区域卷积神经网络（R-CNN）系列¶

区域卷积神经网络（region-based CNN 或 regions with CNN features，R-CNN）是将深度模型应用于目标检测的开创性工作之一 [1]。在本节中，我们将介绍 R-CNN 和它的一系列改进方法：快速的 R-CNN（Fast R-CNN）[3]、更快的 R-CNN（Faster R-CNN）[4] 以及掩码 R-CNN（Mask R-CNN）[5]。限于篇幅，这里只介绍这些模型的设计思路。

1. R-CNN¶

R-CNN 首先对图像选取若干提议区域（如锚框也是一种选取方法）并标注它们的类别和边界框（如偏移量）。然后，用卷积神经网络对每个提议区域做前向计算抽取特征。之后，我们用每个提议区域的特征预测类别和边界框。图 9.5 描述了 R-CNN 模型。

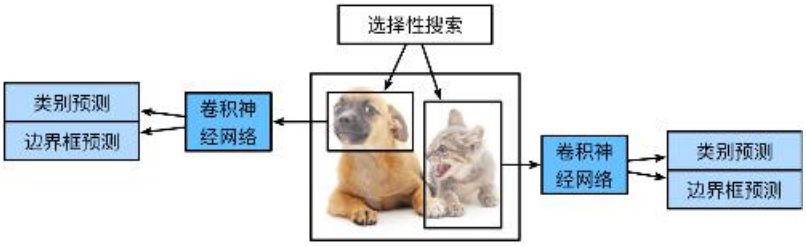

图 9.5 R-CNN 模型

具体来说，R-CNN 主要由以下 4 步构成。

1.

对输入图像使用选择性搜索（selective search）来选取多个高质量的提议区域 [2]。这些提议区域通常是在多个尺度下选取的，并具有不同的形状和大小。每个提议区域将被标注类别和真实边界框。

2.

选取一个预训练的卷积神经网络，并将其在输出层之前截断。将每个提议区域变形为网络需要的输入尺寸，并通过前向计算输出抽取的提议区域特征。

3.

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。其中每个支持向量机用来判断样本是否属于某一个类别。

4.

将每个提议区域的特征连同其标注的边界框作为一个样本，训练线性回归模型来预测真实边界框。

R-CNN 虽然通过预训练的卷积神经网络有效抽取了图像特征，但它的主要缺点是速度慢。想象一下，我们可能从一张图像中选出上千个提议区域，对该图像做目标检测将导致上千次的卷积神经网络的前向计算。这个巨大的计算量令 R-CNN 难以在实际应用中被广泛采用。

2. Fast R-CNN¶

R-CNN 的主要性能瓶颈在于需要对每个提议区域独立抽取特征。由于这些区域通常有大量重叠，独立的特征抽取会导致大量的重复计算。Fast R-CNN 对 R-CNN 的一个主要改进在于只对整个图像做卷积神经网络的前向计算。

图 9.6 描述了 Fast R-CNN 模型。

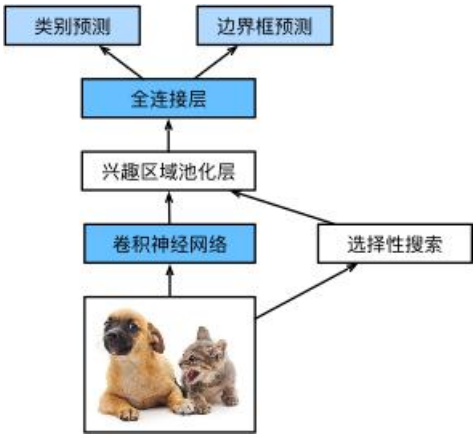

图 9.6 Fast R-CNN 模型

它的主要计算步骤如下。

1.

与 R-CNN 相比，Fast R-CNN 用来提取特征的卷积神经网络的输入是整个图像，而不是各个提议区域。而且，这个网络通常会参与训练，即更新模型参数。设输入为一张图像，将卷积神经网络的输出的形状记为 1×c×h1×w11×c×h1×w1 1 \times c \times h_1 \times w_1 。

2.

