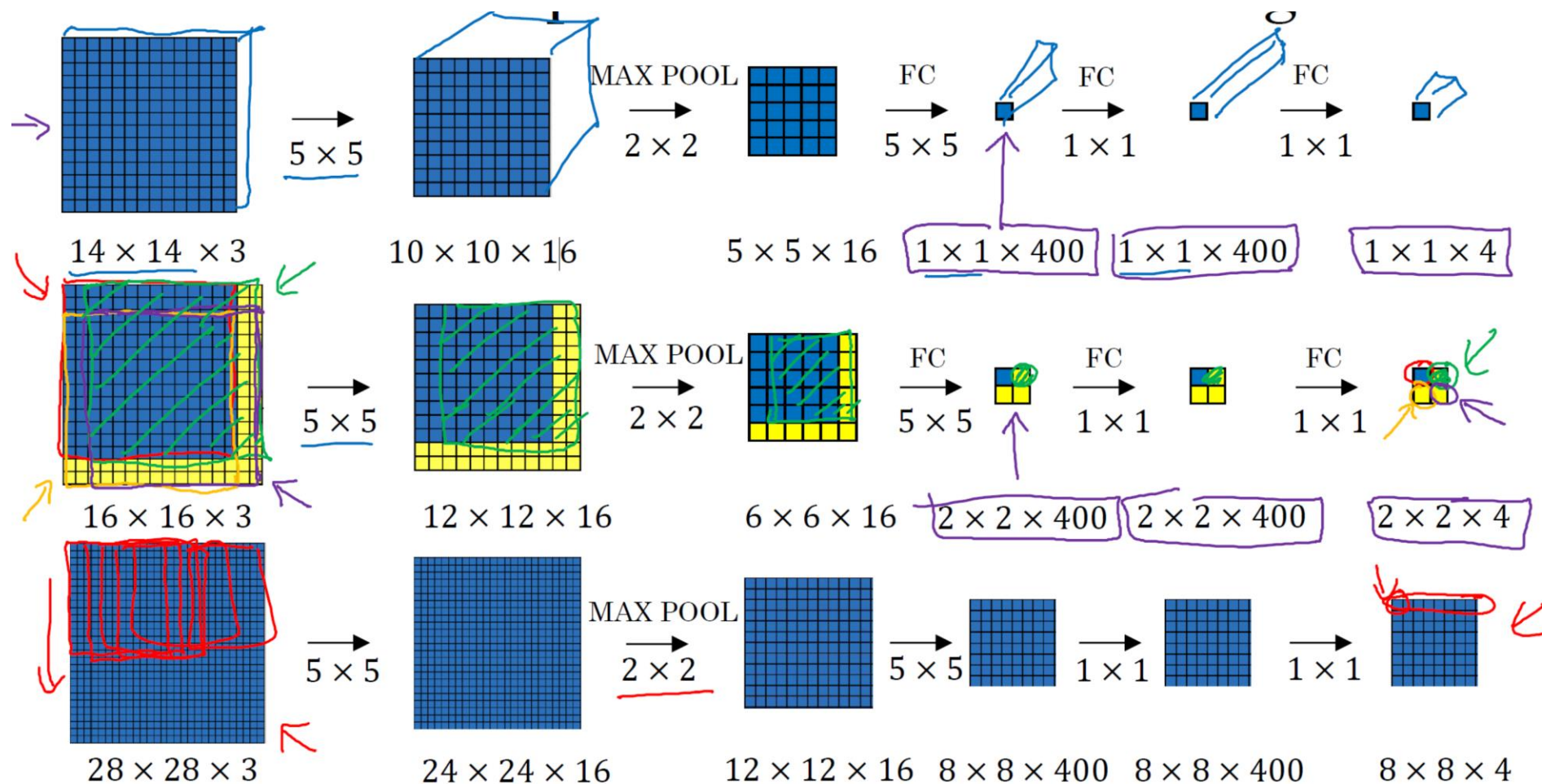
滑动窗口的卷积实现



2. 目标检测算法

24

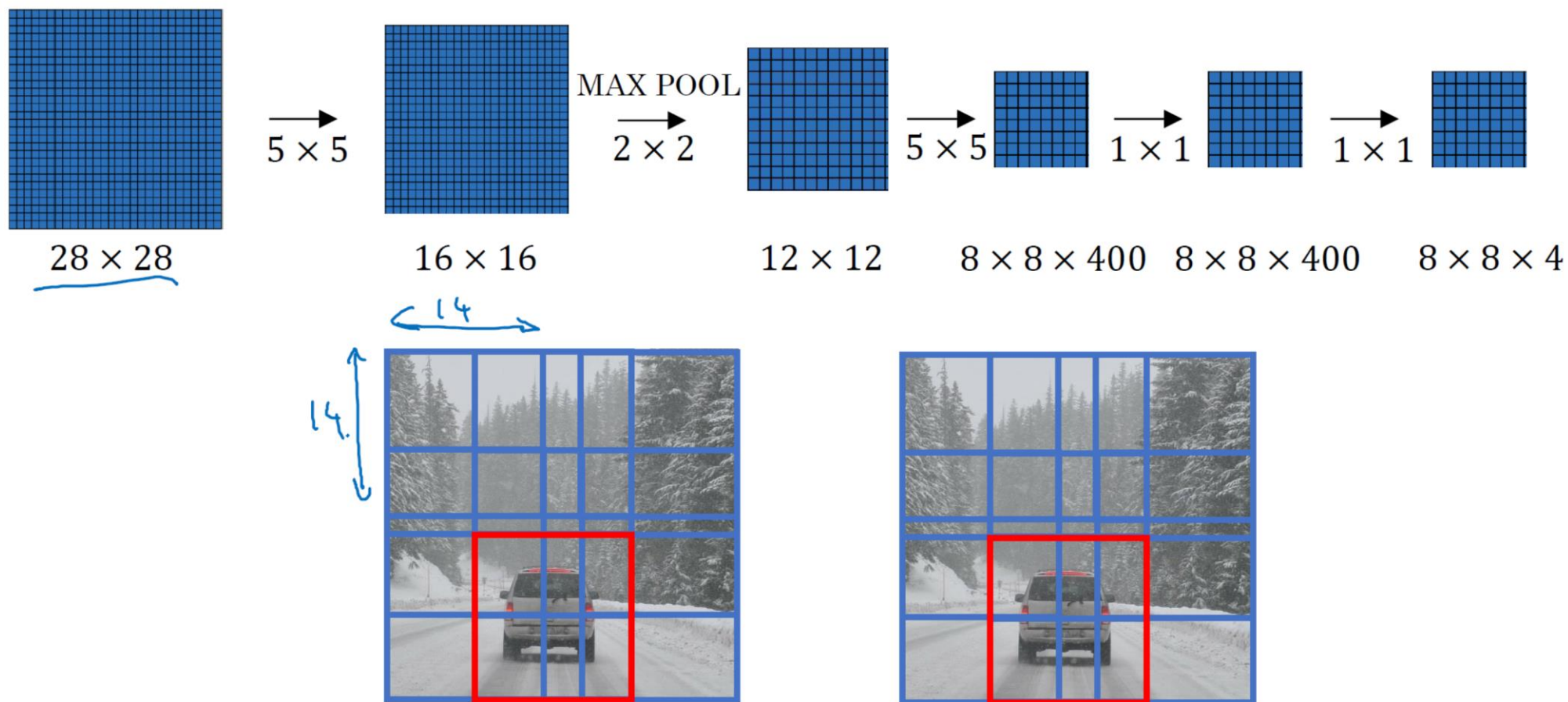
滑动窗口的卷积实现



2. 目标检测算法

25

