Midterm Project - Report

张千一 18302010032

2022 年 5 月 3 日

目录

1 在 CIFAR-100 数据集上训练 CNN

1.1 数据集介绍与划分

CIFAR-100 数据集共有 60000 张带标签图像，这些图像被分为 100 个类，且类之间完全互斥。其中每个类有 600 张大小为 $32 \times 32 \times 3$ 的 RGB 彩色图像，500 张作为训练集，100 张作为测试集。对于每一张图像，它有 fine_labels（细粒度）和 coarse_labels（粗粒度）两个标签，对应图 1 中的 Classes 和 Superclass：

Superclass	**Classes**
aquatic mammals	beaver, dolphin, otter, seal, whale
fish	aquarium fish, flatfish, ray, shark, trout
flowers	orchids, poppies, roses, sunflowers, tulips
food containers	bottles, bowls, cans, cups, plates
fruit and vegetables	apples, mushrooms, oranges, pears, sweet peppers
household electrical devices	clock, computer keyboard, lamp, telephone, television
household furniture	bed, chair, couch, table, wardrobe
insects	bee, beetle, butterfly, caterpillar, cockroach
large carnivores	bear, leopard, lion, tiger, wolf
large man-made outdoor things	bridge, castle, house, road, skyscraper
large natural outdoor scenes	cloud, forest, mountain, plain, sea
large omnivores and herbivores	camel, cattle, chimpanzee, elephant, kangaroo
medium-sized mammals	fox, porcupine, possum, raccoon, skunk
non-insect invertebrates	crab, lobster, snail, spider, worm
people	baby, boy, girl, man, woman
reptiles	crocodile, dinosaur, lizard, snake, turtle
small mammals	hamster, mouse, rabbit, shrew, squirrel
trees	maple, oak, palm, pine, willow
vehicles 1	bicycle, bus, motorcycle, pickup truck, train
vehicles 2	lawn-mower, rocket, streetcar, tank, tractor

图 1: CIFAR-100 数据集的标签

下面展示了 CIFAR-100 训练集中的 9 张图像：

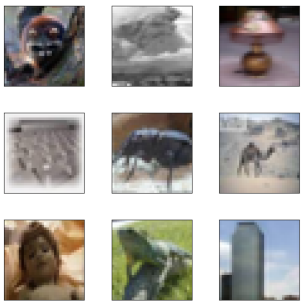


图 2: CIFAR-100 数据集的图像示例

1.2 baseline 模型

1.2.1 模型基本结构

残差网络 (ResNet) 由微软研究院的何恺明、张祥雨、任少卿、孙剑等人在 Deep Residual Learning for Image Recognition 一文中提出，在 2015 年的 ILSVRC（ImageNet Large Scale Visual Recognition Challenge）中取得了冠军。ResNet 认为，理论上，可以训练一个相对浅层的网络，然后在这个训练好的相对浅层的网络上堆几层恒等映射层，即输出等于输入的层，构建出一个更深层的网络。这两个网络得到的结果应该是一模一样的，因而在训练集上，深层的网络不会比浅层的网络效果差。

ResNet 通过加入 shortcut connections，变得更加容易被优化。包含一个 shortcut connection 的几层网络被称为一个残差块（residual block），如图 3 所示。

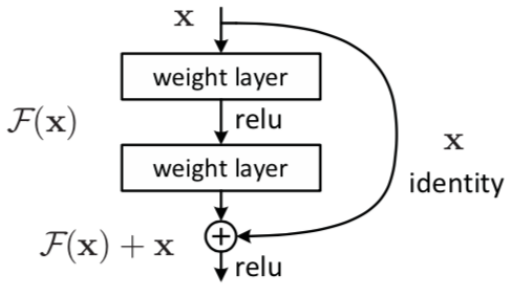


图 3: 残差块

如图 4 所示，展示了 ResNet 的几种结构：

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	[3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Table 1. Architectures for ImageNet. Building blocks are shown in brackets (see also Fig. 5), with the numbers of blocks stacked. Down-sampling is performed by conv3_1, conv4_1, and conv5_1 with a stride of 2.