假设选择性搜索生成 nn n 个提议区域。这些形状各异的提议区域在卷积神经网络的输出上分别标出形状各异的兴趣区域。这些兴趣区域需要抽取出形状相同的特征（假设高和宽均分别指定为 h2h2 h_2 和 w2w2 w_2 ）以便于连结后输出。Fast R-CNN 引入兴趣区域池化（region of interest pooling，RoI 池化）层，将卷积神经网络的输出和提议区域作为输入，输出连结后的各个提议区域抽取的特征，形状为 n×c×h2×w2n×c×h2×w2 n \times c \times h_2 \times w_2 。

3.

通过全连接层将输出形状变换为 n×dn×d n \times d ，其中超参数 dd d 取决于模型设计。

4.

预测类别时，将全连接层的输出的形状再变换为 n×qn×q n \times q 并使用 softmax 回归（qq q 为类别个数）。预测边界框时，将全连接层的输出的形状变换为 n×4n×4 n \times 4 。也就是说，我们为每个提议区域预测类别和边界框。

5.

在池化层中，我们通过设置池化窗口、填充和步幅来控制输出形状。而兴趣区域池化层对每个区域的输出形状是可以直接指定的，例如，指定每个区域输出的高和宽分别为 h2h2 h_2 和 w2w2 w_2 。假设某一兴趣区域窗口的高和宽分别为 hh h 和 ww w ，该窗口将被划分为形状为 h2×w2h2×w2 h_2 \times w_2 的子窗口网格，且每个子窗口的大小大约为(h/h2)×(w/w2)(h/h2)×(w/w2) (h/h_2) \times (w/w_2) 。任一子窗口的高和宽要取整，其中的最大元素作为该子窗口的输出。因此，兴趣区域池化层可从形状各异的兴趣区域中均抽取出形状相同的特征。

图 9.7 中，我们在 4×44×4 4 \times 4 的输入上选取了左上角的 3×33×3 3\times 3 区域作为兴趣区域。对于该兴趣区域，我们通过 2×22×2 2\times 2 兴趣区域池化层得到一个 2×22×2 2\times 2 的输出。4 个划分后的子窗口分别含有元素 0、1、4、5（5 最大），2、6（6 最大），8、9（9 最大），10。

图 9.7 2×22×2 2\times 2 兴趣区域池化层

我们使用 `ROIPooling` 函数来演示兴趣区域池化层的计算。假设卷积神经网络抽取的特征 `X` 的高和宽均为 4 且只有单通道。

```
In [1]:

from mxnet import nd


X = nd.arange(16).reshape((1, 1, 4, 4))

X

Out[1]:

[[[[ 0.  1.  2.  3.]

   [ 4.  5.  6.  7.]

   [ 8.  9. 10. 11.]
```

```
   [12. 13. 14. 15.]]]]

<NDArray 1x1x4x4 @cpu(0)>
```

假设图像的高和宽均为 40 像素。再假设选择性搜索在图像上生成了两个提议区域：每个区域由 5 个元素表示，分别为区域目标类别、左上角的 xx x 和 yy y 轴坐标以及右下角的 xx x 和 yy y 轴坐标。

```
In [2]:

rois = nd.array([[0, 0, 0, 20, 20], [0, 0, 10, 30, 30]])
```

由于 `X` 的高和宽是图像的高和宽的 1/101/10 1/10 ，以上两个提议区域中的坐标先按 `spatial_scale` 自乘 0.1，然后在 `X` 上分别标出兴趣区域 `X[:,:,0:3,0:3]` 和 `X[:,:,1:4,0:4]`。最后对这两个兴趣区域分别划分子窗口网格并抽取高和宽为 2 的特征。

```
In [3]:

nd.ROIPooling(X, rois, pooled_size=(2, 2), spatial_scale=0.1)

Out[3]:

[[[[ 5.  6.]
   [ 9. 10.]]]


 [[[ 9. 11.]
   [13. 15.]]]]

<NDArray 2x1x2x2 @cpu(0)>
```

3. Faster R-CNN¶

Fast R-CNN 通常需要在选择性搜索中生成较多的提议区域，以获得较精确的目标检测结果。Faster R-CNN 提出将选择性搜索替换成区域提议网络（region proposal network），从而减少提议区域的生成数量，并保证目标检测的精度。

