第 2 章　目标检测与目标跟踪理论知识

路口拥堵检测的关键在于对车辆的精准识别与连续跟踪。通过准确检测车辆位置、数量及速度，评估拥堵状态。为实现这一目标，目标检测与目标跟踪技术成为拥堵检测的核心支撑。目标检测用于识别图像中每一帧车辆的位置和类别，而目标跟踪则负责在连续帧中保持车辆身份一致，获取其运动轨迹。这两项技术的协同配合，能够实现车流量统计，是构建拥堵评估模型的基础。因此，本章将围绕目标检测与目标跟踪技术展开，系统介绍其核心理论与关键方法。首先介绍 CNN 的基本组成部分，接着介绍两类主流的目标检测算法 FasterR-CNN 两阶段和 YOLOv8 单阶段，最后介绍当前广泛应用于多目标跟踪的 SORT 与 DeepSORT 算法，为后续章节提供理论支撑。

2.1　卷积神经网络

CNN 是深度学习中的核心模型之一，在目标检测中起着至关重要的作用。CNN 通过学习图像中的局部特征，能够从复杂的视觉数据中提取有用的特征。其网络结构图如图 2.1 所示，基本结构包括输入层、卷积层、池化层和全连接层。通过上述层协同作用，CNN 能够实现准确的目标分类与定位。经典的 CNN 包括 LeNet-5、AlexNet、VGGNet、GoogLeNet 和 ResNet 等，它们在计算机视觉任务中提供了强大的特征提取能力。下面分小节介绍 CNN 网络所用到的核心模块。

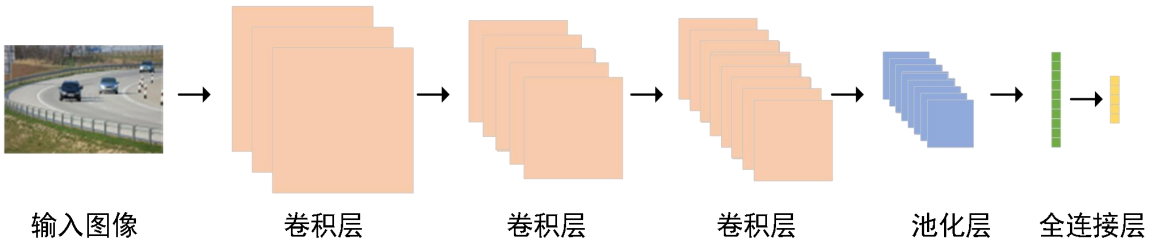


图 2.1　CNN 结构图

2.1.1　卷积层

卷积层是 CNN 中的关键部分，主要用于提取输入数据的特征。通过与图像进行卷积操作，卷积层能够识别图像的局部特征，如边缘和纹理。卷积操作通过卷积核滑动整个图像，生成特征图，该层的权重共享机制使得计算更加高效，且能保留图像的空间结构，适用于目标检测任务中的位置和空间关系学习。卷积层计算过程如图 2.2 所示。

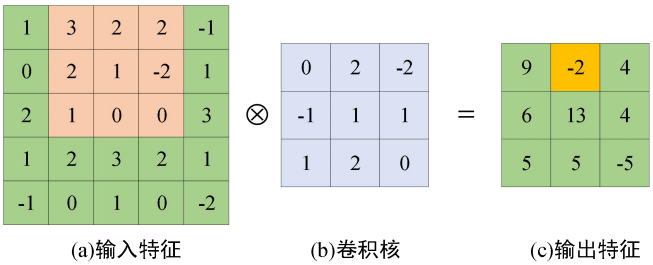


图 2.2　卷积计算示意图

2.1.2　池化层

池化层是 CNN 中的重要组成部分，主要用于减少特征图的尺寸，降低计算复杂度，并防止过拟合。池化操作通过在特征图上滑动一个窗口，计算窗口内的最大值或平均值，生成新的特征图。常见的池化方式有最大池化和平均池化：其中最大池化选择窗口中的最大值，能够保留最显著的特征；平均池化计算窗口内像素的平均值，更注重整体信息的平滑表达。最大池化与平均池化计算过程如图 2.3 所示。

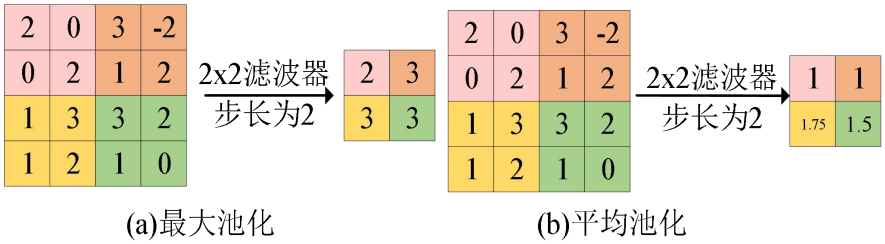


图 2.3　最大池化与平均池化计算过程图

2.1.3　全连接层

全连接层通常位于网络的末端，用于将前面提取到的特征进行整合。在全连接层中，每个神经元都与前一层的所有神经元相连接。它的主要功能是将卷积层和池化层提取到的局部特征转化为全局特征，并为最终的分类或回归任务提供输入。全连接层结构示意图如图 2.4 所示。