图 4: ResNet 的几种结构

其中，ResNet-18 和 ResNet-34 使用两个 3×3 的卷积层作为一个块，而 ResNet-50，ResNet-101 和 ResNet-152 将其替换为 $1\times1+3\times3+1\times1$ 的卷积进行计算优化，这样的块即减少了计算量又能够保持精度，被称为 BottleNeck 块。

另外，为了使得输入和输出保持相同的维度，若特征图维度不同，对于卷积层的残差块，需要在 1×1 卷积后添加批归一化（Batch Normalization）处理，如图 5 所示。

考虑到训练的效率，我们使用 ResNet-18 作为 baseline 模型。由于 CIFAR-100 的图像较小，ResNet-18 中的第一层卷积使用 7×7 卷积核可能过大，难以提取 CIFAR-100 图像的特征，我们将其调整为 3×3 卷积核。此外，模型的其它的 baseline 如下：

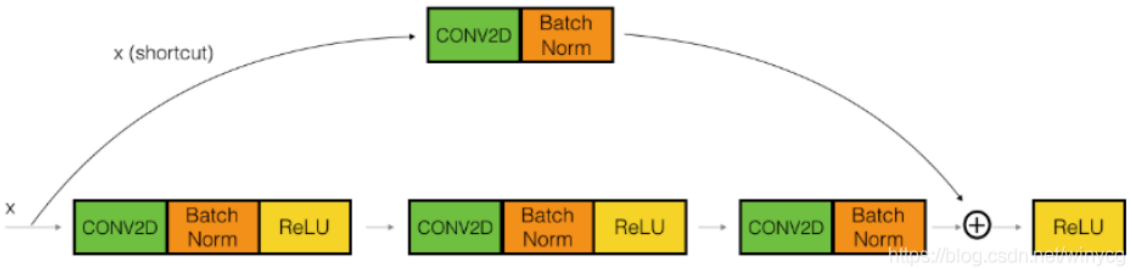
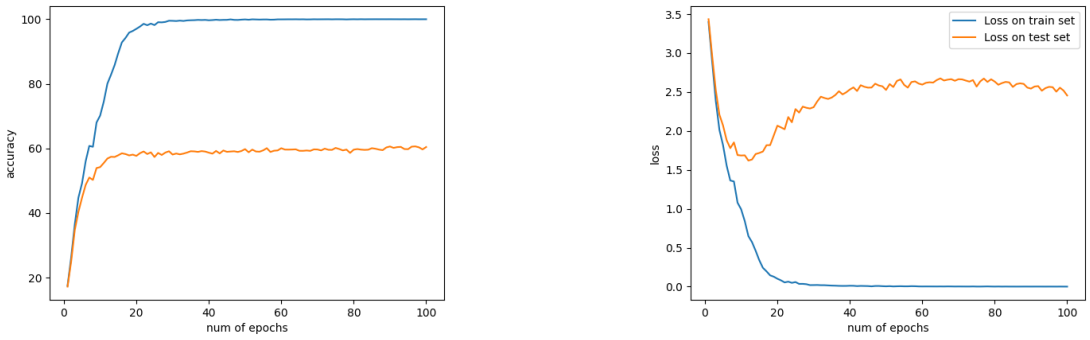


图 5: 批归一化

- epoch：100
- batch size：100
- drop out：不使用
- 损失函数：交叉熵损失
- 正则化：l_2 正则化，正则化参数 $\lambda = 10^{-5}$
- 优化器：Adam
- 动量系数：$\beta_1 = 0.9, \beta_2 = 0.999$
- 初始学习率：$\alpha = 10^{-3}$
- 学习率衰减：每个 epoch 后学习率衰减为上一个 epoch 的 0.98

我们得到 baseline 模型在测试集上的平均误差为 **2.455**，分类精度为 **0.604**。模型的性能曲线如下所示：



(a) accuracy 曲线

(b) loss 曲线

图 6: baseline 模型的性能曲线

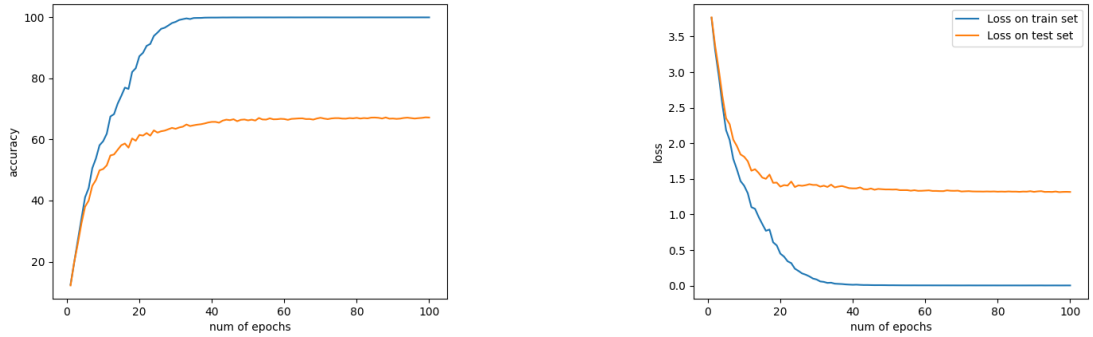
可以发现，模型的精度较低，且模型存在较严重的振荡和过拟合现象。

1.2.2 模型超参数调整

为了缓解模型的过拟合现象，我们将 baseline 模型的超参数进行调整，以得到较高的精度。调整后模型的超参数如下：

- epoch：100
- batch size：100
- drop out：使用
- 损失函数：交叉熵损失
- 正则化：l_2 正则化，正则化参数 $\lambda = 10^{-3}$
- 优化器：SGD
- 动量系数：$\beta = 0.9$
- 初始学习率：$\alpha = 10^{-2}$
- 学习率衰减：每个 epoch 后学习率衰减为上一个 epoch 的 0.98