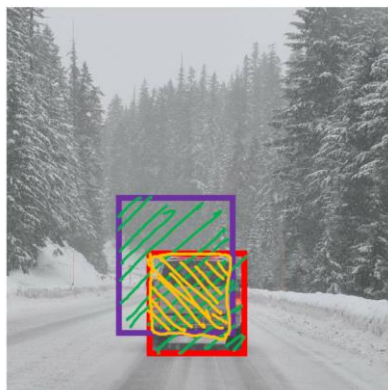
滑动窗口的卷积实现



2. 目标检测算法

26

交并比

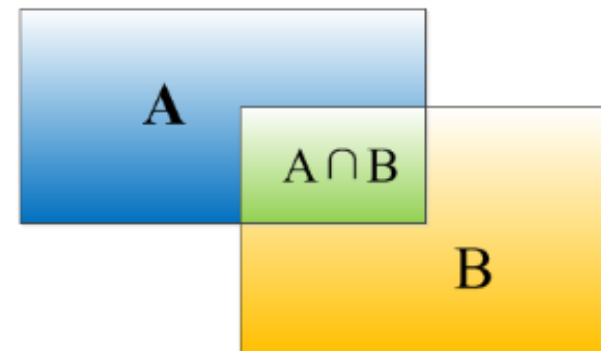


Intersection over Union (IoU)

