

动手一天学深度学习

1 基础 · 2 卷积网络 · 3 计算 · 4 计算机视觉

深度学习实训营 2019

何通

<http://1day-zh.d2l.ai>

大纲

- 图像增广
- 微调
- 目标检测
- 边界框和锚框
- 单发多框检验 (SSD) 模型
- 只看一次 (YOLO)
- 区域卷积神经网络 (R-CNN)
- 计算机视觉训练技巧

图像增广



$p=1.0$



size_percent=0.30



$p=0.50$



cutoff=0.00

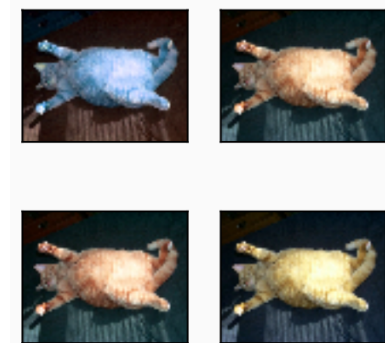
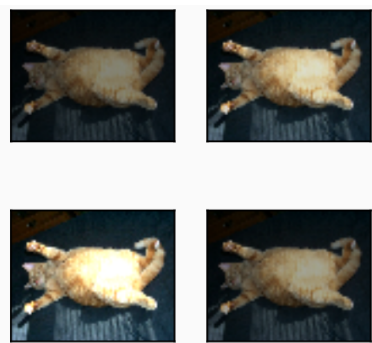
CES'19 上的真实故事

- 一家初创公司要演示一款智能自动售货机，通过相机镜头识别客户挑选的物品。
- 但演示失败了，因为展厅有不同的：
 - 光照色温
 - 桌子上有光反射
- 他们通宵加班：
 - 重新收集数据并训练新模型
 - 订购了桌布

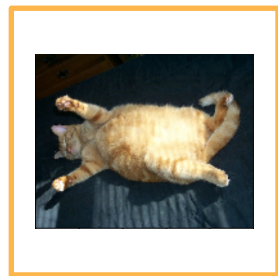


数据增光

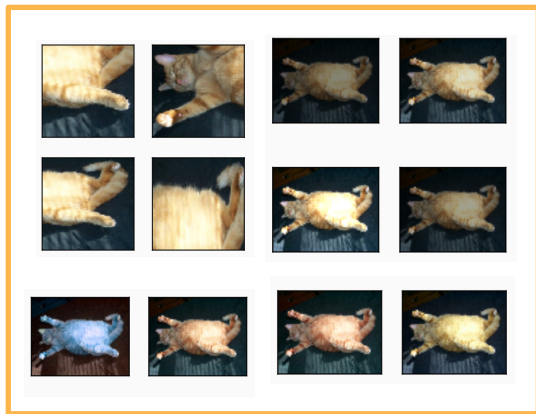
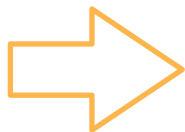
- 现有数据集基础上，增广为具有更多多样性的数据集：
- 在训练数据集中添加各种背景噪音
- 转换为其他图像：改变颜色，改变形状



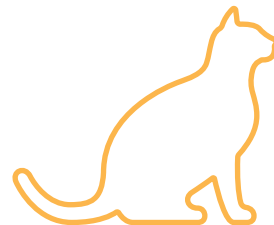
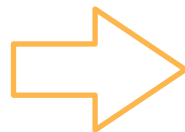
Training with Augmented Data



原始数据集



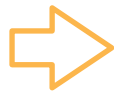
增广后数据集



模型

Flip

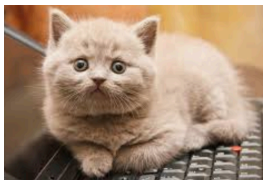
- 左右翻转



- 上下翻转

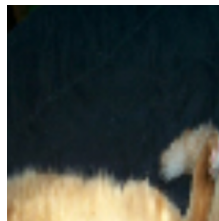


- 不一定都有效



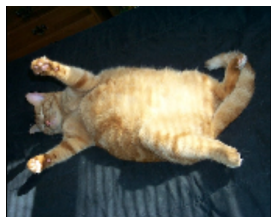
裁剪

- 从图像中随机裁剪一块区域，然后调整其大小
 - 随机宽高比（例如 $[3/4, 4/3]$ ）
 - 随机的区域
 - 随机区域大小（例如 $[8\%, 100\%]$ ）

