

计算机视觉基础

- 第一章 概论
- 第二章 基础知识
- 第三章 图像分类
- 第四章 图像语义分割
- 第五章 目标检测
- 第六章 识别
- 第七章 目标跟踪
- 第八章 多目视觉
- 第九章 视觉问答

第二章 基础知识

- 2.1 数字图像
- 2.2 照相机成像模型
- 2.3 传统计算机视觉方法基础知识
- 2.4 深度学习基础知识

1. 灰度图像:

- 灰度图像只记录图像中的每个像素点的强度信息，也称为灰度值，但不记录跟颜色有关的信息。

2.1 数字图像

[illegible]

图2-1 灰度图像示例

2.1 数字图像

数字图像表示

2. 彩色图像:

2.1 RGB颜色模型

- 红 (Red, 缩写为R)、绿 (Green, 缩写为G)、蓝 (Blue, 缩写为B)

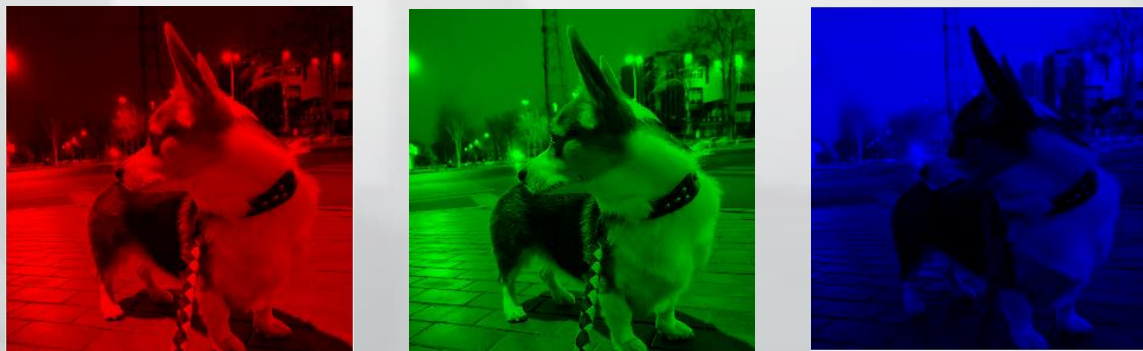


图2-2 彩色图像的RGB三通道表示

2.1 数字图像

2. 彩色图像:

2.1 RGB颜色模型

- 彩色图像和灰度图像之间可以相互转换。如果用 $Gray$ 表示灰度图像的强度值, R 、 G 、 B 分别表示RGB三个颜色通道的强度, 那么灰度图像可以通过如下公式计算得到:
 $Gray=0.3R+0.59G+0.11B$.

数字图像表示

2. 彩色图像:

2.1 HSV模型

- 色调 (Hue, 缩写为H)、饱和度 (Saturation, 缩写为S)、明度 (Value, 缩写为V)

2.1 数字图像



图2-3 彩色图像的HSV三通道表示实例

2.1 数字图像

数字图像表示

➤ RGB颜色通道表示形式和HSV颜色通道表示形式可以相互转换。例如，RGB颜色通道转化成HSV颜色通道的计算方法为：

$$H = \begin{cases} 60 \times \frac{G - B}{\max - \min}, & \text{if } R = \max; \\ 120 + 60 \times \frac{B - R}{\max - \min}, & \text{if } G = \max; \\ 240 + 60 \times \frac{R - G}{\max - \min}, & \text{if } B = \max; \\ H = H + 360, & \text{if } H < 0; \end{cases}$$
$$S = \frac{\max - \min}{\max};$$
$$V = \max.$$

其中， $\max$ 和 $\min$ 分别为三通道强度值的最大值和最小值：
 $\max = \max(R, G, B)$ ， $\min = \min(R, G, B)$ 。

2.1 数字图像

3. 深度图像

- 深度图像可以作为数字图像的一个通道，记录的是照相机到物体表面的距离。
- 一般获取深度图像的照相机同时有彩色图像摄像头，所以深度图像通常跟彩色图像绑定在一起，被称为RGB-D图像。