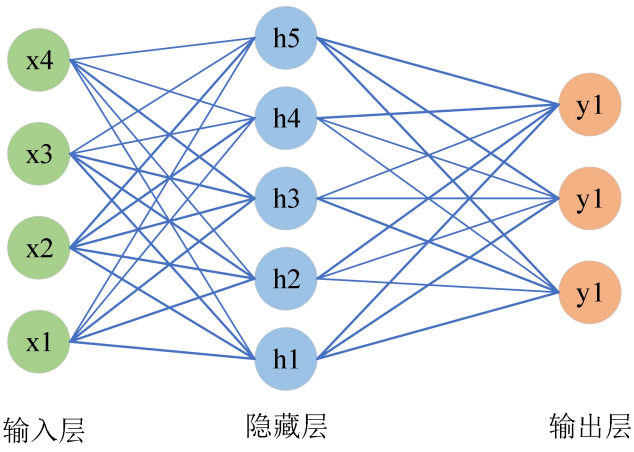


图 2.4　全连接层结构示意图

2.1.4　激活函数

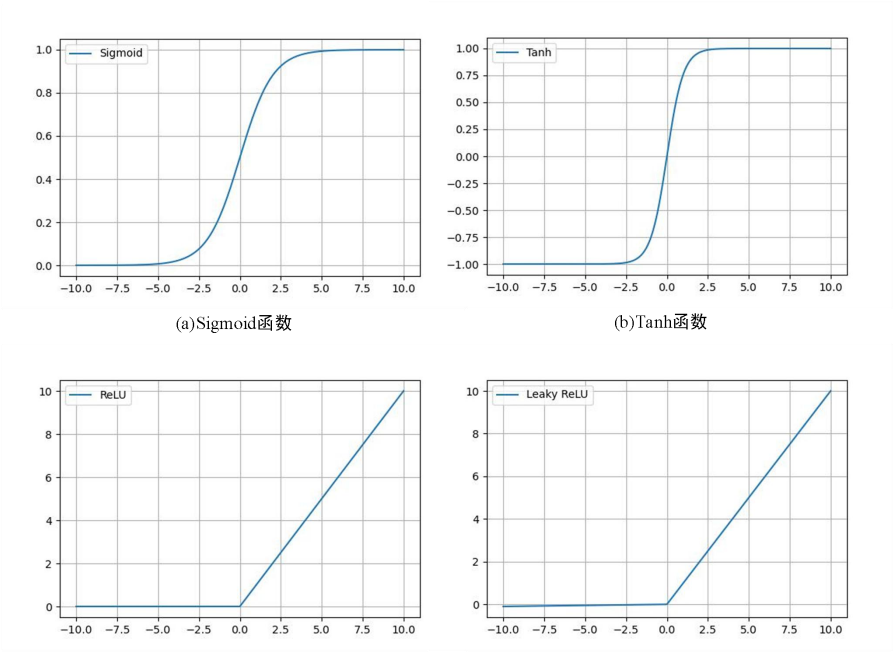
在深度学习中，激活函数是模型中的重要组成部分，其主要作用是为 CNN 引入非线性变换，使得网络能够拟合复杂的非线性关系。常见的激活函数有 Sigmoid、Tanh、ReLU 和 Leaky ReLU，四种激活函数图像如图 2.5 所示。

Sigmoid 函数是最早应用于 CNN 中的激活函数之一。它将输入值压缩到 0 到 1 之间，常用于二分类问题的输出层。其公式如式（2-1）所示。该函数的输出是一个平滑的 S 型曲线，具有将任何输入映射到 0 到 1 概率区间的性质。然而，Sigmoid 函数容易出现梯度消失问题，尤其是在深层网络中。当输入值远离 0 时，Sigmoid 输出趋近于 0 或 1，此时梯度接近于 0，导致反向传播时梯度逐层衰减。

$$\sigma(x)=\frac{1}{1+e^{-x}} \tag{2-1}$$

Tanh 函数是 Sigmoid 函数的变种，其输出范围为-1 到 1，具有更强的梯度，因此能够缓解一些梯度消失的问题。其公式如式（2-2）所示。该函数的输出在 -1 和 1 之间，具有中心对称的特点，这使得其在处理数据时能够更好地保持负值信息。但在输入值极大或极小时，仍然可能出现梯度消失的问题。

$$\mathrm{Tanh}(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{2-2}$$



(a)Sigmoid函数　(b)Tanh函数

(c)ReLU函数　(d)LeakyReLU函数

图 2.5　激活函数图

ReLU 是目前最常用的激活函数之一，它将输入值小于零的部分输出零，输入值大于零时则保持原值。其公式如式（2-3）所示。ReLU 能够有效解决梯度消失问题，加速神经网络的训练过程。然而，当输入值为负时，ReLU 会导致“死神经元”问题，即神经元永远不再激活。

$$\mathrm{Re}\,LU(x)=\max(0,x) \tag{2-3}$$

为了克服 ReLU 的死神经元问题，Leaky ReLU 提出了一个小的斜率来使得负输入也能够产生输出。其公式如式（2-4）所示。其中，α 是一个小的常数（通常为 0.01）。Leaky ReLU 允许在负值输入时输出一个小的负值，从而避免了神经元的完全失效。

$$Leaky\,\mathrm{Re}\,LU(x)=\max(\alpha x,x) \tag{2-4}$$

2.1.5　损失函数

损失函数是深度学习中的核心模块之一，用于衡量模型预测结果与真实标签之间的差距。通过最小化损失函数，模型能够在反向传播中不断优化其参数，从而提高其