$$= \frac{\text{Size of } \text{Intersection}}{\text{Size of } \text{Union}}$$

"Correct" if $\text{IoU} \geq 0.5$ ←

0.6 ←

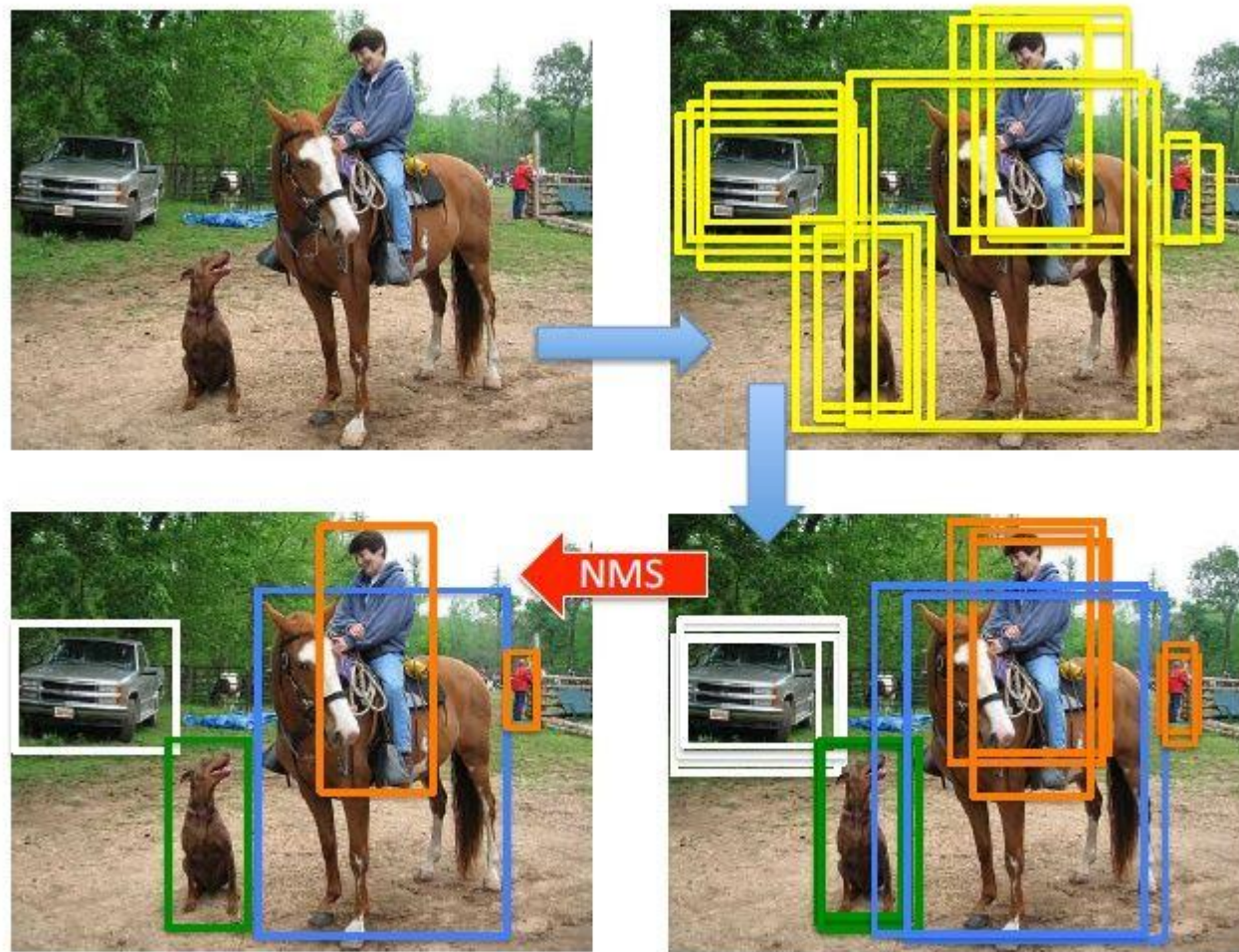


交并比: $\text{IOU} = (\text{A} \cap \text{B}) / (\text{A} \cup \text{B})$

2. 目标检测算法

27

非极大值抑制 (Non-Maximum Suppression, NMS)，顾名思义就是抑制不是极大值的元素，用于目标检测中，就是提取置信度高的目标检测框，而抑制置信度低的误检框。一般来说，用在当解析模型输出到目标框时，目标框会非常多，具体数量由anchor数量决定，其中有很多重复的框定位到同一个目标，NMS用来去除这些重复的框，获得真正的目标框。如下图所示，人、马、车上有很多框，通过NMS，得到唯一的检测框。



2. 目标检测算法

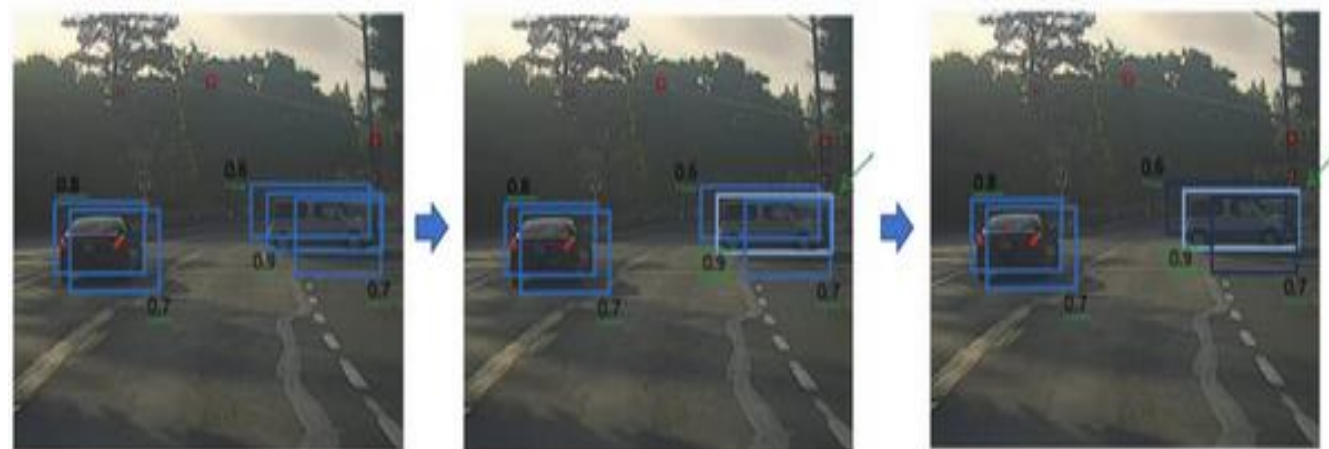
28

非极大值抑制 (Non-max suppression)