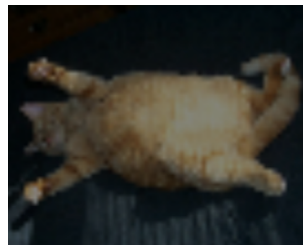


变色

- 调整色调，饱和度和亮度（例如 $[0.5, 1.5]$ ）



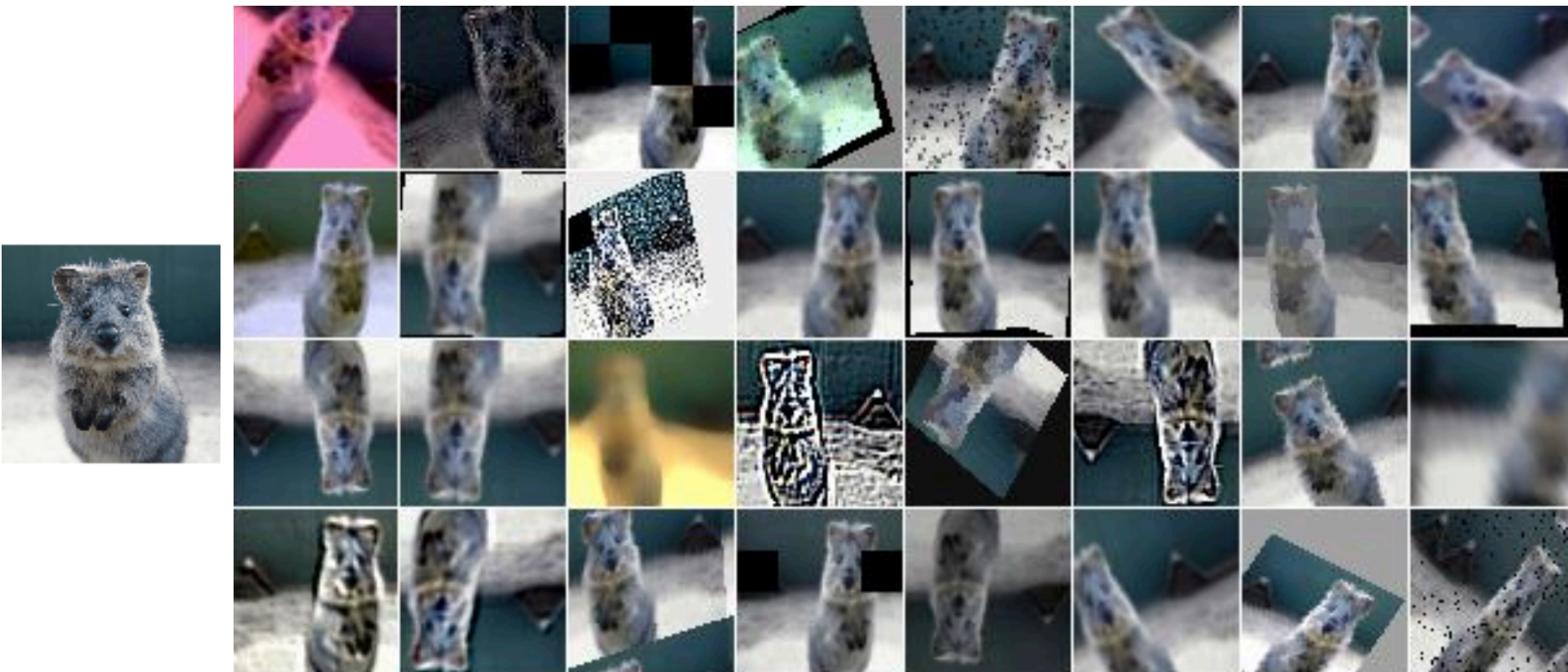
亮度



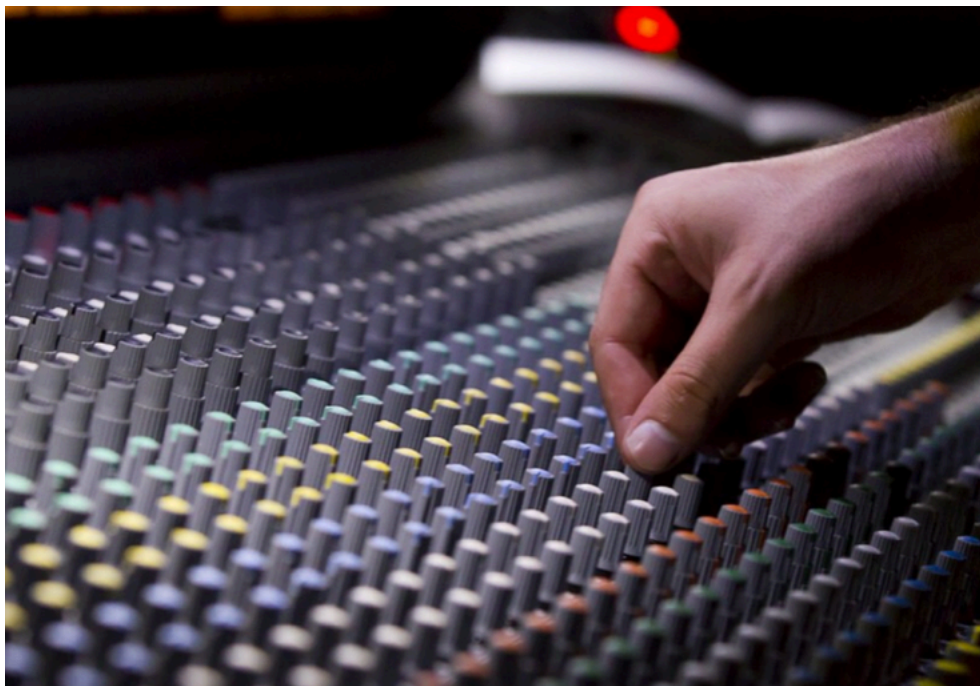
色调



更多的增广方法



微调



标记数据集非常昂贵

流行数据集



数据点

1.2 M

50K

60 K

类别

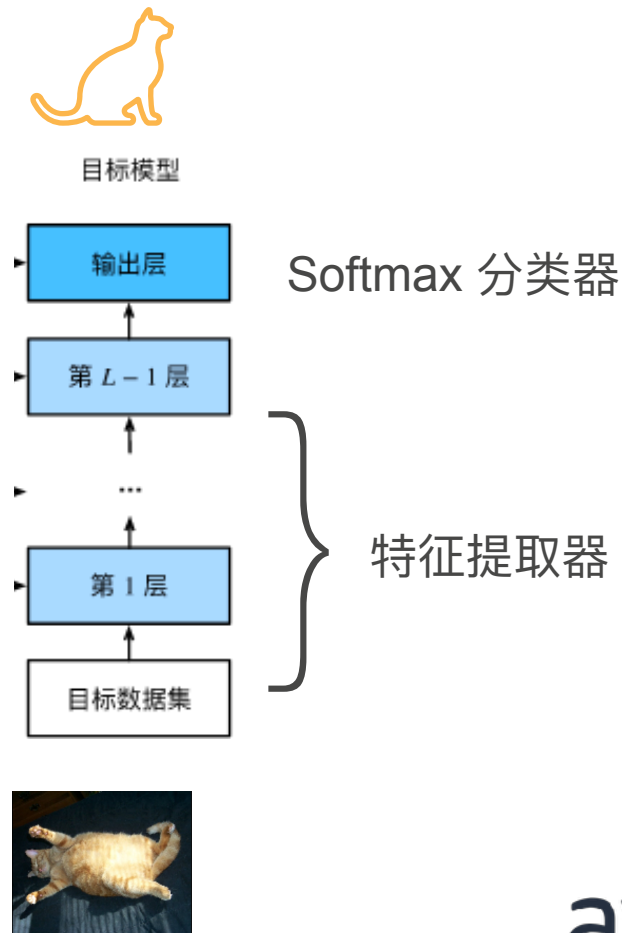
1,000

100

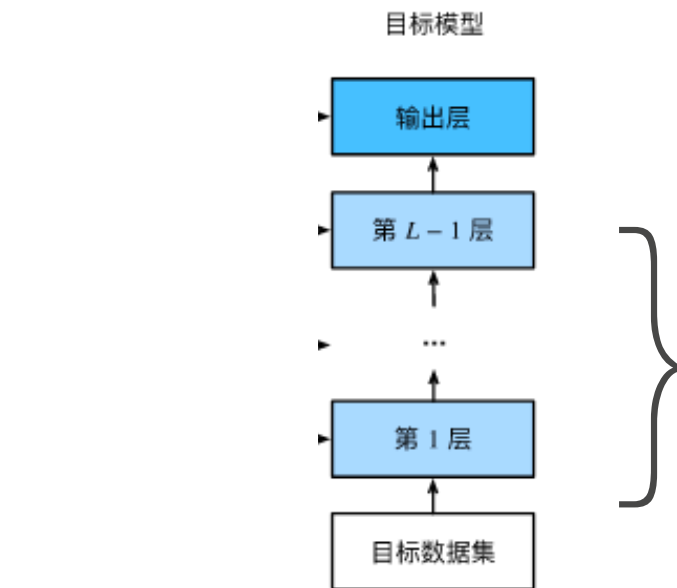
10

网络结构

- 一个神经网络模型可以大致分为两部分
 - 特征提取器将原始像素映射为线性可分离的特征
 - 用线性分类做决定



微调



不直接使用原分类器的最后一层参数，因为标签可能不一致

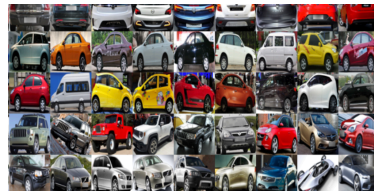


可能仍然是很好的图像特征提取器

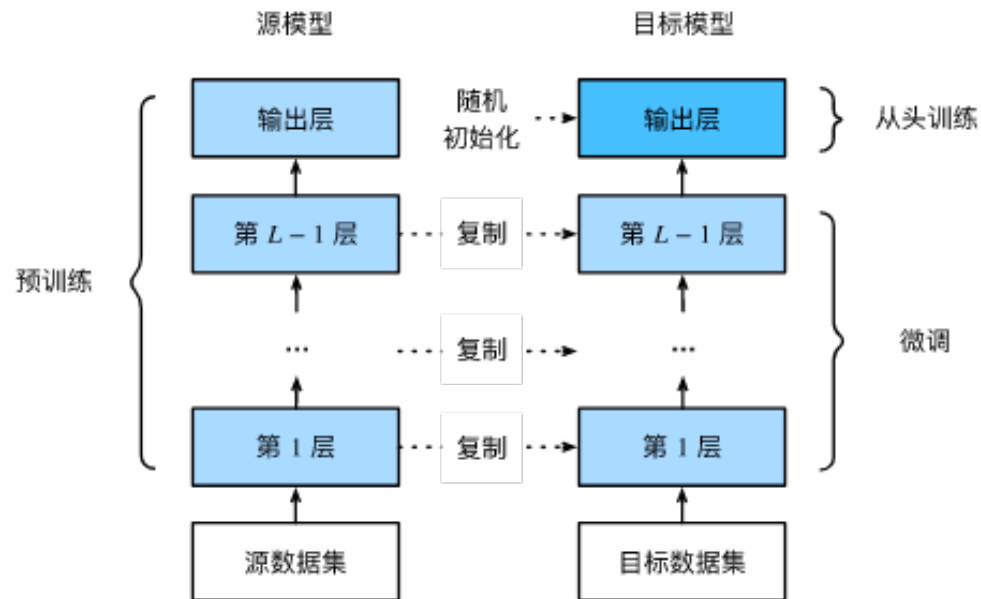
源数据集



目标数据集



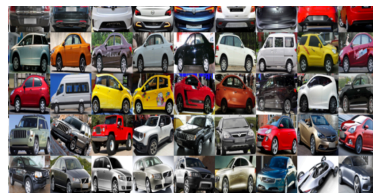
微调中的权重初始化



源数据集



目标数据集



微调训练

- 在已有神经网络上微调目标数据集，但具有强大的正规化
 - 使用较小的学习率
 - 使用较少的迭代周期
- 如果源数据集比目标数据集更复杂，则微调通常会得到更高质量的模型