经过调整后，我们得到模型在测试集上的平均误差为 **1.317**，分类精度为 **0.672**。模型的性能曲线如下所示：



(a) accuracy 曲线　　(b) loss 曲线

图 7: 调整后的性能曲线

可以发现，模型的过拟合现象得到了一定程度的改善，精度得到了提高，且不再出现测试集上 loss 随训练过程增加的现象。

1.3 数据增强

由于数据增强后，模型较难学习到图像的特征，因此我们调整训练的 epoch 数至 200 个，且调整学习率衰减系数至 0.98。

1.3.1 随机裁剪和水平翻转

我们可以对模型的数据作一些简单的数据增强。基于 torchvision 自带的 transforms.RandomCrop 和 transforms.RandomHorizontalFlip 函数，我们可以简单地实现对训练集图像进行向四周填充后随机裁剪以及随机翻转的操作。这两种操作不会对图像的基本信息造成太大影响。

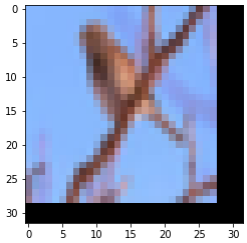
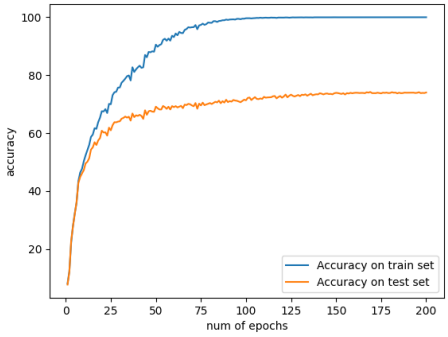
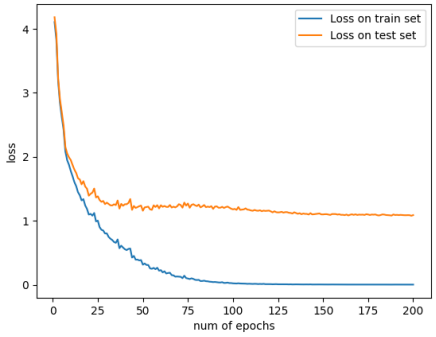


图 8: 向四周填充后随机裁剪的效果

经过随机裁剪和水平翻转后，我们得到模型在测试集上的平均误差为 **1.090**，分类精度为 **0.740**。模型的性能曲线如下所示：



(a) accuracy 曲线



(b) loss 曲线

图 9: 增强后模型的性能曲线

1.3.2 Cut out

Cut out 的原理是：随机将样本中的部分区域去掉，然后填充 0 像素值，分类的结果不变。下面展示了三张训练集样本 Cut out 后的图像：

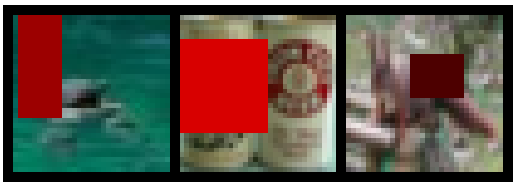
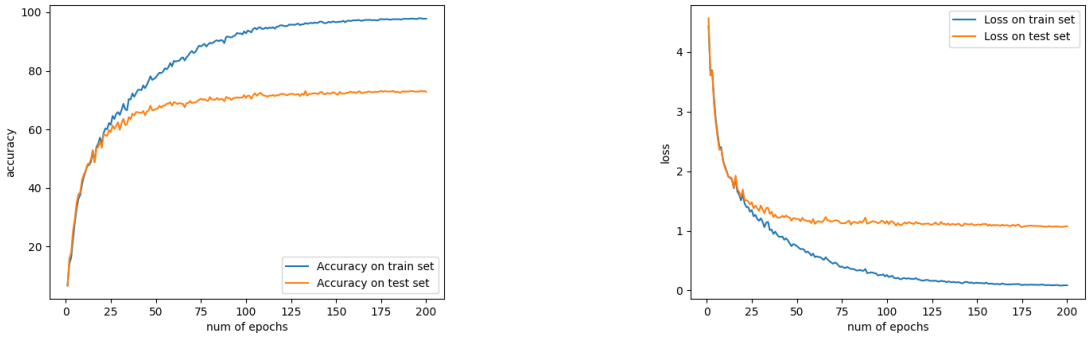