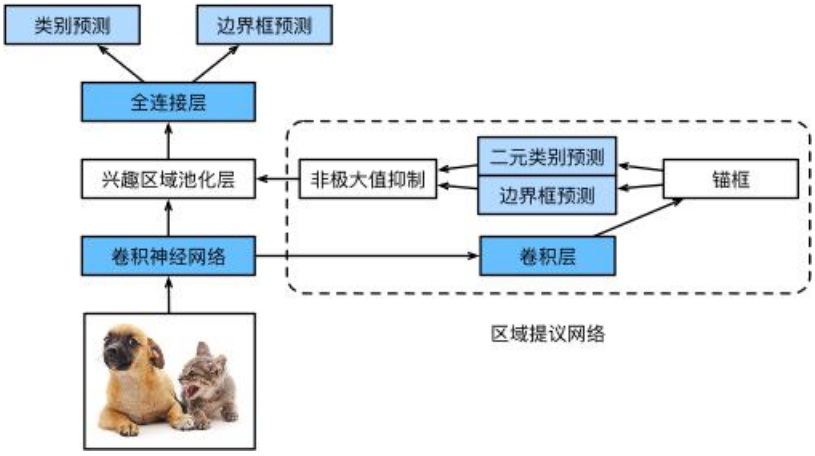

图 9.8 Faster R-CNN 模型

图 9.8 描述了 Faster R-CNN 模型。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。具体来说，区域提议网络的计算步骤如下：

1.

使用填充为 1 的 3×33×3 3\times 3 卷积层变换卷积神经网络的输出，并将输出通道数记为 cc c 。这样，卷积神经网络为图像抽取的特征图中的每个单元均得到一个长度为 cc c 的新特征。

2.

以特征图每个单元为中心，生成多个不同大小和宽高比的锚框并标注它们。

3.

用锚框中心单元长度为 cc c 的特征分别预测该锚框的二元类别（含目标还是背景）和边界框。

4.

使用非极大值抑制，从预测类别为目标的预测边界框中移除相似的结果。最终输出的预测边界框即兴趣区域池化层所需要的提议区域。

值得一提的是，区域提议网络作为 Faster R-CNN 的一部分，是和整个模型一起训练得到的。也就是说，Faster R-CNN 的目标函数既包括目标检测中的类别和边界框预测，又包括区域提议网络中锚框的二元类别和边界框预测。最终，区域提议网络能够学习到如何生成高质量的提议区域，从而在减少提议区域数量的情况下也能保证目标检测的精度。

4. 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。

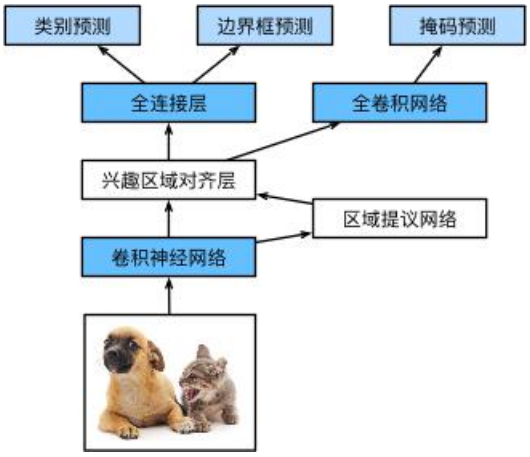

图 9.9 Mask R-CNN 模型

如图 9.9 所示，Mask R-CNN 在 Faster R-CNN 的基础上做了修改。Mask R-CNN 将兴趣区域池化层替换成了兴趣区域对齐层，即通过双线性插值（bilinear interpolation）来保留特征图上的空间信息，从而更适于像素级预测。兴趣区域对齐层的输出包含了所有兴趣区域的形状相同的特征图。它们既用来预测兴趣区域的类别和边界框，又通过额外的全卷积网络预测目标的像素级位置。

5. 参考文献¶

[1] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[2] Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. International journal of computer vision, 104(2), 154-171.

[3] Girshick, R. (2015). Fast r-cnn. arXiv preprint arXiv:1504.08083.

[4] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[5] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017, October). Mask r-cnn. In Computer Vision (ICCV), 2017 IEEE International Conference on (pp. 2980-2988). IEEE.

[6] GluonCV 工具包。https://gluon-cv.mxnet.io/