19x19

首先这个 19×19 网格上执行一下算法，你会得到 $19 \times 19 \times 8$ 的输出尺寸。不过对于这个例子来说，我们简化一下，就说你只做汽车检测，我们就去掉 c_1 、 c_2 和 c_3 ，然后假设这条线对于 19×19 的每一个输出，对于361个格子的每个输出，你会得到这样的输出预测，就是格子中有对象的概率 (p_c)，然后是边界框参数 (b_x 、 b_y 、 b_h 和 b_w)。如果你只检测一种对象，那么就没有 c_1 、 c_2 和 c_3 这些预测分量。

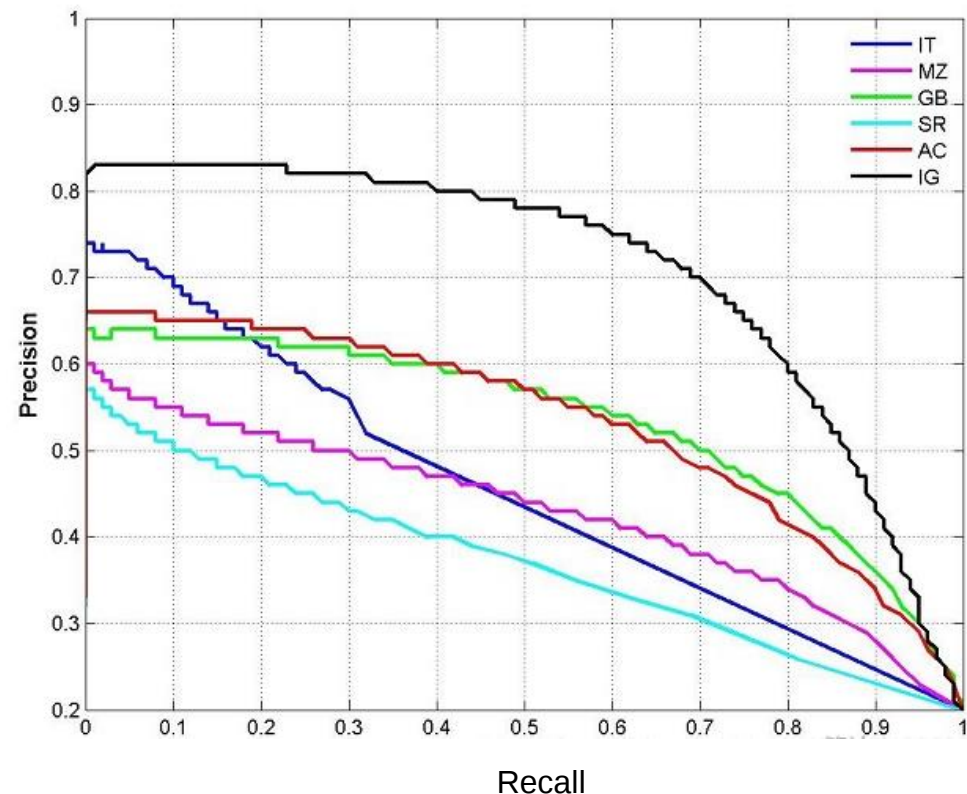


2.目标检测算法

29

mAP (Mean Average Precision)

多个类别的目标检测中，每一个类别都可以绘制一条P-R曲线，各类别AP的均值（即所有类别的AP和/类别数目）即是mAP，mAP衡量的是训练出来的模型在所有类别上的检测能力的好坏。



AP衡量的是学出来的模型在每个类别上的好坏，mAP衡量的是学出的模型在所有类别上的好坏，得到AP后mAP的计算就变得很简单了，就是取所有AP的平均值。

3.YOLO算法

30

01 目标定位

02 目标检测算法

03 YOLO算法

04 Faster RCNN算法

3.YOLO算法

31

01 目标检测概述

02 目标检测算法

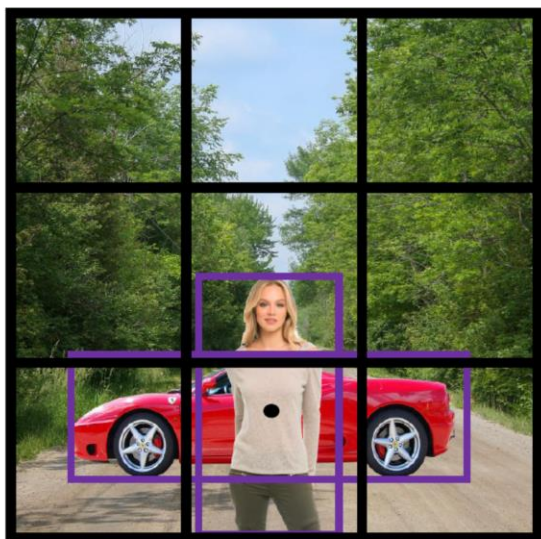
03 YOLO算法

04 Faster RCNN算法

3.YOLO算法

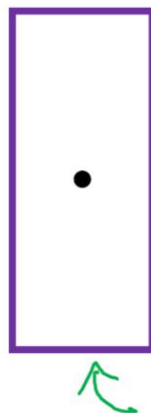
32

Anchor Boxes



$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Anchor box 1:



Anchor box 2:



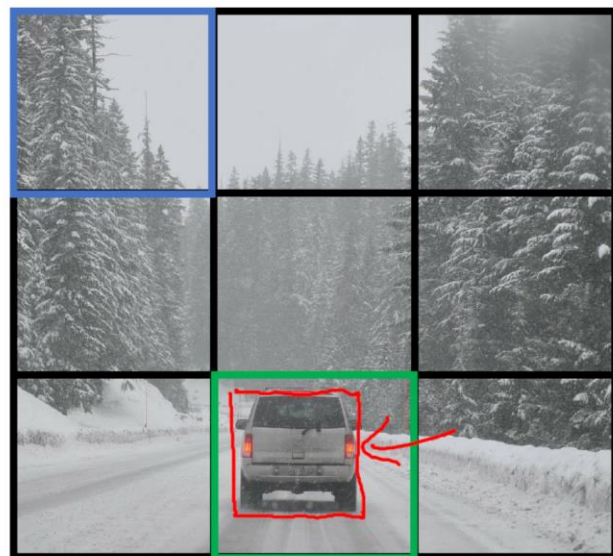
$y =$

p_c	Anchor box 1
b_x	
b_y	
b_h	
b_w	
c_1	Anchor box 2
c_2	
p_c	
b_x	
$\vdots$	
c_3	