图 10: cutout 的效果

基于 torchtoolbox 包自带的 Cutout 方法，我们在数据增强时增加 Cut out 操作，设置对图像进行

Cutout 的概率为 0.5。进行 Cut out 后，得到模型在测试集上的平均误差为 **1.076**，分类精度为 **0.729**。模型的性能曲线如图 11 所示：



(a) accuracy 曲线　　(b) loss 曲线

图 11: cutout 后模型的性能曲线

我们可以可视化五个中间 block 各一个特征通道的输出，如图 12 所示：

(a) 输入层　(b) Conv1 层输出　(c) Conv2 层输出

(d) Conv3 层输出　(e) Conv4 层输出　(f) Conv5 层输出

图 12: 隐藏层输出可视化

1.3.3 Cutmix

Cutmix 的原理是：将一部分区域去掉，但不是填充随机的固定像素值，而是随机填充训练集中的其他数据的区域像素值，分类结果按一定的比例分配。

Cutmix 可以用公式表示：

$$\tilde{x} = \mathbf{M} \odot x_A + (\mathbf{1} - \mathbf{M}) \odot x_B$$

$$\tilde{y} = \lambda y_A + (1 - \lambda) y_B$$

其中 $\lambda \sim Beta(\alpha, \alpha)$，$\mathbf{M} \in \{0,1\}^{W \times H}$ 表示选择来自哪张图片的掩码，$\odot$ 表示逐元素相乘。为此，我们还需要确定一个随机采样得到的 bounding box，即 $B = (r_x, r_y, r_w, r_h)$，采样方法为：

$$r_x \sim \text{Unif}(0, W), \qquad r_w = W\sqrt{1-\lambda}$$

$$r_y \sim \text{Unif}(0, H), \qquad r_h = H\sqrt{1-\lambda}$$

这样即可保证裁剪区域的大小为原图像的 $1 - \lambda$ 倍。

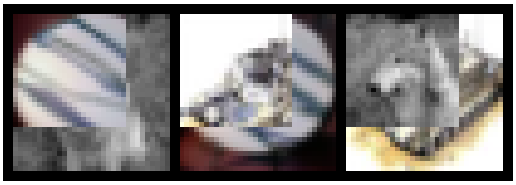
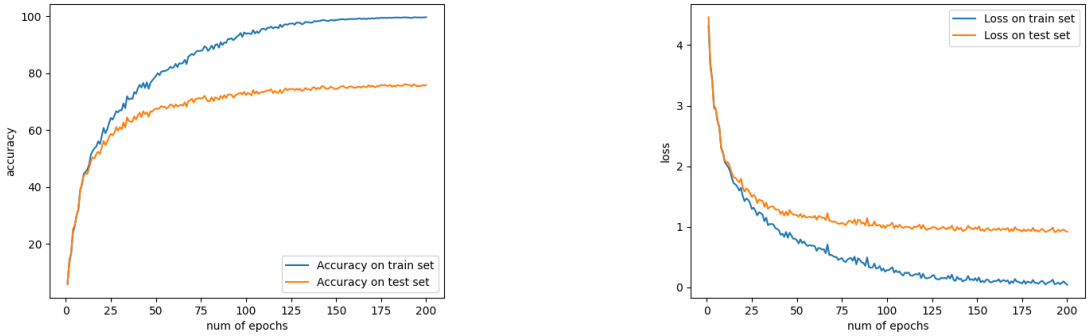


图 13: cutmix 的效果

我们取 $\alpha = 1$，此时 $\lambda \sim \text{Unif}(0, 1)$。图 13 展示了以三张训练集样本作为一个 batch，经过 cutmix 后的图像：