图2-4 RGB-D图像实例

2.2 照相机成像模型

离镜头距离为 Z 的物体 P 通过透镜成像到焦平面 F

物体 P 成像为 p

镜头光圈孔径大小为 d

镜头焦距大小为 f

物体及其影像与透镜间的距离和透镜焦距之间的关系满足： $\frac{1}{z} + \frac{1}{Z} = \frac{1}{f}$

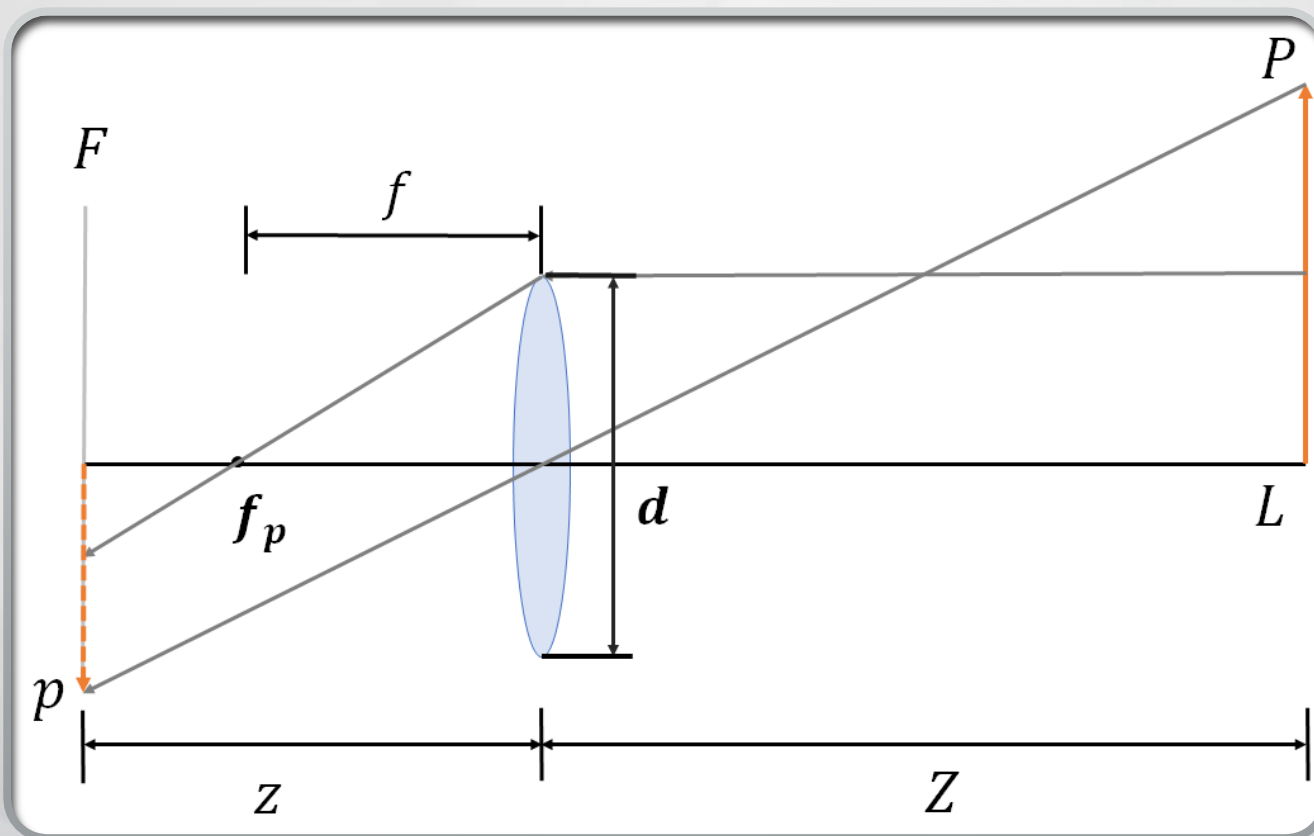


图2-4 RGB-D图像实例

2.3 传统计算机视觉方法基础知识

传统方法的主要组成模块

1. 图像预处理

- 目标对齐、几何归一化等处理，以便提高不同数据之间的一致性

2. 特征提取和表示

- 提取反应图像低层（如边缘边界信息）、中层（如物体的组成部件）或高层（如物体或场景信息）的特性
- 对提取的特征进行描述、汇聚和变换（例如词袋模型）

3. 利用数据训练机器学习模型

- 主要是分类器（例如支持向量机SVM）或者是回归器进行从特征到数据标签的映射模型学习

2.3 传统计算机 视觉方法基础 知识

1. 图像预处理

- 1.1 目标对齐
- 1.2 几何归一化
- 1.3 最大最小值归一化

2.3 传统计算机 视觉方法基础 知识

2. 特征提取和表示

2.1 尺度不变特征变换 (SIFT)