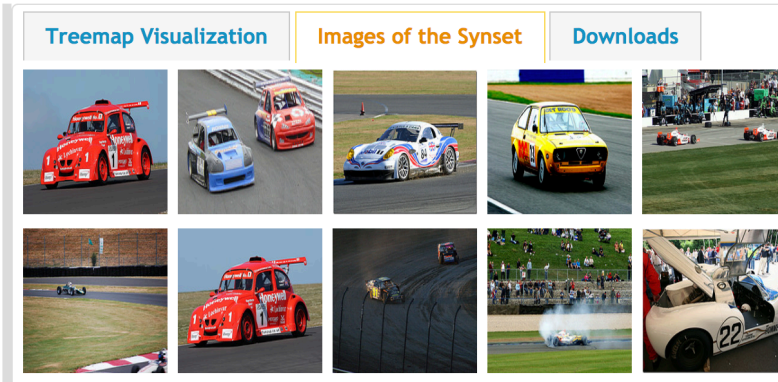
重用已有分类器的参数

- 源数据集可能包含目标数据集中的某些类别
- 在初始化期间使用来自预训练模型的相应权重向量



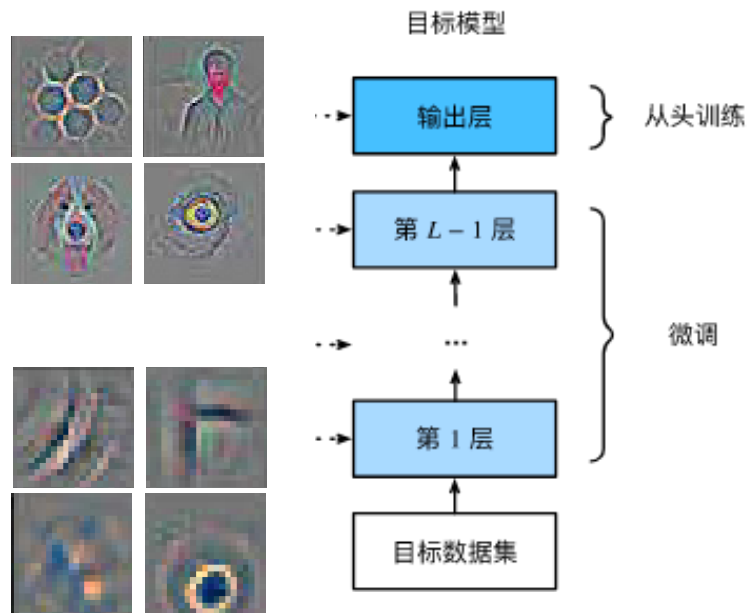
Racer, race car, racing car

A fast car that competes in races



固定一些层的参数

- 神经网络学习分层特征表示
 - 低层特征是通用的
 - 高层特征与数据集中的对象更相关
- 在微调期间固定低层的参数
 - 一个有用的正则项



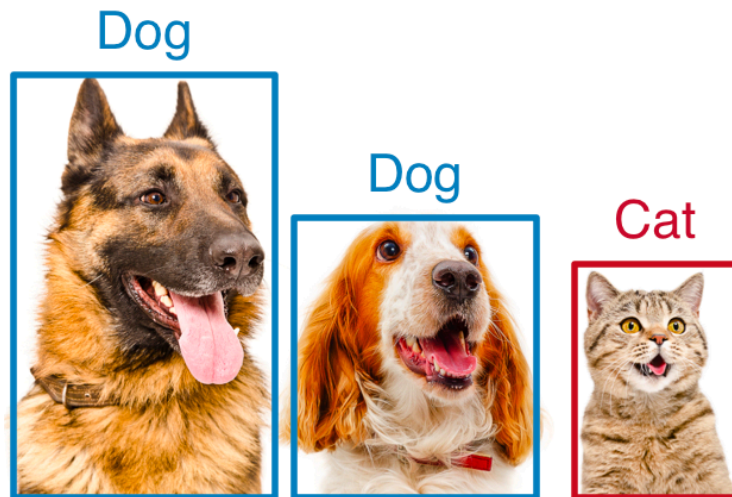
目标检测

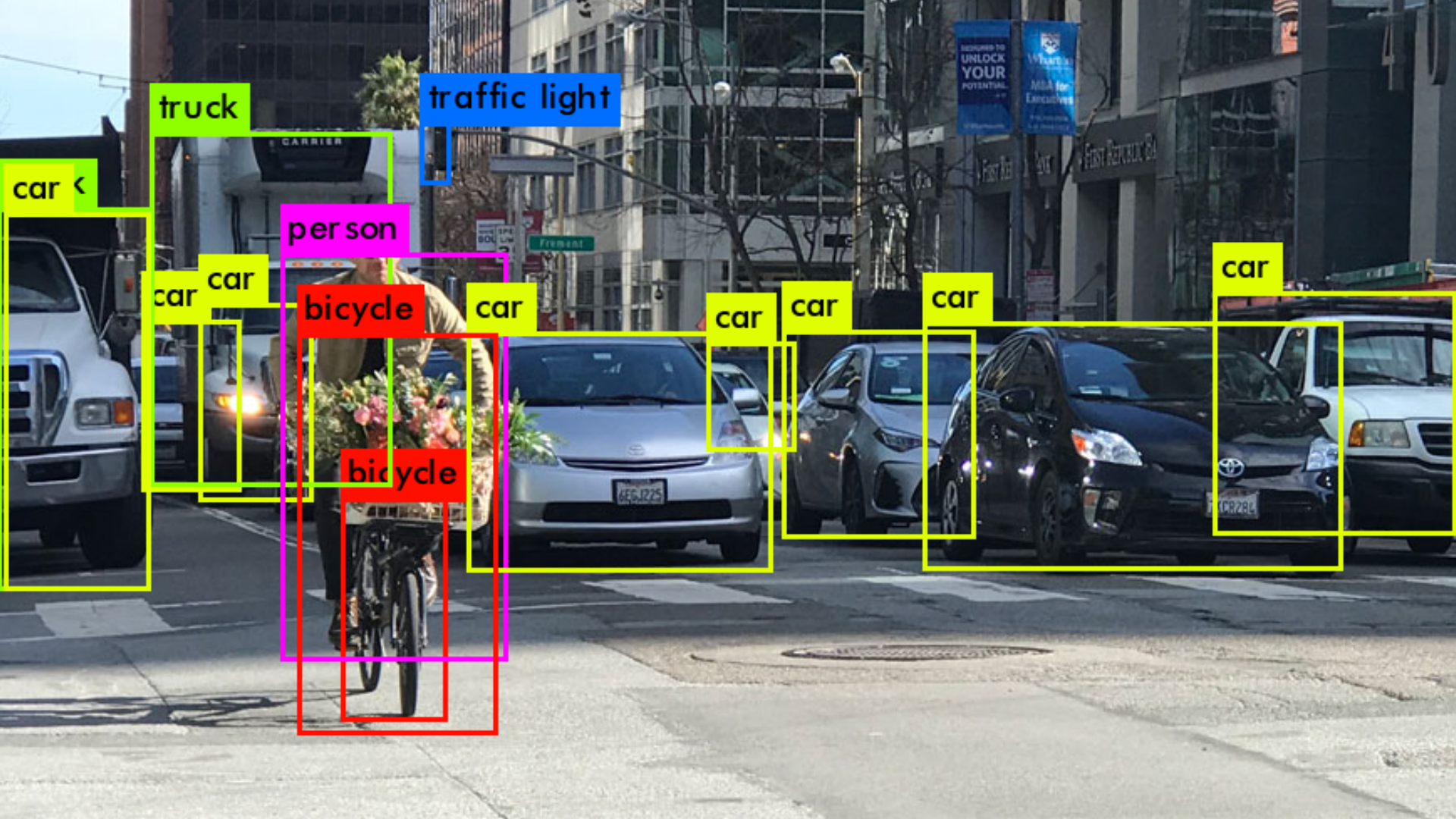


图像分类



目标检测





truck

traffic light

person

bicycle

bicycle

car

car

car

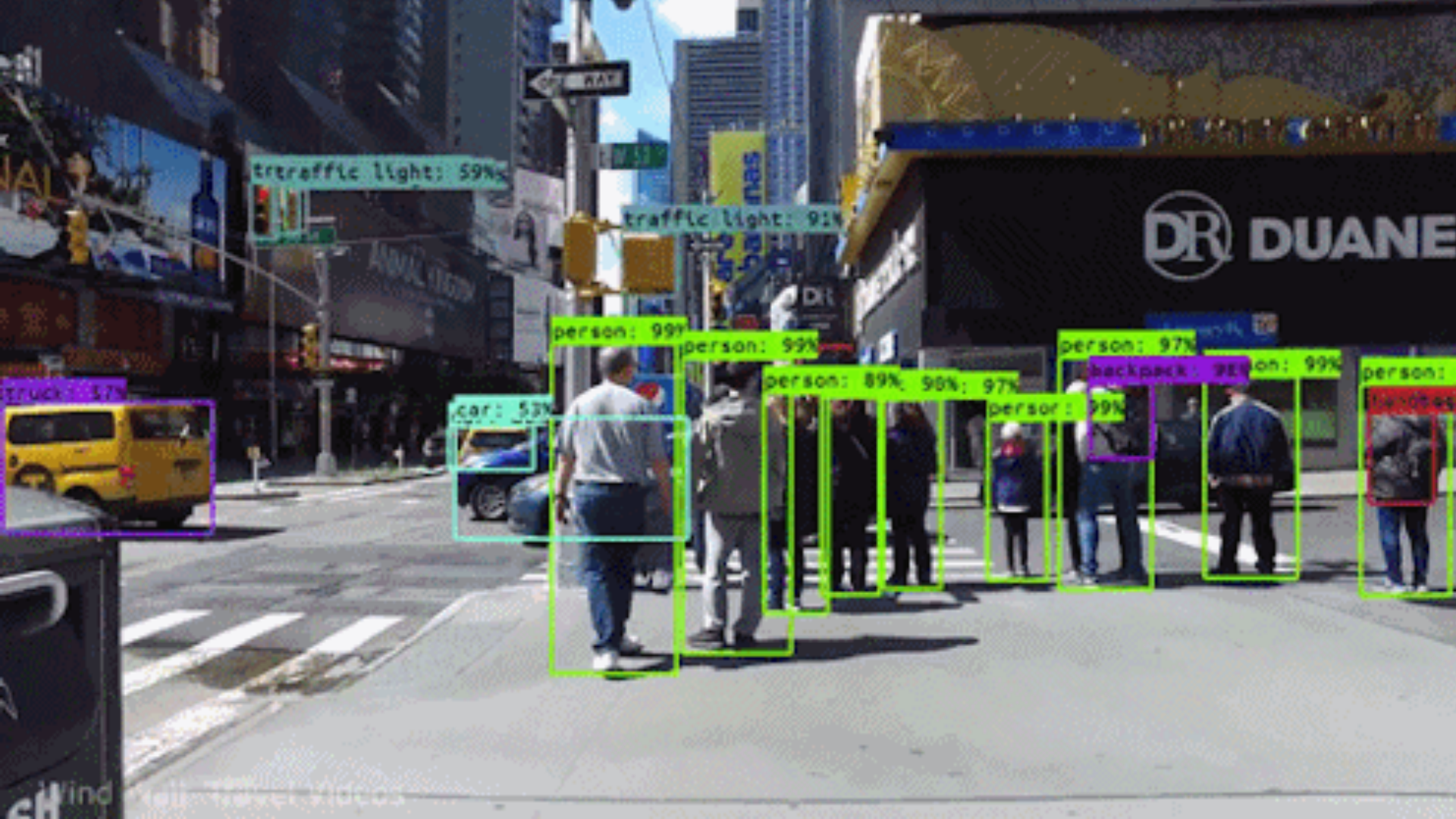
car

car

car

car

car <



truck: 97%

car: 53%

person: 99%

person: 99%

person: 89% 98% 97%

person: 99%

person: 97%

backpack: 98%

person: 99%

person: 99%

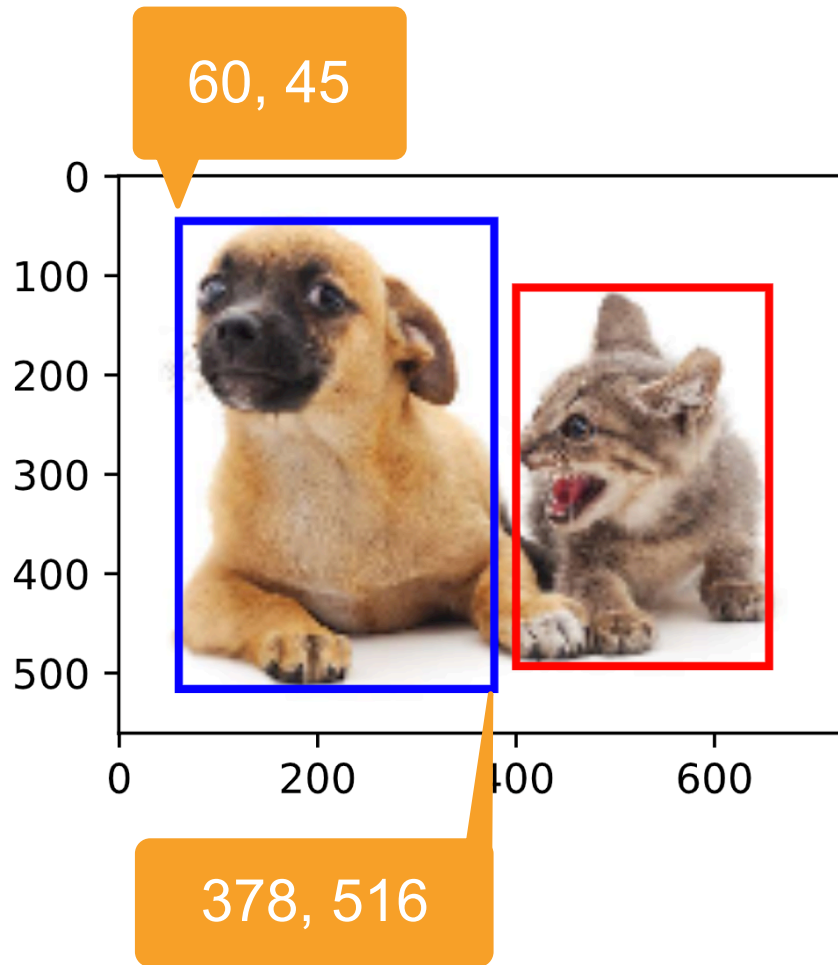
person: 99%

边界框和锚框



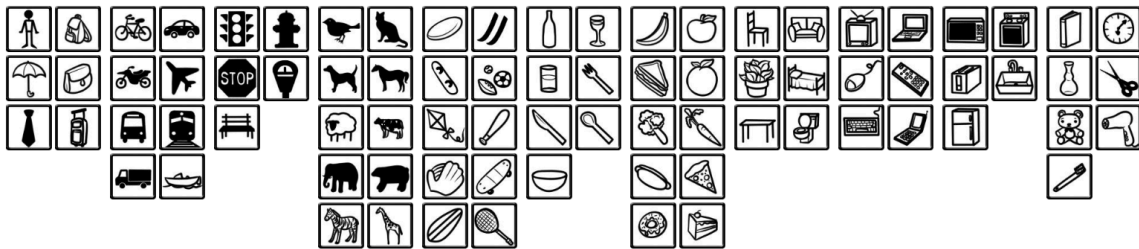
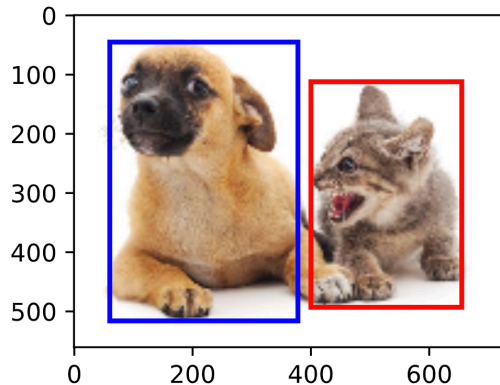
边界框

- 用4个数定义一个边界框
 - (左上 x, 左上 y, 右下 x, 右下 y)
 - (左上 x, 左上 y, 宽, 高)



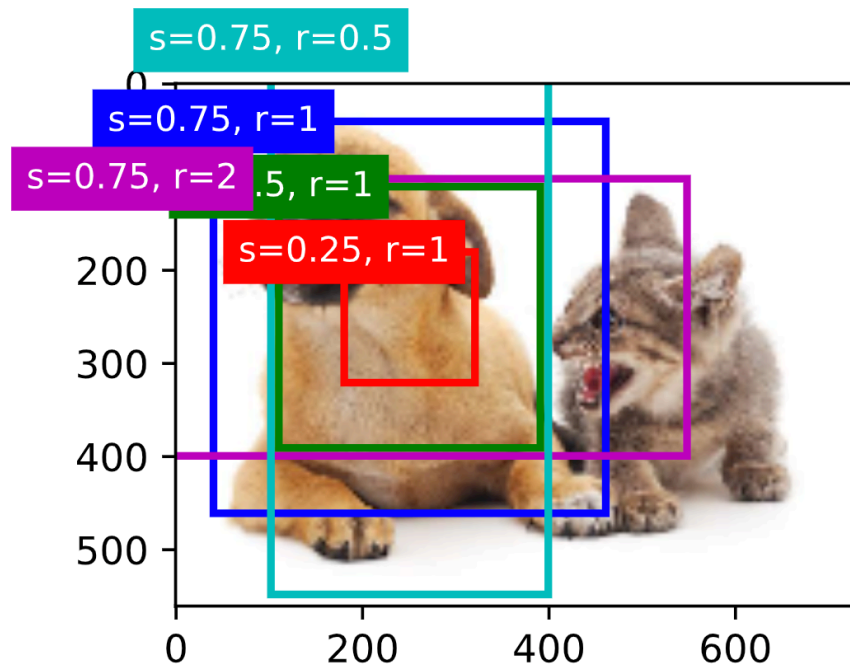
目标检测数据集

- 每行都有一个目标物体
 - 图像名称，物体类别，边界框
 - COCO (cocodataset.org)
 - 80 类物体
 - 330K 张图
 - 1.5M 个目标物体
- 