进行 cutmix 后，我们得到模型在测试集上的平均误差为 **0.916**，分类精度为 **0.759**。模型的性能曲线如图 14 所示：

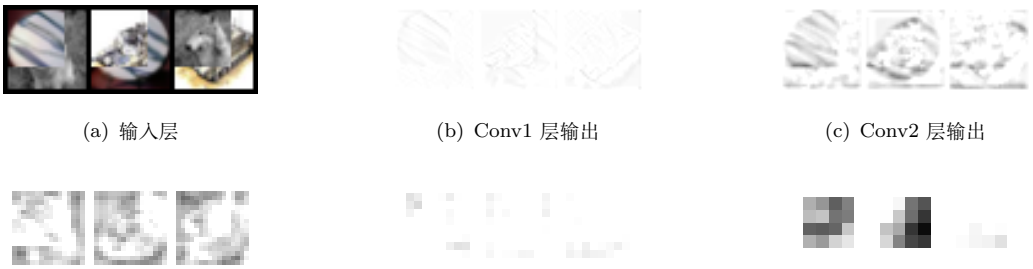


(a) accuracy 曲线

(b) loss 曲线

图 14: cutmix 后模型的性能曲线

可视化五个中间 block 各一个特征通道的输出，如图 15 所示。



(a) 输入层

(b) Conv1 层输出

(c) Conv2 层输出

(d) Conv3 层输出

(e) Conv4 层输出

(f) Conv5 层输出

图 15: 隐藏层输出可视化

1.3.4 Mix up

Mixup 的原理是：将两张图按比例进行插值来混合样本。

Mixup 可以用公式表示：

$$\tilde{x} = \lambda x_A + (1-\lambda)x_B$$

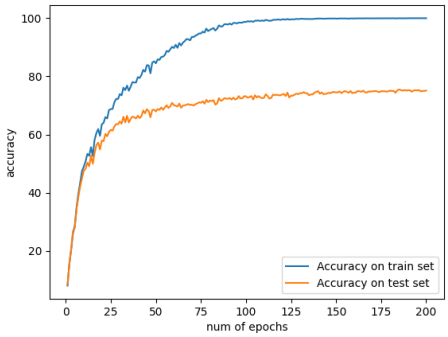
$$\tilde{y} = \lambda y_A + (1-\lambda)y_B$$

其中 $\lambda \sim Beta(\alpha, \alpha)$

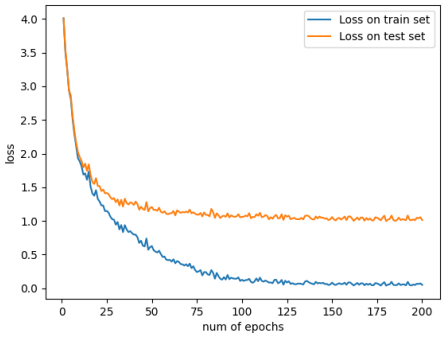
我们取 $\alpha = 0.2$，如图 16 所示，展示了以三张训练集样本作为一个 batch，经过 Mix up 后的图像。

图 16: mixup 的效果

进行 mixup 后，我们得到模型在测试集上的平均误差为 **1.012**，分类精度为 **0.751**。模型的性能曲线如图 17 所示：



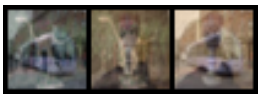
(a) accuracy 曲线



(b) loss 曲线

图 17: mixup 后模型的性能曲线

可视化五个中间 block 各一个特征通道的输出，如图 18 所示。



(a) 输入层

(b) Conv1 层输出

(c) Conv2 层输出

(d) Conv3 层输出

(e) Conv4 层输出

(f) Conv5 层输出

图 18: 隐藏层输出可视化

1.4 小结

我们总共得出下面四个模型：

- baseline 模型（ResNet-18，经过超参数调整）
- baseline 模型 + cutout
- baseline 模型 + cutmix
- baseline 模型 + mixup

四个模型在测试集上的 loss 和 accurary 如下表所示：

模型	测试集误差	测试集精度
baseline	1.040	74.0%
baseline + cutout	1.076	72.9%
baseline + cutmix	0.916	75.9%
baseline + mixup	1.012	75.1%

表 1: 不同数据增强方式对模型性能的影响

我们发现，在 baseline 模型上，增加 cutmix 方法对数据进行增强，相对于 cutout 和 mixup 两种方法达到的测试集上的误差最小，精度最高。可以总结出 cutmix 的以下几个优点：

- 在训练过程中不会出现非信息像素，从而能够提高训练效率；
- 保留了 cutout 方法 regional dropout 的优势，能够关注目标的 non-discriminative parts；
- 通过要求模型从局部视图识别对象，对 cut 区域中添加其他样本的信息，能够进一步增强模型的定位能力；
- 相较于 mixup 方法，不会有图像混合后不自然的情形，能够提升模型分类的表现；
- 训练和推理的代价保持不变。

2 在 VOC 数据集上训练 Faster R-CNN 和 YOLO V3

2.1 数据集介绍与划分

2.1.1 数据集总体概况

PASCAL 对于目标检测或分割类型来说属于先驱者的地位。对于现在的研究者来说比较重要的两个年份的数据集是 **PASCAL VOC 2007** 与 **PASCAL VOC 2012**，这两个数据集频频在现在的一些检测或分割类的论文当中出现。

由于从 2007 年之后，PASCAL VOC 的测试集均不再公布，因此我们选择 PASCAL VOC 2007 作为我们的数据集。下面展示了 PASCAL VOC 2007 数据集的 20 个类别及其层级结构：