3.YOLO算法

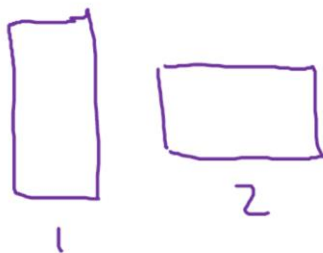
33

Training



- 1 - pedestrian
- 2 - car ←
- 3 - motorcycle

$y =$



$$\begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

y is $3 \times 3 \times 2 \times 8$

$19 \times 19 \times 16$
 $19 \times 19 \times 40$

#anchors $5 + \#classes$



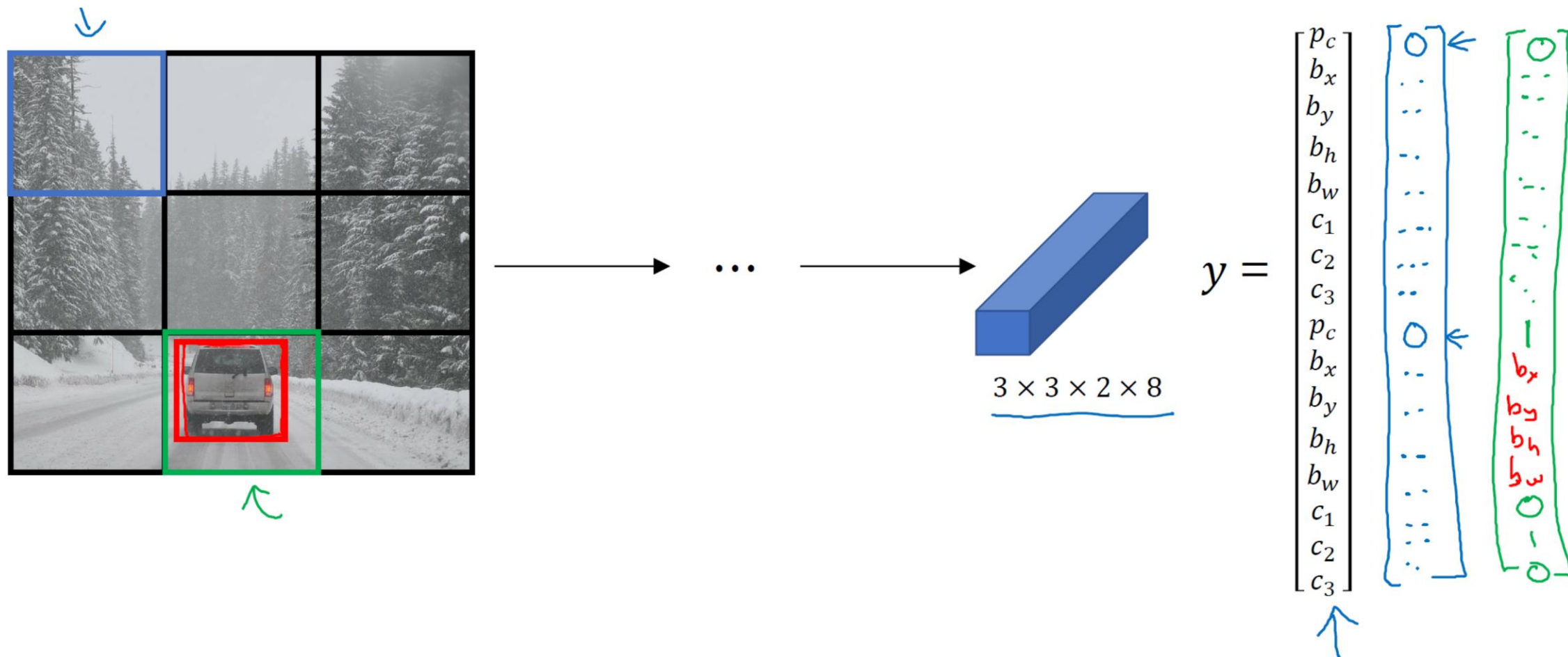
ConvNet



3.YOLO算法

34

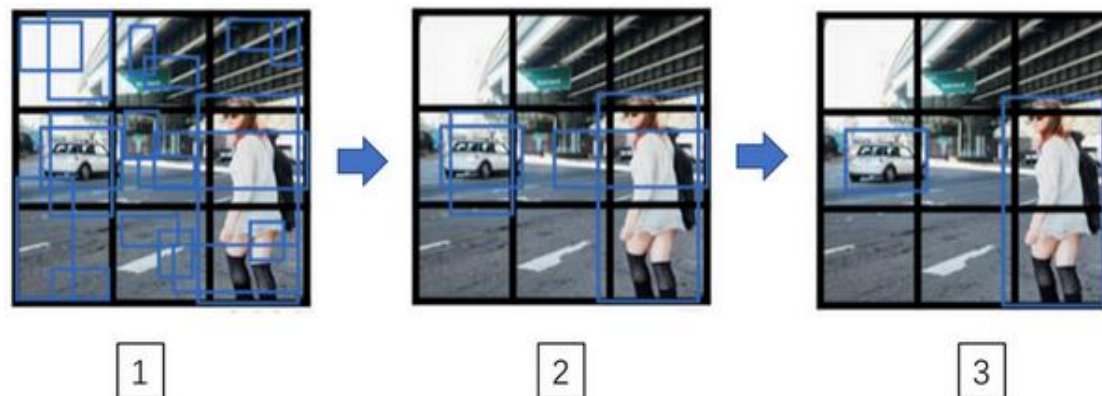
进行预测



3.YOLO算法

35

输出非极大化抑制的结果



最后你要运行一下这个非极大值抑制，为了让内容更有趣一些，我们看看一张新的测试图像，这就是运行非极大值抑制的过程。如果你使用两个**anchor box**，那么对于9个格子中任何一个都会有两个预测的边界框，其中一个的概率 p_c 很低。但9个格子中，每个都有两个预测的边界框，比如说我们得到的边界框是是这样的，注意有一些边界框可以超出所在格子的高度和宽度（编号1所示）。接下来你抛弃概率很低的预测，去掉这些连神经网络都说，这里很可能什么都没有，所以你需要抛弃这些（编号2所示）。