锚框

- 目标物体检测算法
 - 选择多个区域，称为锚框
 - 预测每个锚框是否包含目标物体
 - 如果是，则再预测从锚框到实际边界框的偏移量

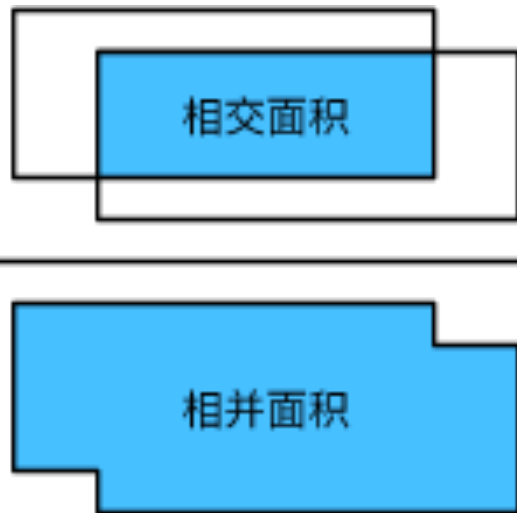


交并比

- 交并比 (IoU) 测量两个框之间的相似性：
 - 0 表示不重叠
 - 1 表示完全相同
- Jaccard系数, 给定集合A和B:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

IoU =



将标签分配给锚框

- 每个锚框都是一个训练样例
- 标记每个锚框的内容为
 - 背景，或
 - 与某个边界框有关
- 可能会生成大量的锚框
 - 大部分锚框都是背景框

将标签分配给锚框

真实边界框索引

	1	2	3	4
1				
2			x_{23}	
3				
4				
5				
6				
7	x_{71}			
8				
9				

假设矩阵 X 中最大值为 x_{23} ，我们将分配真实边界框 B_3 给锚框 A_2 。

然后，丢弃矩阵中第2行和第3列的所有元素，找出剩余阴影部分的最大元素 x_{71} ，为锚框 A_7 分配真实边界框 B_1 。

将标签分配给锚框

真实边界框索引

	1	2	3	4
1				
2			x_{23}	
3				
4				
5				
6				
7	x_{71}			
8				
9				

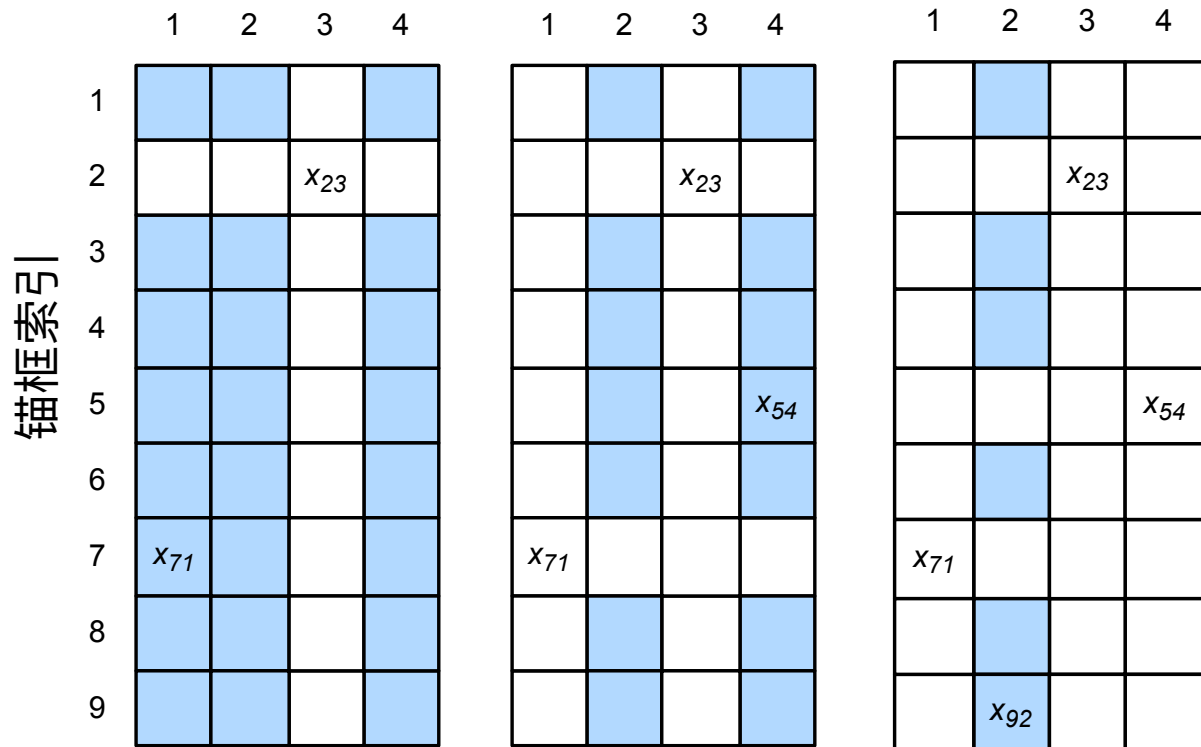
接着，丢弃矩阵中第7行和第1列的所有元素，

找出剩余阴影部分的最大元素 x_{54} ，

为锚框 A_5 分配真实边界框 B_4 。

将标签分配给锚框

真实边界框索引



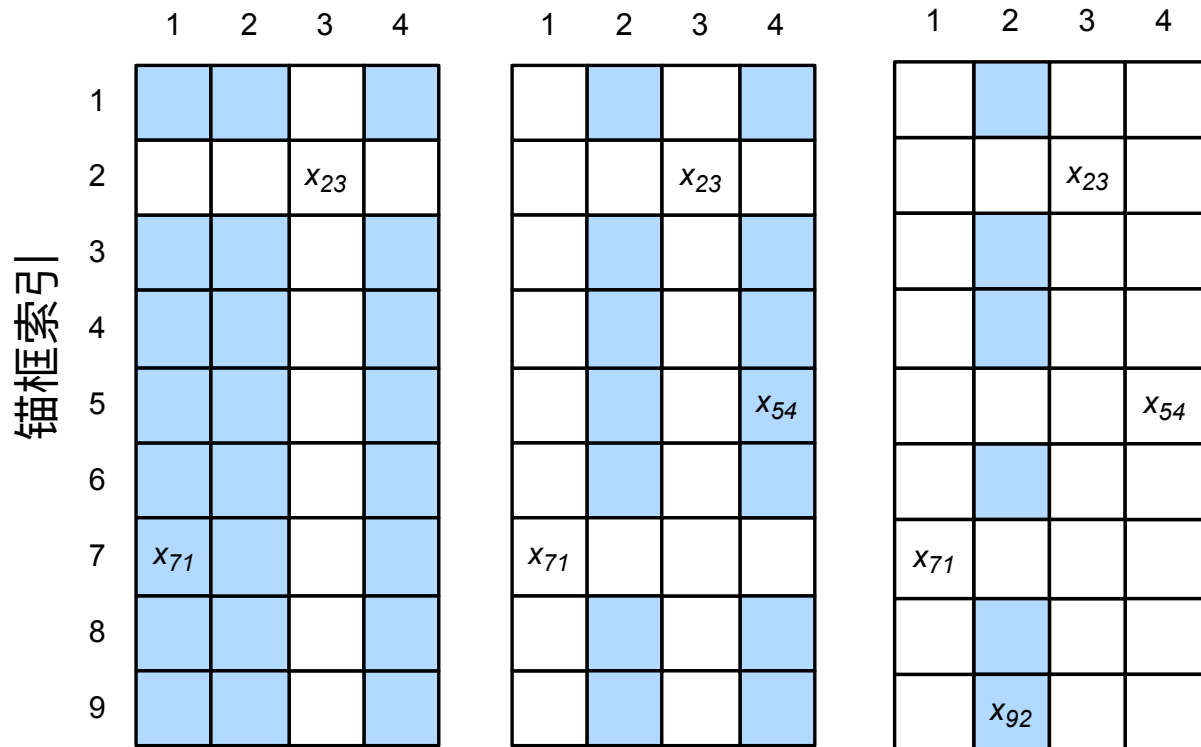
最后，丢弃矩阵中第5行和第4列的所有元素，

找出剩余阴影部分的最大元素 x_{92} ，

为锚框 A_9 分配真实边界框 B_2 。

将标签分配给锚框

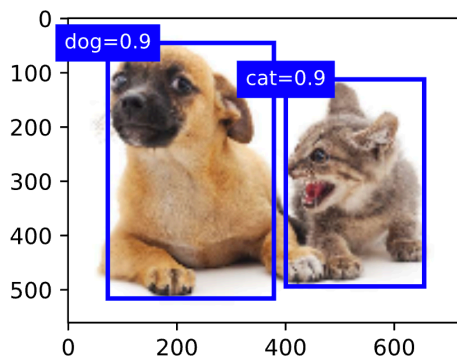
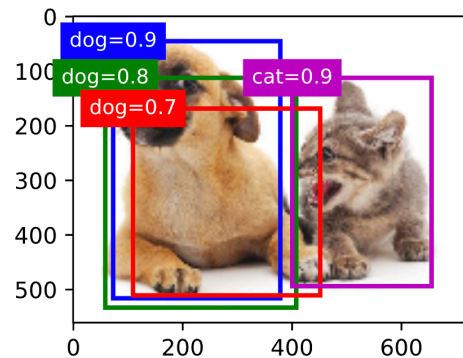
真实边界框索引



之后，我们只需遍历除去 $A_2 A_5 A_7 A_9$ 的剩余锚框，并根据阈值判断是否为剩余锚框分配真实边界框。

输出预测边界框 (NMS)

- 非极大值抑制 (non-maximum suppression, NMS)
 - 每个锚框生成一个边界框预测
 - 选择得分最高的那个预测
 - 计算所有其他预测与所选预测的 IoU, 将值大于 θ 的预测删除
 - 重复上述过程, 直到所有锚框都被选中或者删除



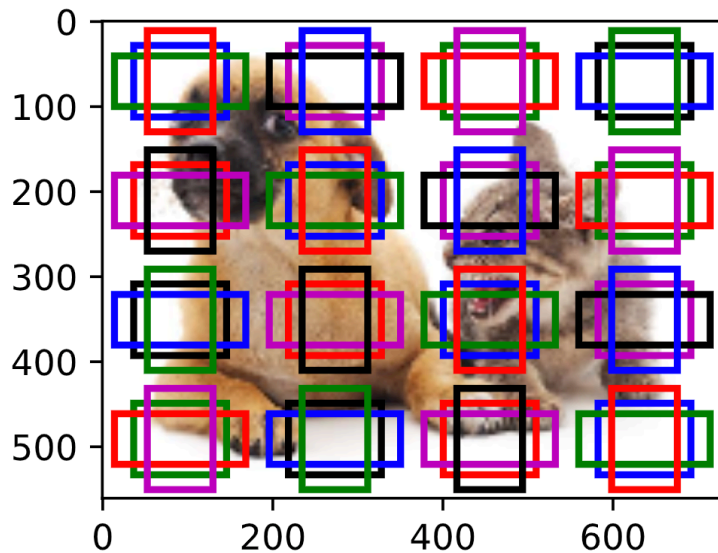
单发多框检测 (SSD)