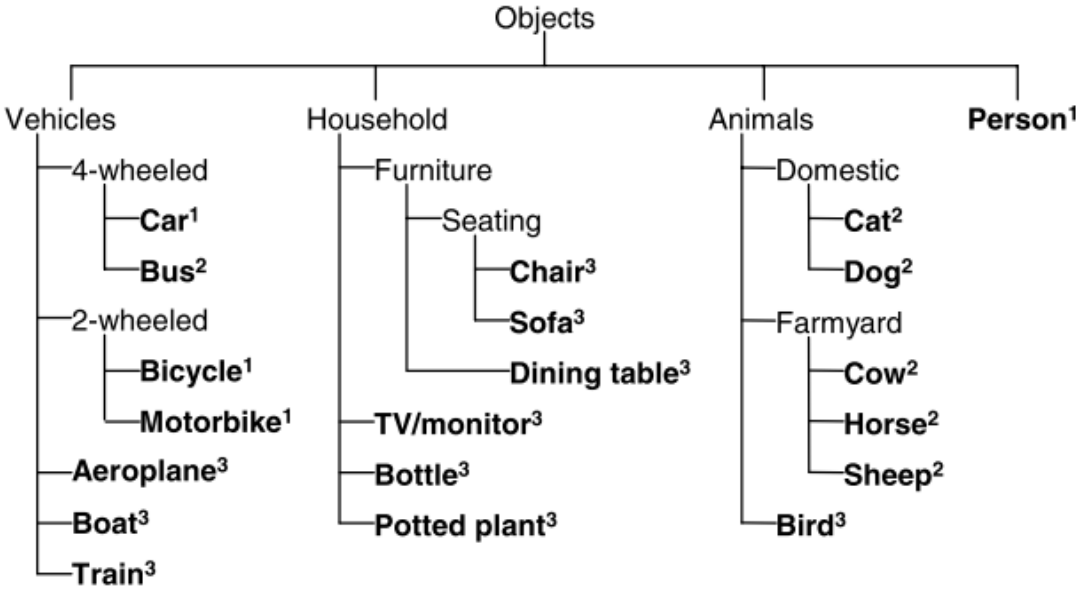


图 19: VOC 2007 数据集的标签

其中：

- 分为 4 个大类，20 个小类。预测时主要输出的小类。
- 数据集主要关注分类和检测，也就是分类和检测用到的数据集相对规模较大。关于其他任务比如分割，动作识别等，其数据集一般是分类和检测数据集的子集。

如图 20 所示，展示了 PASCAL VOC 2007 数据集总体的统计情况。

2.1.2 数据集划分

PASCAL VOC 2007 数据集分为两部分：训练和验证集 trainval，测试集 test，两部分各占数据总量的约 50%。其中 trainval 又分为训练集和测试集，二者分别各占 trainval 的 50%。每张图片中有可能包含不只一个目标 object。

如图 21 所示，展示了数据集在训练集，验证集，测试集上的划分情况。

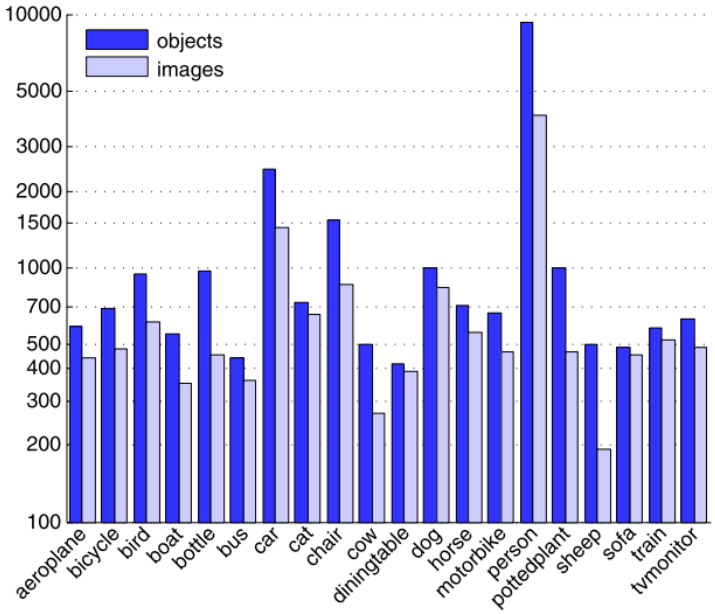


图 20: VOC 2007 数据集的统计情况

Table 2 Statistics of the VOC2007 dataset. The data is divided into two main subsets: training/validation data (`trainval`), and test data (`test`), with the `trainval` data further divided into suggested training (`train`) and validation (`val`) sets. For each subset and class, the number of images (containing at least one object of the corresponding class) and number of object instances are shown. Note that because images may contain objects of several classes, the totals shown in the image columns are not simply the sum of the corresponding column