4.Faster RCNN算法

36

01 目标检测概述

02 目标检测算法

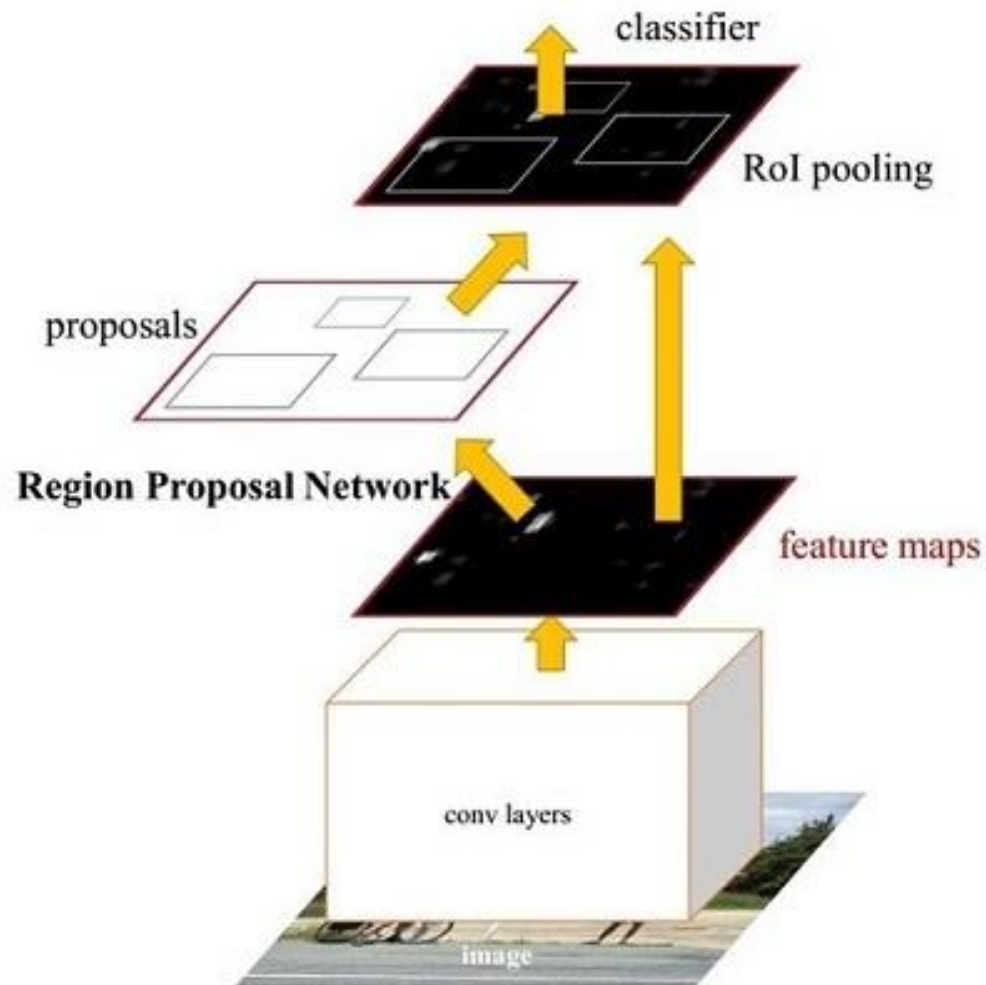
03 YOLO算法

04 Faster RCNN算法

4.Faster RCNN算法

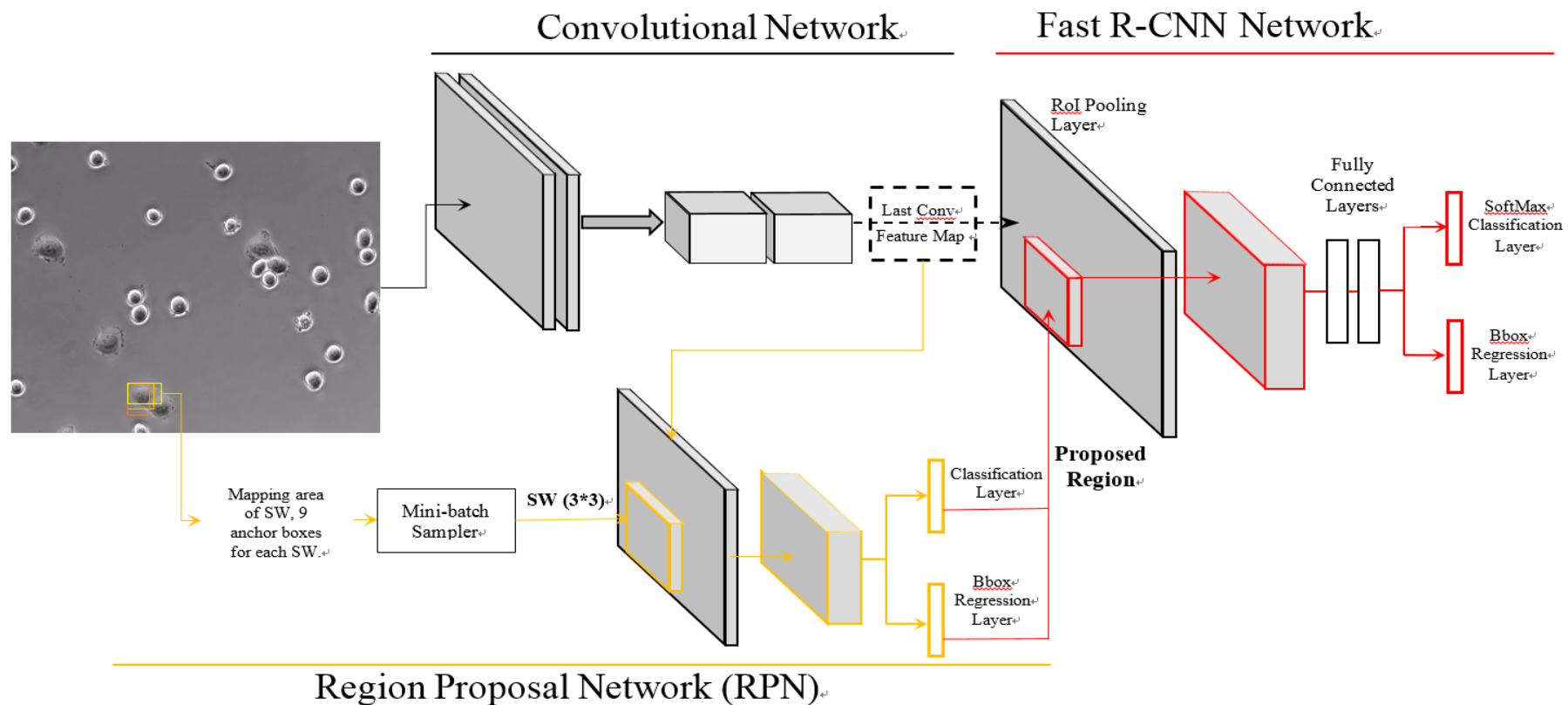
37

- 1.Conv layers
- 2.Region Proposal Networks
- 3.Roi Pooling
- 4.Classification



4.Faster RCNN算法

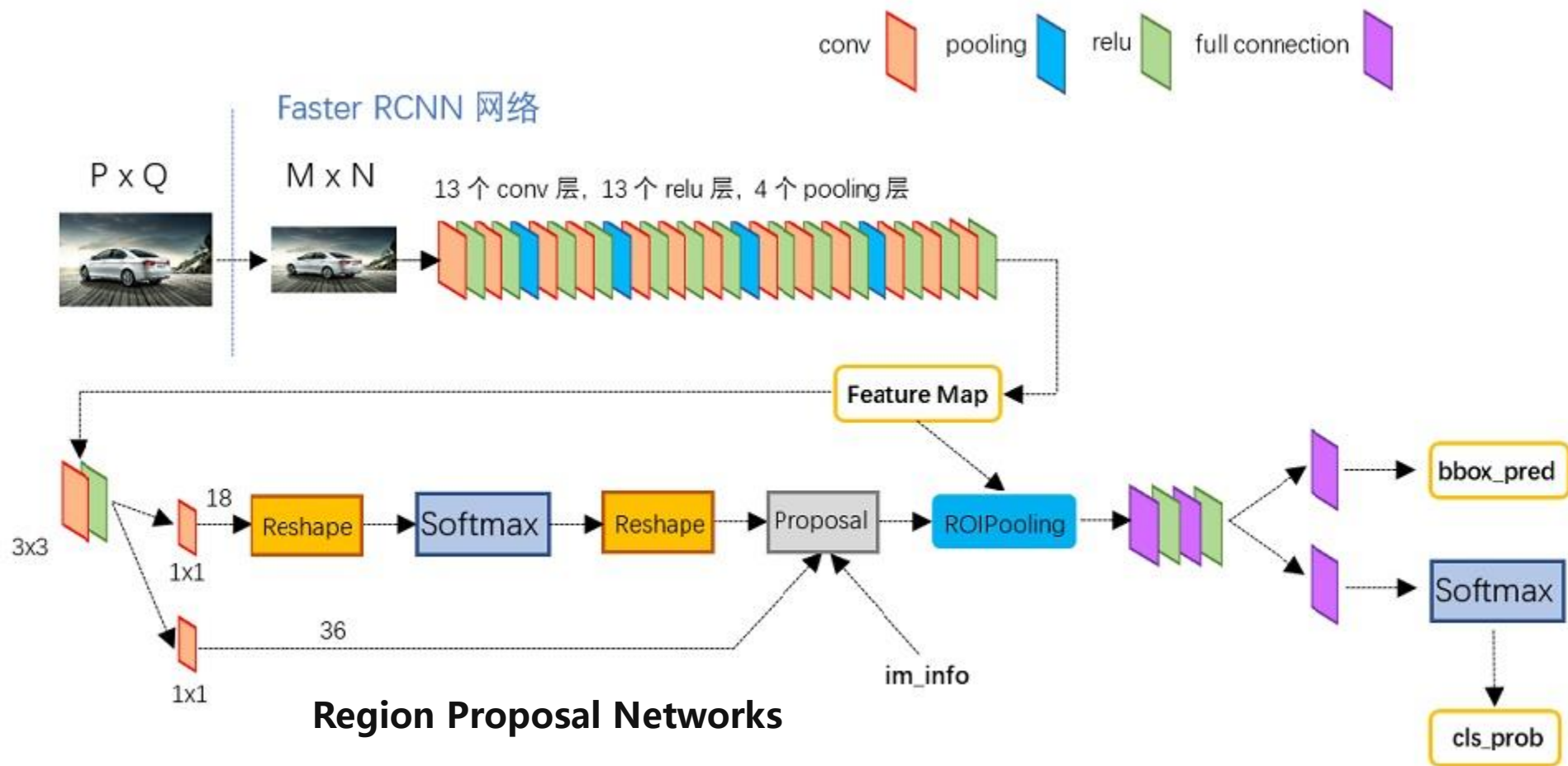
38



4.Faster RCNN算法

39

RPN网络的作用： RPN专门用来提取候选框，一方面RPN耗时少，另一方面RPN可以很容易结合到Fast RCNN中，成为一个整体。



4.Faster RCNN算法

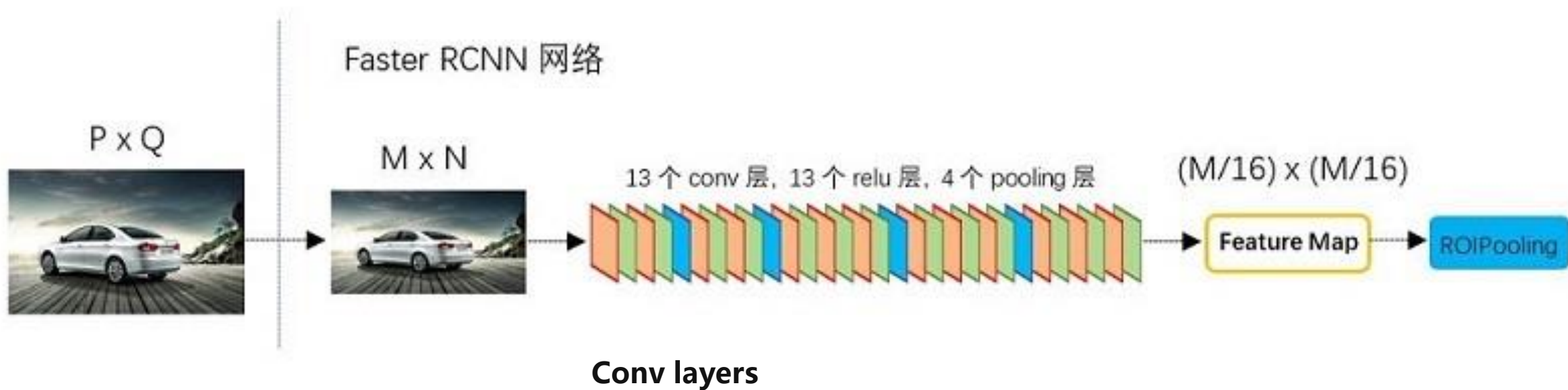
40

Faster RCNN训练步骤

- 第一步，训练RPN，该网络用ImageNet预训练的模型初始化，并端到端微调，用于生成region proposal;
- 第二步，训练Faster RCNN，由imageNet model初始化，利用第一步的RPN生成的region proposals作为输入数据，训练Fast R-CNN一个单独的检测网络，这时候两个网络还没有共享卷积层;
- 第三步，调优RPN，用第二步的Faster RCNN model初始化RPN再次进行训练，但固定共享的卷积层，并且只微调RPN独有的层，现在两个网络共享卷积层了;
- 第四步，调优Faster RCNN,由第三步的RPN model初始化Faster RCNN网络，输入数据为第三步生成的proposals。保持共享的卷积层固定，微调Faster RCNN的FC层。这样，两个网络共享相同的卷积层，构成一个统一的网络。

4.Faster RCNN算法

41



1. IAN GOODFELLOW等, 《深度学习》, 人民邮电出版社, 2017
2. Andrew Ng, <http://www.deeplearning.ai>

谢 谢!