生成锚框

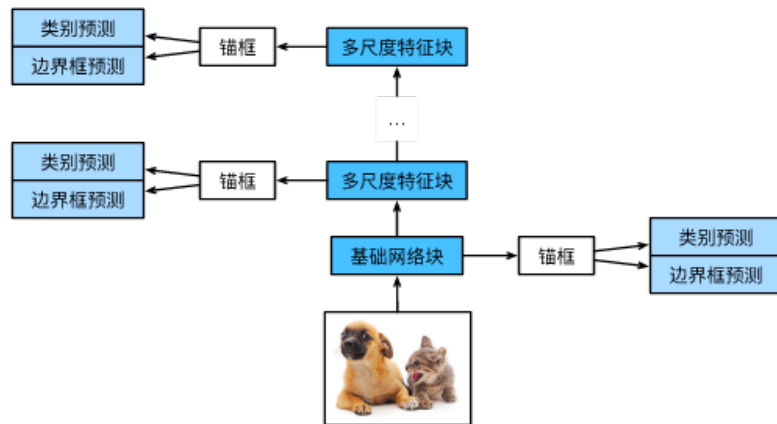
- 对每个像素生成以该像素为中心的多个锚框
- 给定 n 个 $s_1, \dots, s_n$ 和 m 个比率 $r_1, \dots, r_m$, 生成 $n+m-1$ 个锚框

$(s_1, r_1), (s_2, r_1), \dots, (s_n, r_1), (s_1, r_2), \dots, (s_1, r_m)$

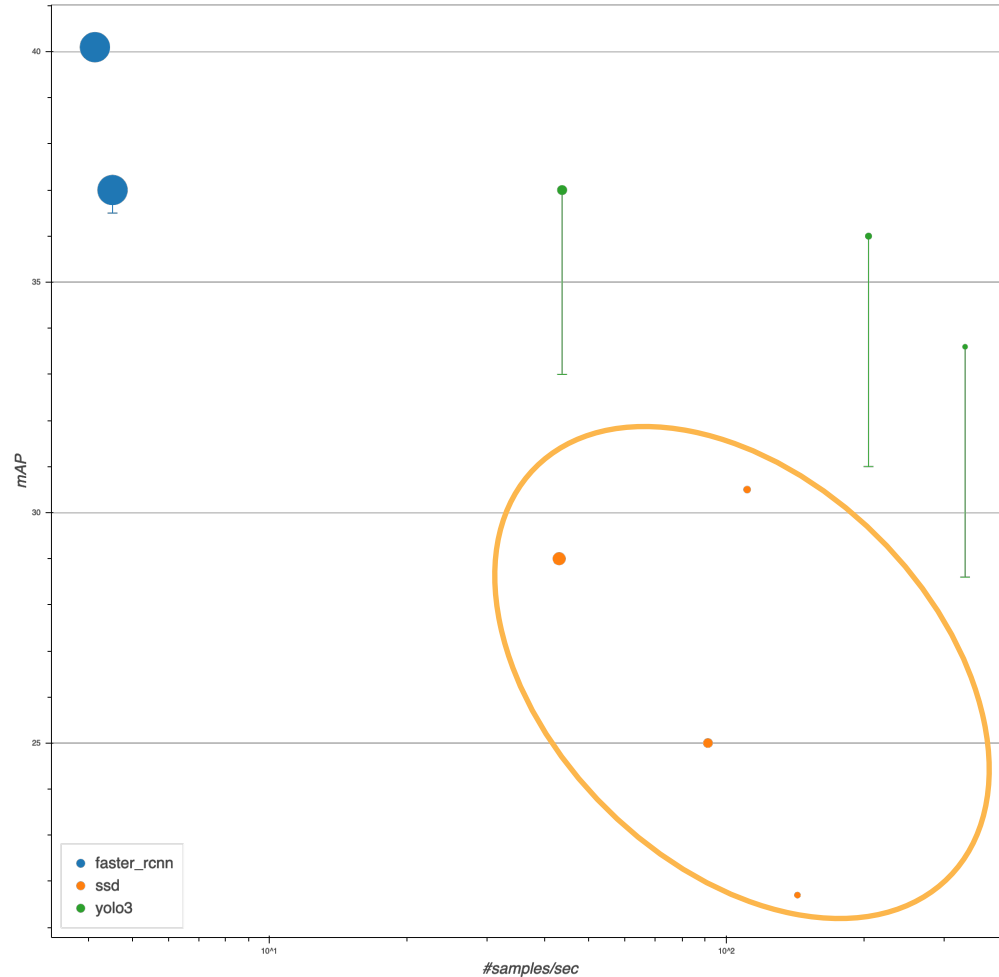


单发多框检测（SSD）模型

- 单发多框检测是一个多尺度的目标检测模型
 - 先是基础网络从原始图像中抽取特征，使它输出的高和宽较大，可以用来检测尺寸**较小**的目标
 - 接下来的每个多尺度特征块将上一层提供的特征图的高和宽缩小（如减半），并使特征图中每个单元在输入图像上的感受野变得更广阔
 - 越靠近顶部的多尺度特征块输出的特征图越小，故而基于特征图生成的锚框也越少，加之特征图中每个单元感受野越大，因此更适合检测尺寸**较大**的目标预测每个锚框的类和边界框



https://gluon-cv.mxnet.io/model_zoo/detection.html



SSD

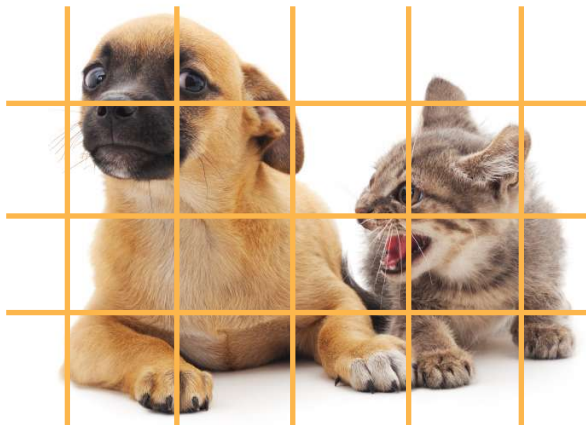
只看一次 (YOLO)



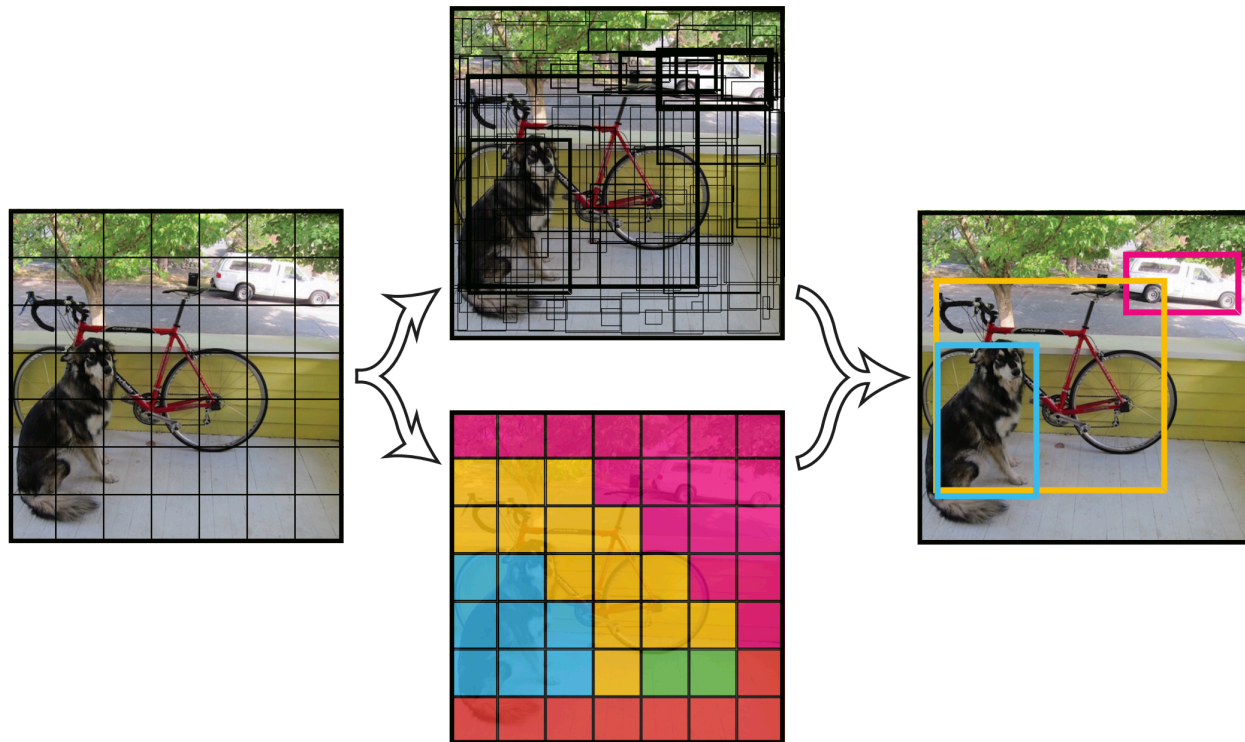
**KEEP
CALM
BECAUSE
#YOLO**

YOLO

- 锚框在 SSD 中高度重叠
- YOLO 将输入图像均匀地切割成 $S \times S$ 个锚框中心
- 每个锚框预测
 - B个边界框坐标和框内是否有物体
 - 每个类别的概率
- V2 和 V3 增加了更多改进