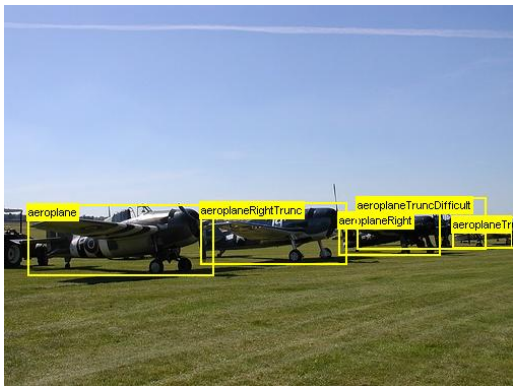
	`train`		`val`		`trainval`		`test`	
	images	objects	images	objects	images	objects	images	objects
Aeroplane	112	151	126	155	238	306	204	285
Bicycle	116	176	127	177	243	353	239	337
Bird	180	243	150	243	330	486	282	459
Boat	81	140	100	150	181	290	172	263
Bottle	139	253	105	252	244	505	212	469
Bus	97	115	89	114	186	229	174	213
Car	376	625	337	625	713	1,250	721	1,201
Cat	163	186	174	190	337	376	322	358
Chair	224	400	221	398	445	798	417	756
Cow	69	136	72	123	141	259	127	244
Dining table	97	103	103	112	200	215	190	206
Dog	203	253	218	257	421	510	418	489
Horse	139	182	148	180	287	362	274	348
Motorbike	120	167	125	172	245	339	222	325
Person	1,025	2,358	983	2,332	2,008	4,690	2,007	4,528
Potted plant	133	248	112	266	245	514	224	480
Sheep	48	130	48	127	96	257	97	242
Sofa	111	124	118	124	229	248	223	239
Train	127	145	134	152	261	297	259	282
Tv/monitor	128	166	128	158	256	324	229	308
Total	2,501	6,301	2,510	6,307	5,011	12,608	4,952	12,032

图 21: VOC 2007 数据集的具体划分情况

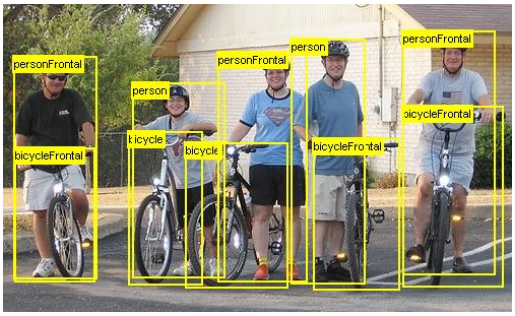
2.1.3 数据集可视化

http://host.robots.ox.ac.uk/pascal/VOC/voc2007/examples/index.html

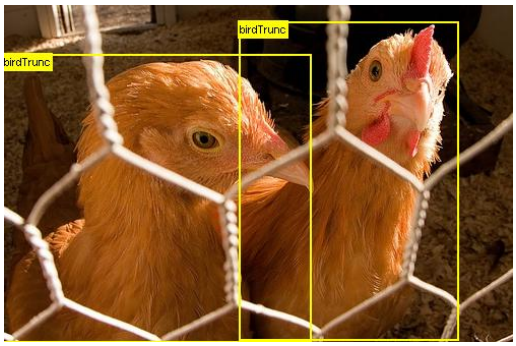
网站中展示了 VOC-2007 中至少包含每个类一个目标的 8 张图像及其目标的 ground truth 位置。下面展示一部分：