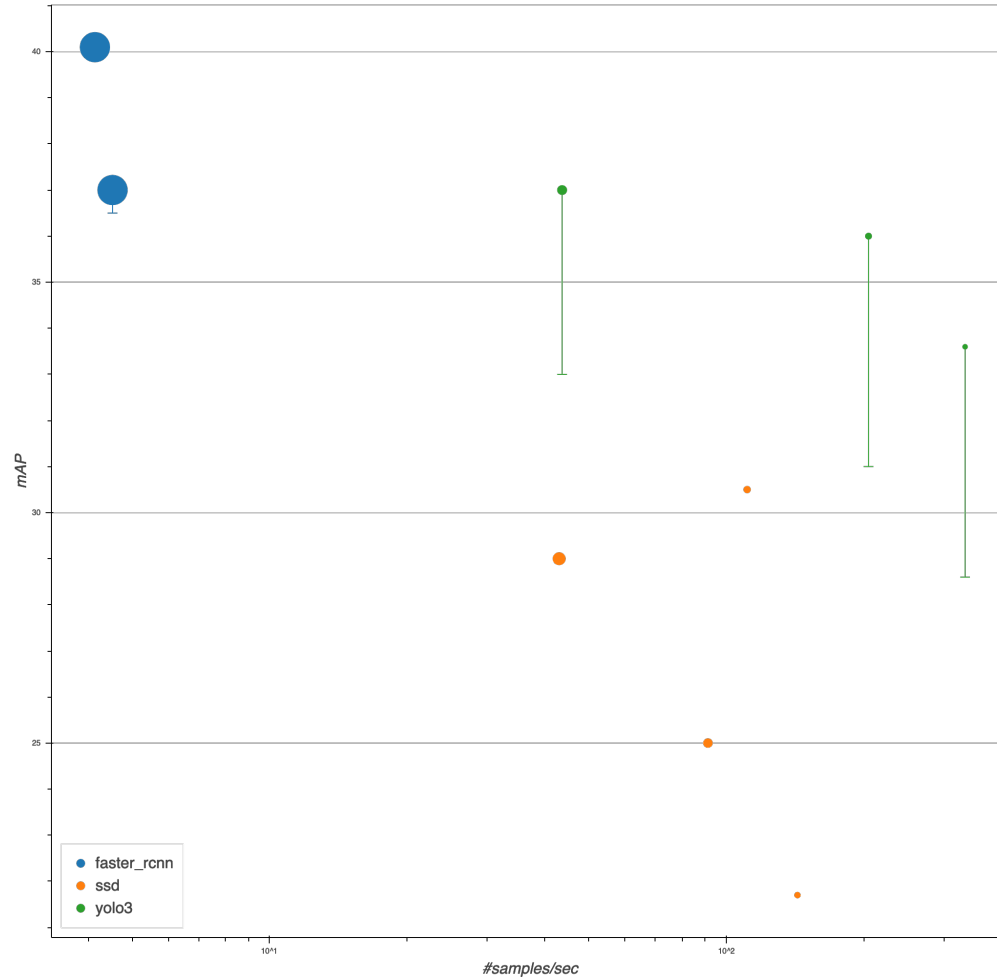


YOLO

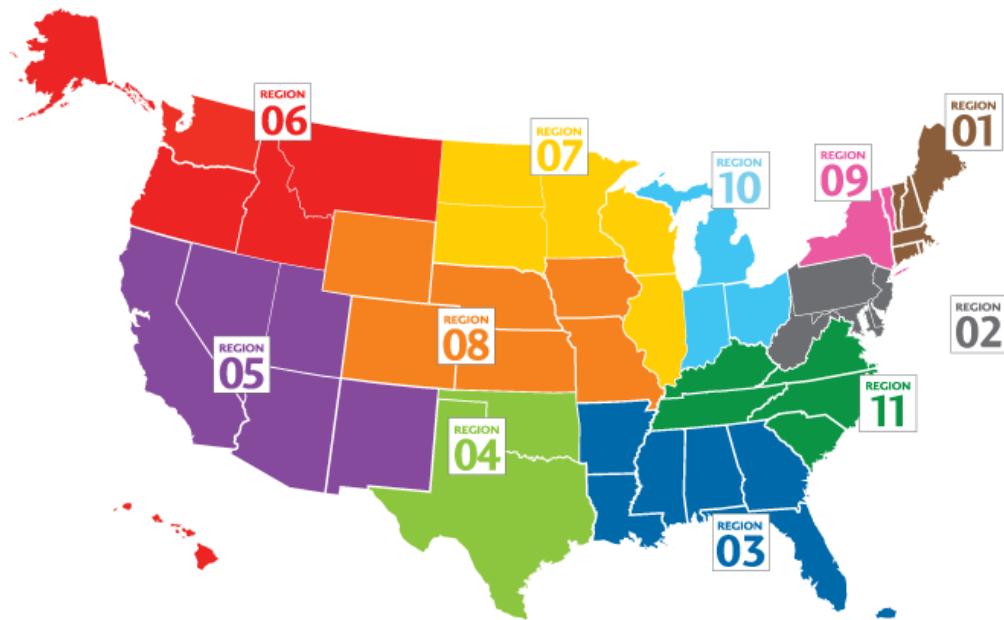


[https://gluon-cv.mxnet.io/
model_zoo/
detection.html](https://gluon-cv.mxnet.io/model_zoo/detection.html)

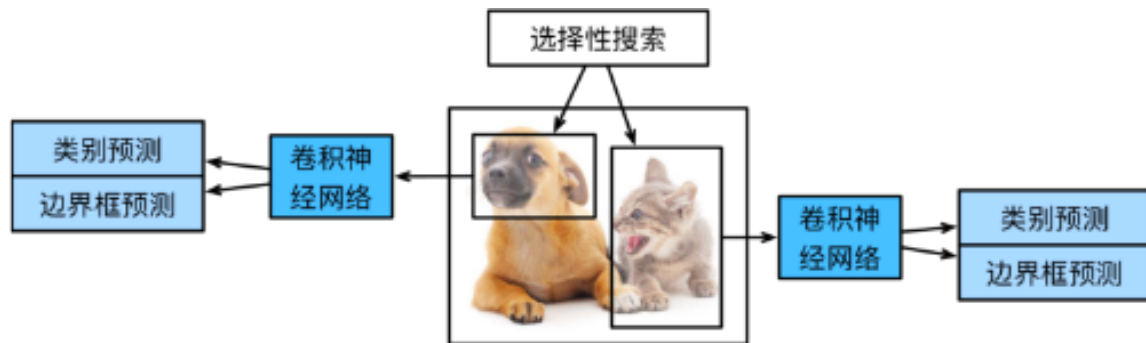
Yolo V3



区域卷积 神经网络

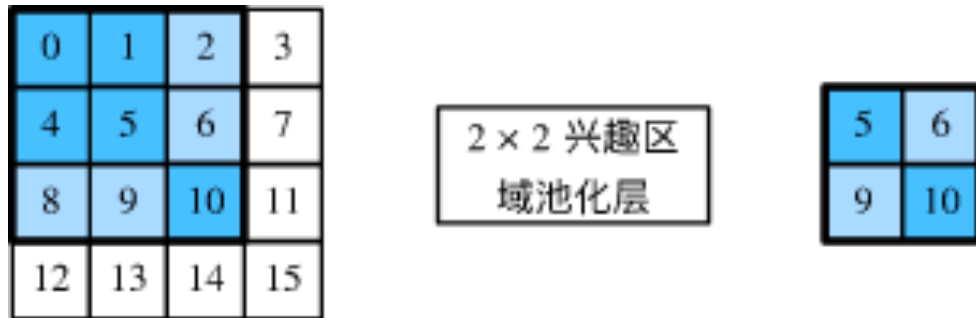


R-CNN



- 区域卷积神经网络 (region-based CNN, R-CNN)
 - 对输入图像使用选择性搜索，来选取多个高质量的区域提议
 - 选取一个预训练的卷积神经网络，并通过前向计算输出抽取的提议区域特征
 - 将每个提议区域的特征连同其标注的类别作为一个样本，训练多个支持向量机对目标分类
 - 训练线性回归模型来预测真实边界框

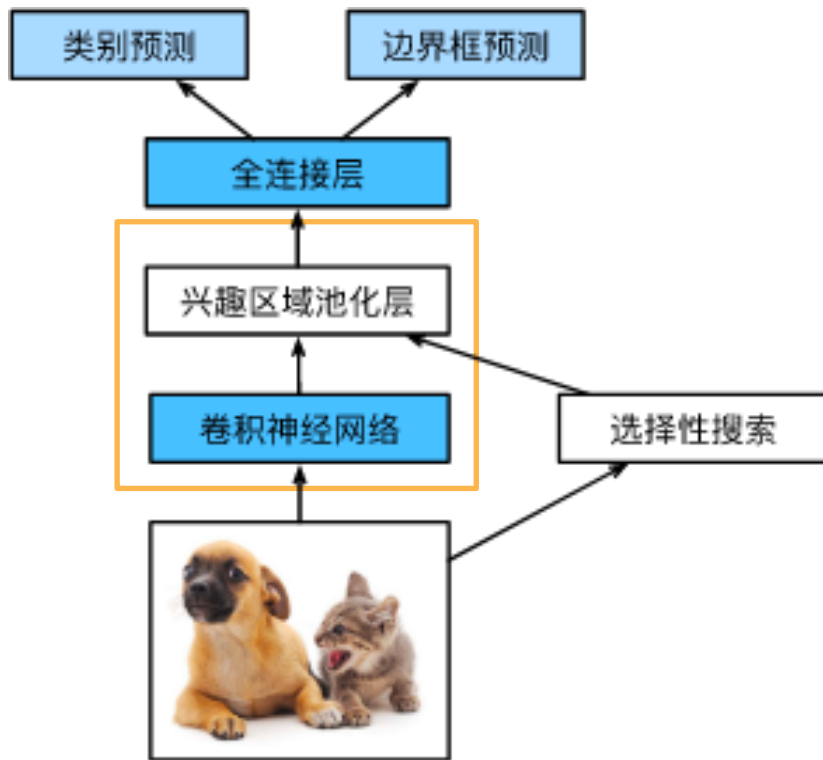
兴趣区域 (RoI) 池化层



- 兴趣区域池化层 (region of interest pooling, RoI) 池化层
 - 将一个锚框均匀地切割成 $n \times m$ 个块，输出每个块中的最大值
 - 每个锚框返回 $n \times m$ 个值

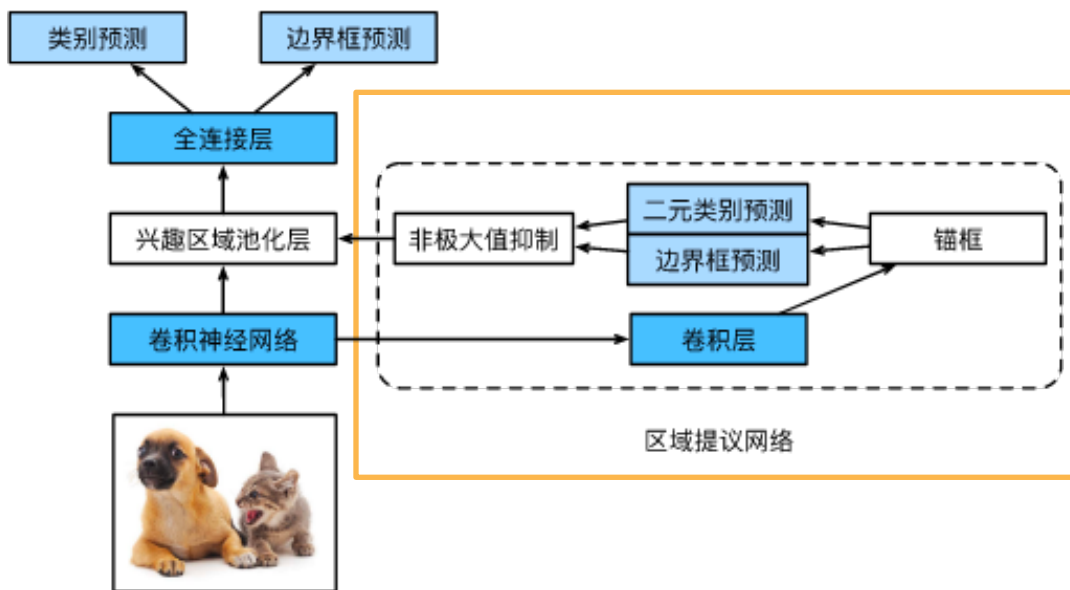
Fast RCNN

- 用CNN快速提取特征：
 - 输入是整个图像，而不是各个提议区域
 - RoI池化为每个锚框返回固定长度的特征



Faster R-CNN

- 与 Fast R-CNN 相比，只有生成提议区域的方法从选择性搜索变成了区域提议网络，而其他部分均保持不变。



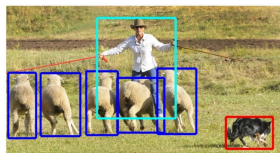
Mask R-CNN

- 如果训练数据还标注了每个目标在图像上的像素级位置，那么Mask R-CNN 能有效利用这些详尽的标注信息进一步提升目标检测的精度

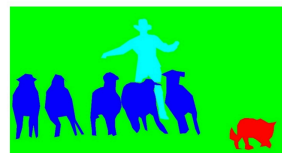
COCO



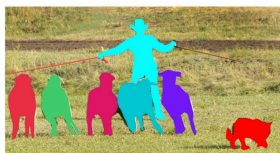
(a) Image classification



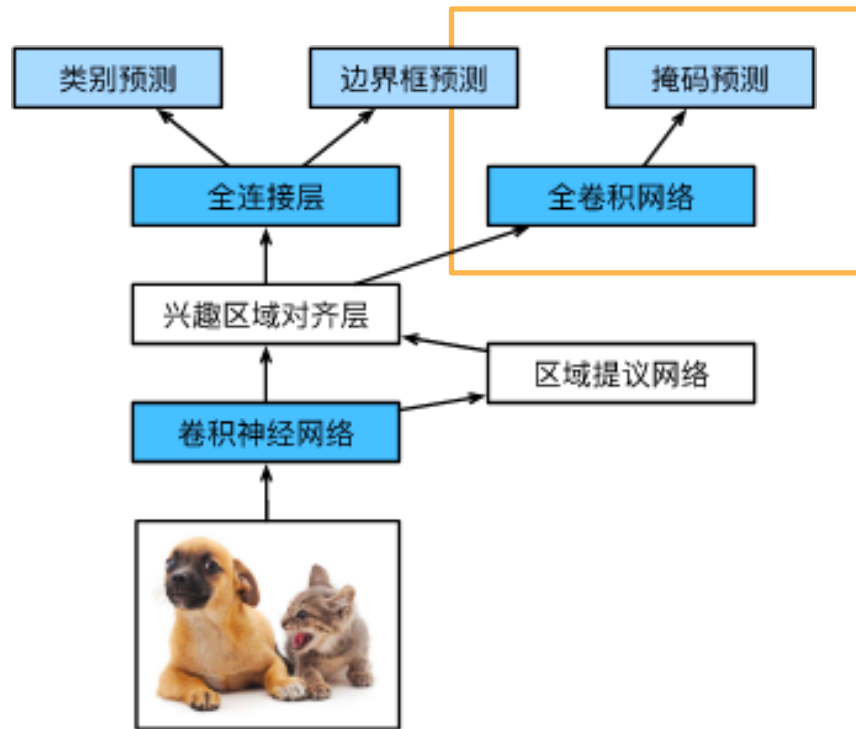
(b) Object localization

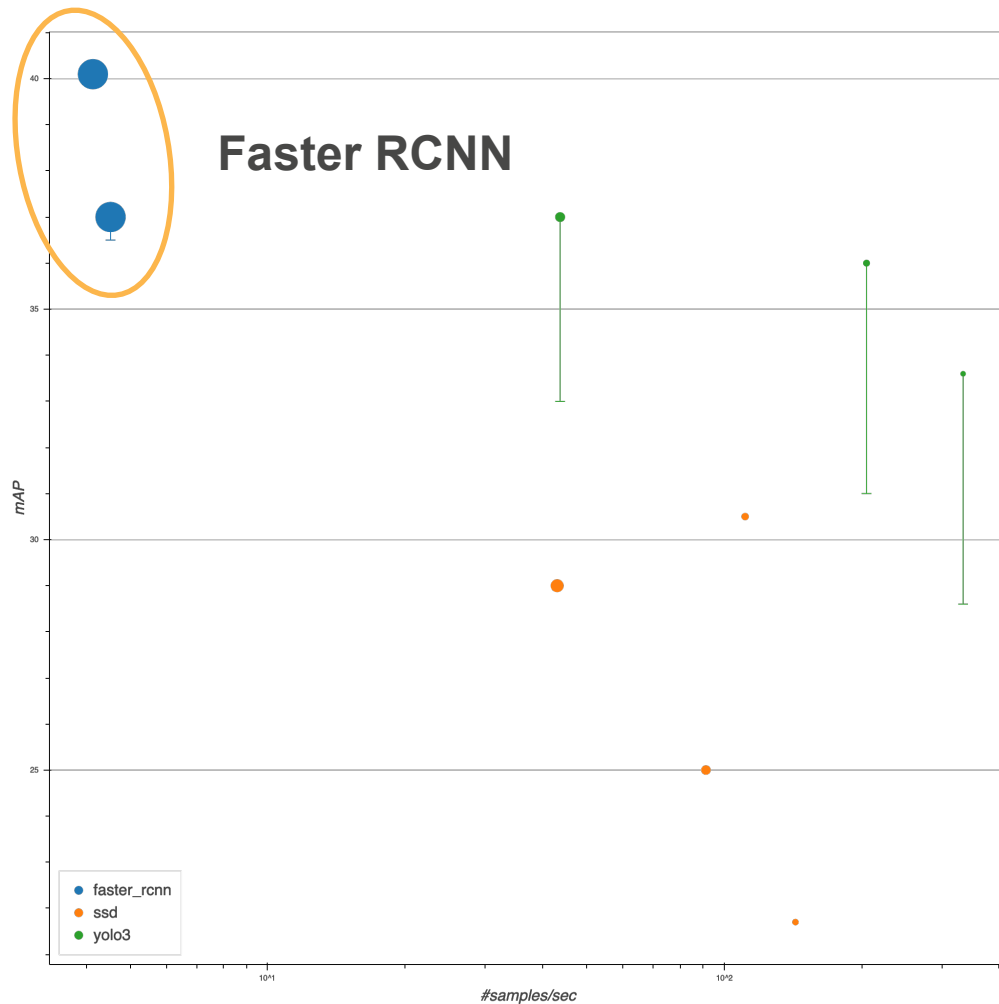


(c) Semantic segmentation



(d) This work





https://gluon-cv.mxnet.io/model_zoo/detection.html

计算机视觉 训练技巧

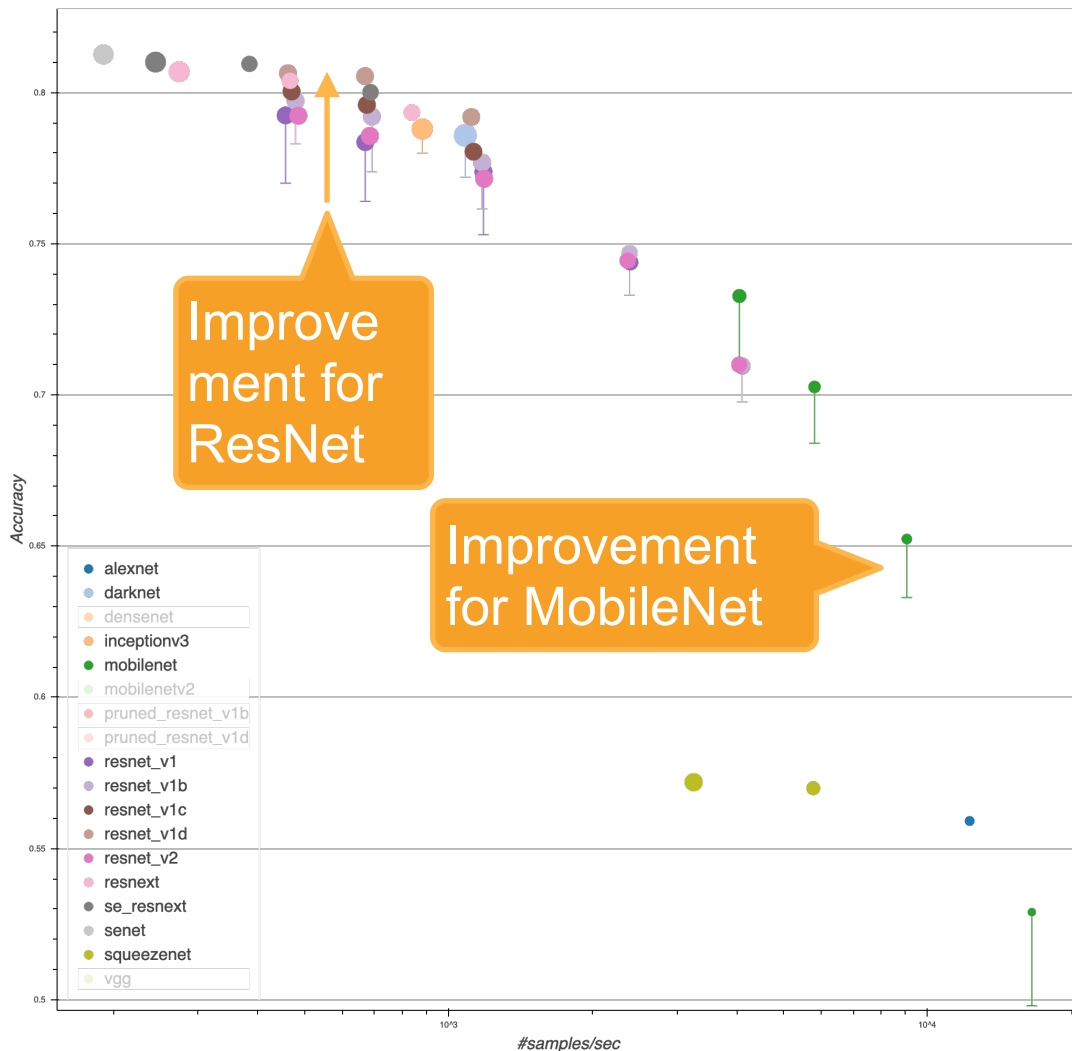


各种训练技巧可以
极大提高图像分类
模型的准确率

GluonCV model zoo

https://gluon-cv.mxnet.io/model_zoo/classification.html

<http://1day-zh.d2l.ai>

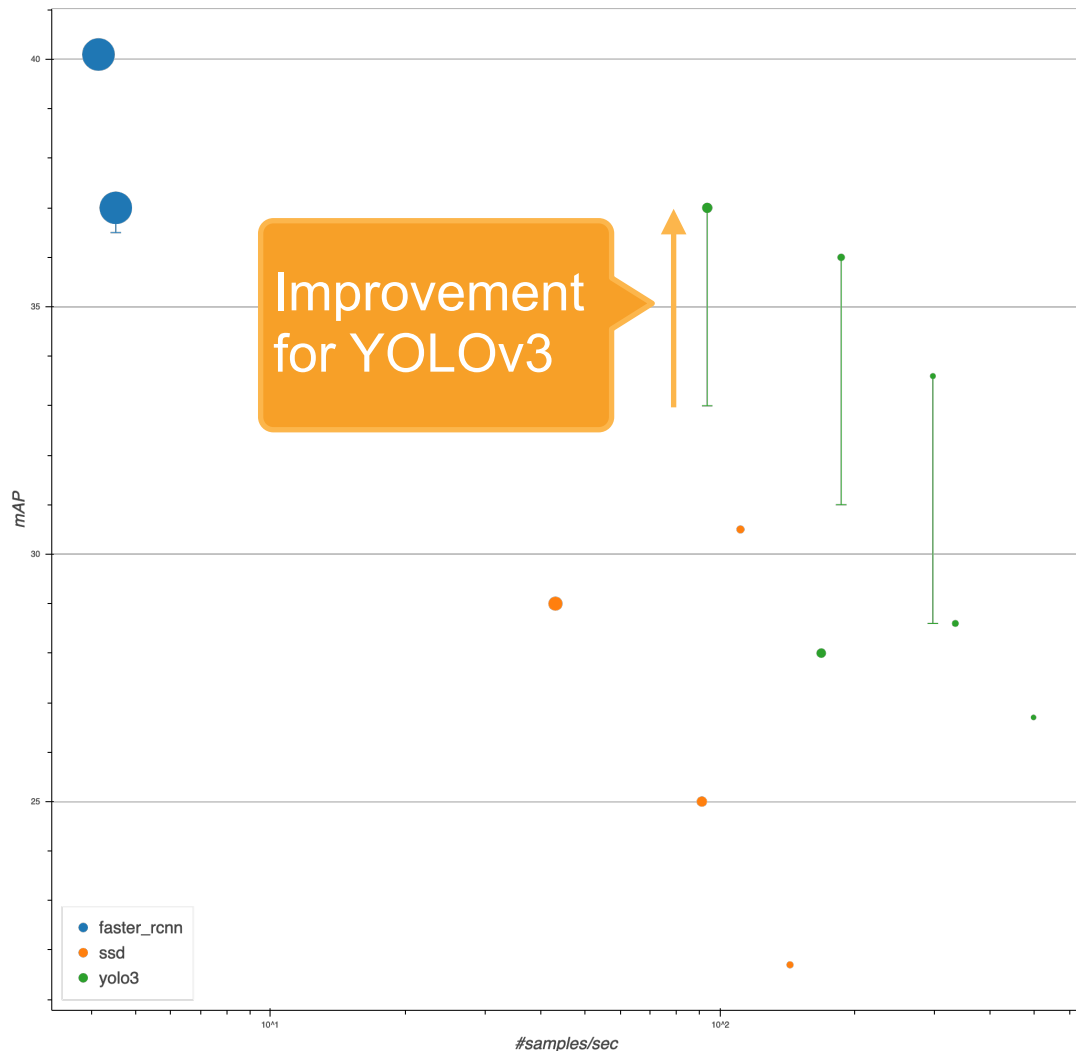


许多训练技巧也
适用于提升目标
检测模型的精度

GluonCV model zoo

https://gluon-cv.mxnet.io/model_zoo/detection.html

<http://1day-zh.d2l.ai>



混合训练数据

- 随机选两个图片 i 和 j ，抽取随机数 $\lambda \in [0,1]$
- 新的训练数据为 $x = \lambda x_i + (1 - \lambda)x_j$ $y = \lambda y_i + (1 - \lambda)y_j$
- 例如