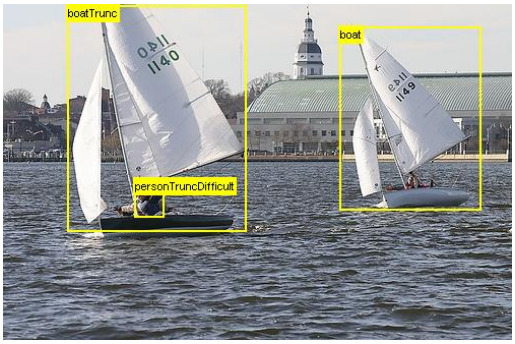
(a) Aeroplanes



(b) Bicycles

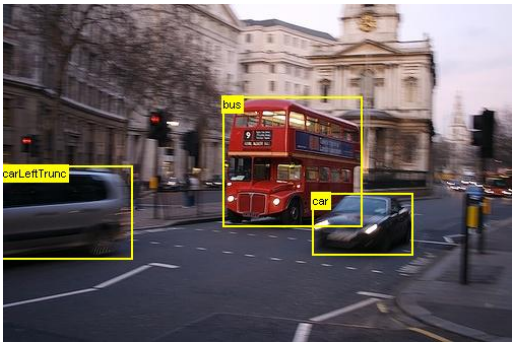


(c) Birds



(d) Boats

(e) Bottles



(f) Buses

图 22: VOC 2007 数据集示例

2.1.4 读取 VOC 数据集

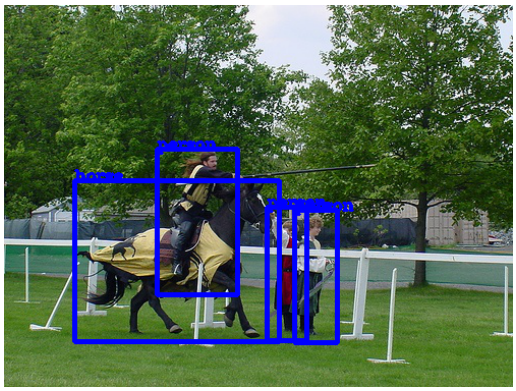
使用 torchvision.datasets 包中的 VOCDetection 下载数据集，在 VOC2007 文件夹中，得到下面几个部分：

- Annotations：包含了记录图像和标签信息的 xml 文件
- ImageSets：主要包含划分的训练集和测试集中图片的名称
- JPEGImages：包含数据集的原始图片
- SegmentationClass：按类别分割的 ground truth 位置
- SegmentationObject：按对象分割的 ground truth 位置

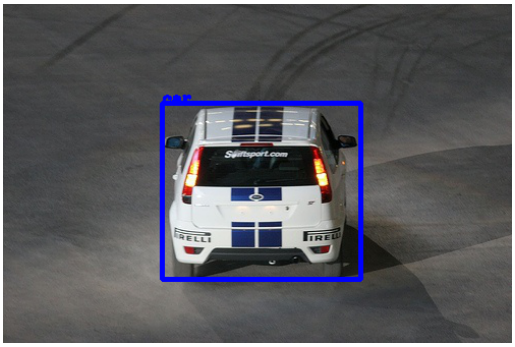
我们读取了四张训练集的图像并对其目标的 ground truth 进行了标注：

(a) 图片 1

(b) 图片 2



(c) 图片 3



(d) 图片 4

图 23: VOC 2007 训练集图像 ground truth 标注示例

2.2 Faster R-CNN 模型

2.2.1 基本结构

经过 R-CNN 和 Fast R-CNN 的积淀，Ross B. Girshick 在 2016 年提出了新的 Faster R-CNN，在结构上，Faster R-CNN 已经将特征抽取 (feature extraction)，proposal 提取，bounding box regression(rect refine)，classification 都整合在了一个网络中，使得综合性能有较大提高，在检测速度方面尤为明显。

整个 Faster R-CNN 分为 4 大部分：共享卷积网络，候选检测框生成网络 RPN（Region Proposal Networks），感兴趣区域池化 RoI（Region of Interest）Pooling 和分类器。如图 24 所示。

其中：

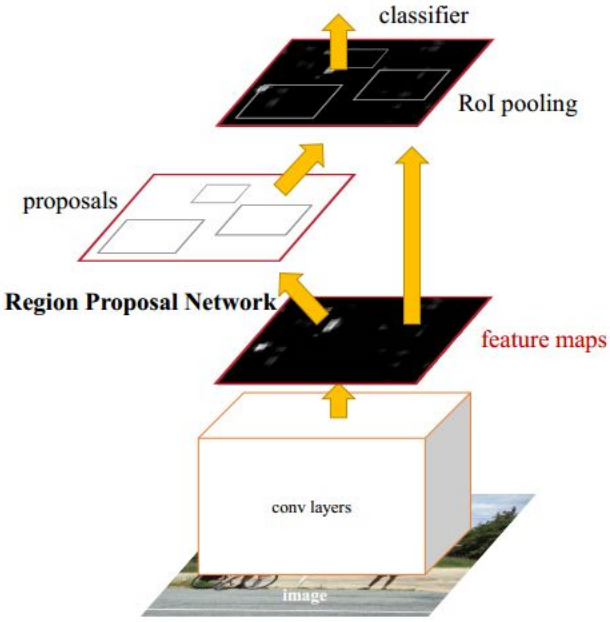


图 24: Faster R-CNN 基本结构

- 共享卷积网络（Conv layers）：提取 image 的 feature maps。该 feature maps 被共享用于后续 RPN 层和全连接层。
- 候选检测框生成网络（RPN）：RPN 网络用于生成 region proposals。该层通过 softmax 判断 anchors 属于 positive 或者 negative，再利用 bounding box regression 修正 anchors 获得精确的 proposals。
- 感兴趣区域池化（RoI Pooling）：该层收集输入的 feature maps 和 proposals，综合这些信息后提取 proposal feature maps，送入后续全连接层判定目标类别。
- 分类器（classifier）：利用 proposal feature maps 计算 proposal 的类别，同时再次 bounding box regression 获得检测框最终的精确位置。

2.2.2 模型训练

2.2.3 模型评价

2.2.4 模型可视化

2.3 YOLO V3 模型