bittern	0
...	0
otter	0
...	0
analog_clock	1

* 0.9 +



bittern	1
...	0
otter	0
...	0
analog_clock	0

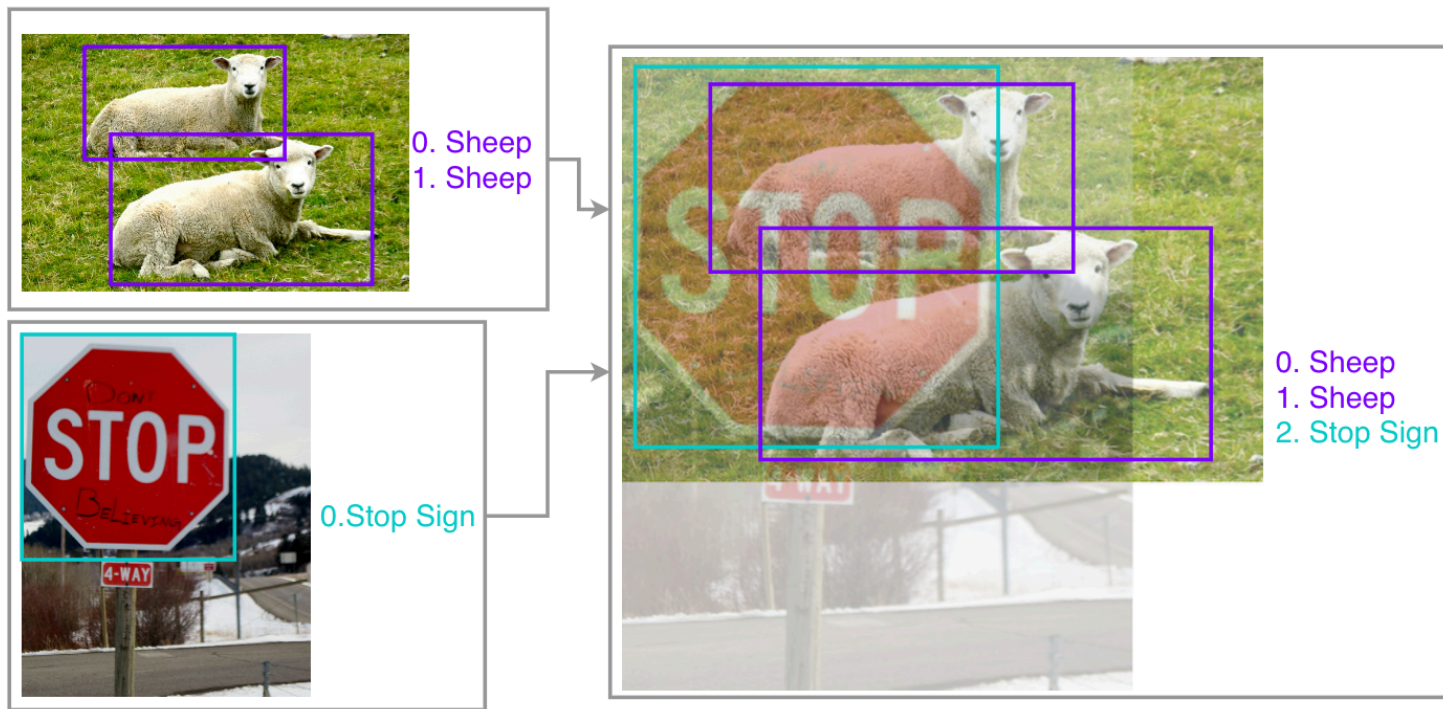
* 0.1 =



bittern	0.1
...	0
otter	0
...	0
analog_clock	0.9

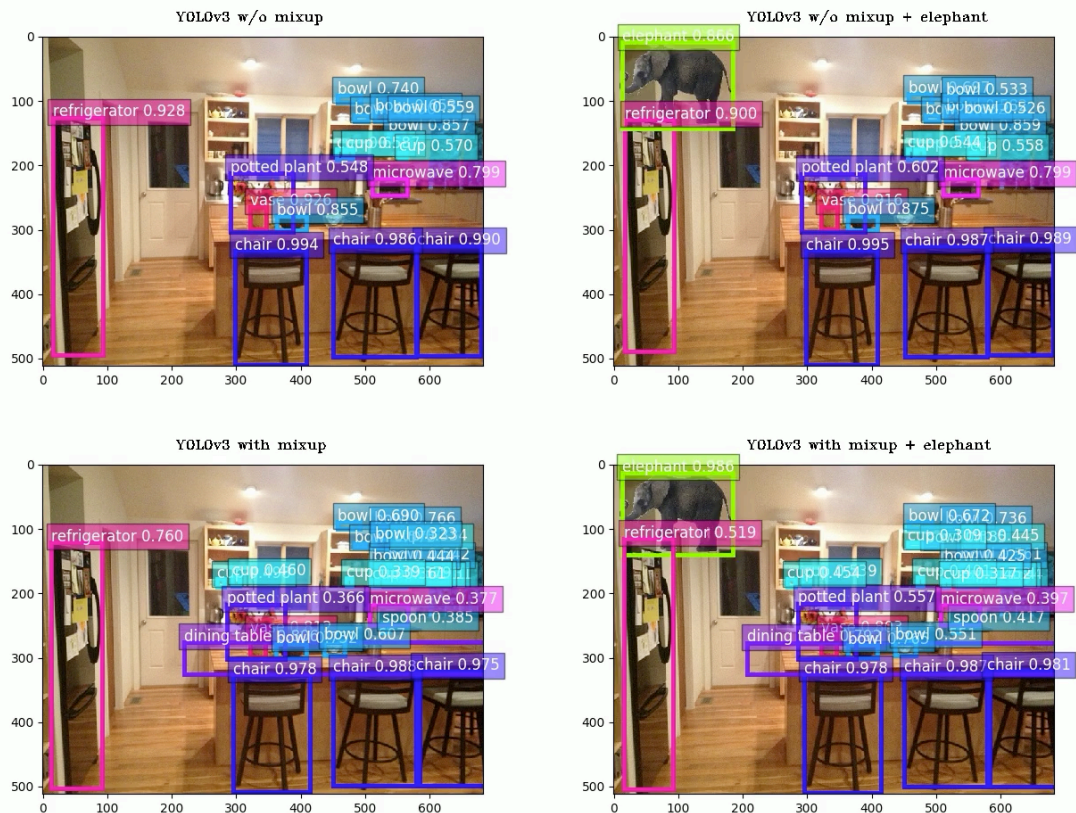
混合训练数据

- 也可以用于目标检测模型



混合训练数据

- 更稳健的模型



标签平滑化

- 假设标签 $\mathbf{y} \in \mathbb{R}^n$ 是单热编码(one-hot)

$$y_i = \begin{cases} 1 & \text{if belongs to class } i \\ 0 & \text{otherwise} \end{cases}$$

- 用softmax很难逼近要预测的 0/1值
- 平滑过后的标签

$$y_i = \begin{cases} 1 - \epsilon & \text{if belongs to class } i \\ \epsilon / (n - 1) & \text{otherwise} \end{cases}$$

- 常设置 $\epsilon = 0.1$

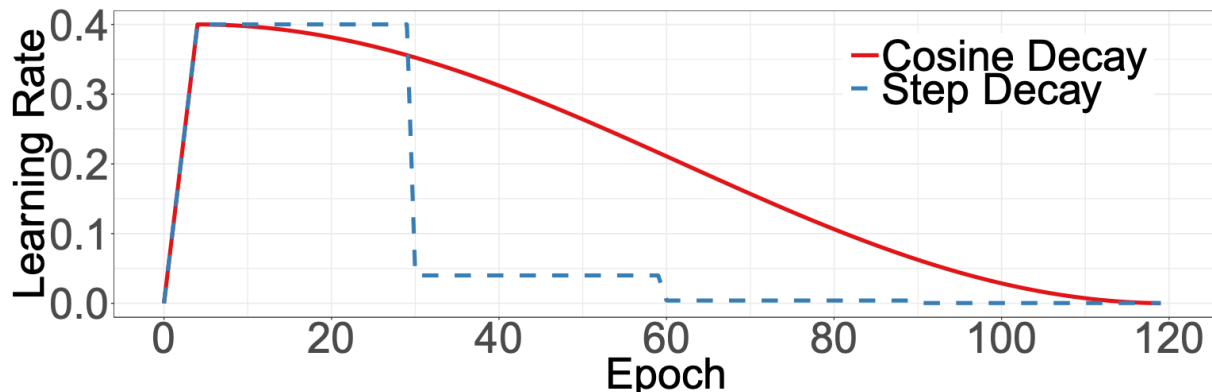
学习率预热

- 随机初始化参数应用大学习率，可能导致数值稳定性问题
- 预热技巧：在开始时使用较小的学习率，然后将其增加到原有值。
 - 如果我们选择初始学习率为0.1并使用5个周期进行预热
 - 以0开始学习速率，在前5个周期将其线性增加到0.1

余弦衰减

- 为了趋近，我们需要降低 SGD 学习率
 - 例如在第 30, 60 和 90 个迭代周期时各减少 10 倍
- 假设在总 T 次迭代（批次）中，余弦衰减计算迭代时的学习率 t 为

$$\eta_t = 1/2 \left(1 + \cos(t\pi/T) \right) \eta$$



同步批量归一化

- BatchNorm 需要大批量才能获得可靠的统计信息
- 由于 GPU 存储器限制，目标对象检测任务用小批量。
 - 例如，每 GPU 1个图像
- 在多 GPU 培训中，每个 GPU 分别计算均值/方差
- 同步批量归一化根据所有 GPU 的统计信息计算全体均值与方差

图像分类结果

Refinements	ResNet-50-D		Inception-V3		MobileNet	
	Top-1	Δ	Top-1	Δ	Top-1	Δ
Efficient	77.16		77.50		71.90	
+ cosine decay	77.91	+0.75	78.19	+0.69	72.83	+0.93
+ label smoothing	78.31	+0.4	78.40	+0.21	72.93	+0.1
+ mixup	79.15	+0.84	78.77	+0.37	73.28	+0.35

He et.al *Bag of Tricks for Image Classification with Convolutional Neural Networks*

YOLO v3 结果

Incremental Tricks	mAP	Δ	Cumu Δ
- data augmentation	64.26	-15.99	-15.99
baseline	80.25	0	0
+ synchronize BN	80.81	+0.56	+0.56
+ random training shapes	81.23	+0.42	+0.98
+ cosine lr schedule	81.69	+0.46	+1.44
+ class label smoothing	82.14	+0.45	+1.89
+ mixup	83.68	+1.54	+3.43

Zhi et al, *Bag of Freebies for Training Object Detection Neural Networks*

<http://1day-zh.d2l.ai>



总结

- 图像增广
- 微调
- 目标检测
- 边界框和锚框
- 单发多框检验 (SSD) 模型
- 只看一次 (YOLO)
- 区域卷积神经网络 (R-CNN)
- 计算机视觉训练技巧