2.1.1 多尺度空间极值检测

2.1.2 关键点方向分配

2.1.3 关键点描述符

2.1.4 关键点匹配

2. 特征提取和表示

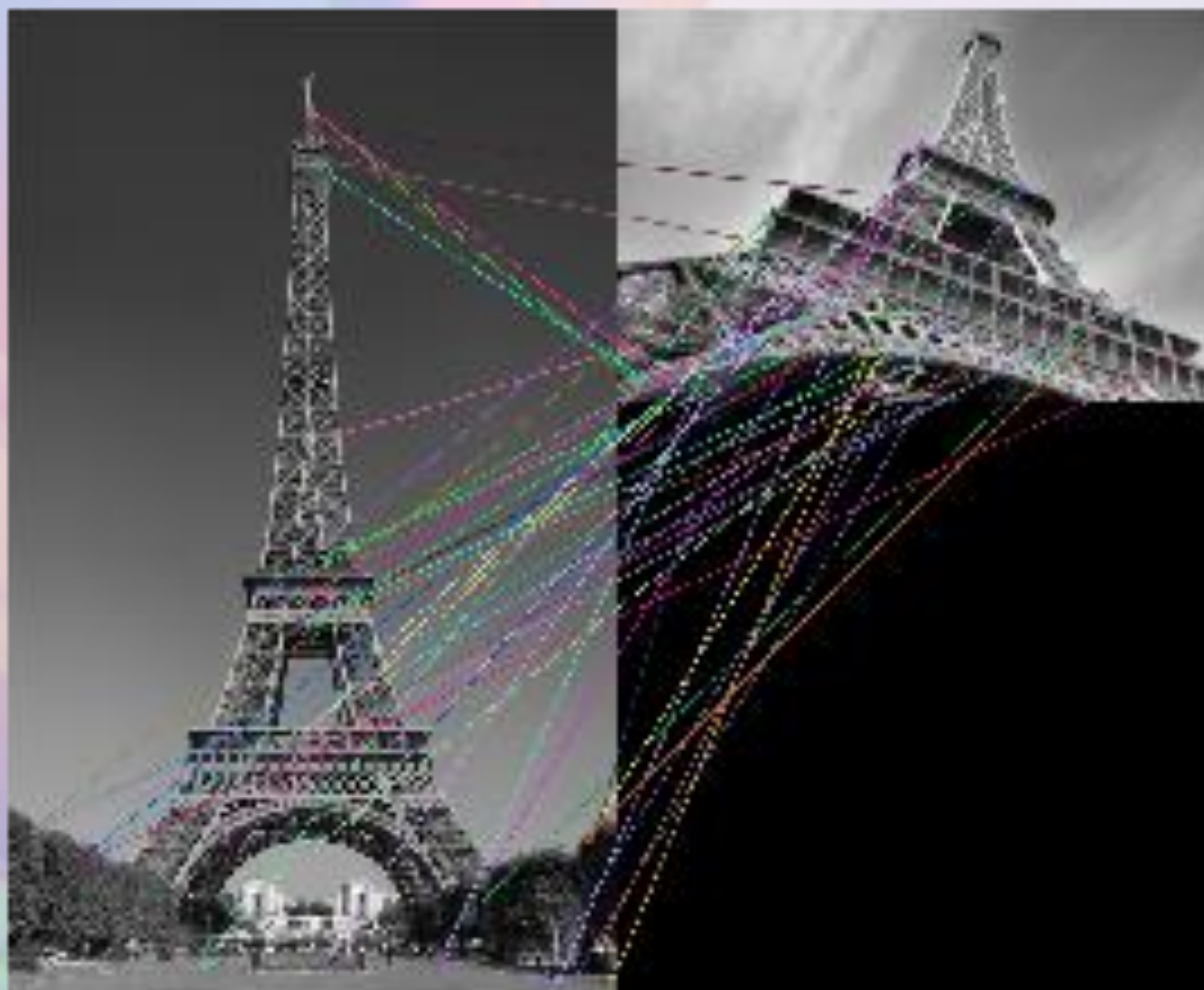
2.1 尺度不变特征变换 (SIFT)

2.3 传统计算机 视觉方法基础 知识

2.3 传统计算机视觉方法基础知识-SIFT特征点匹配



1. 从图1中提取SIFT特征并对提取的特征可视化



2. 对两幅图像中的SIFT特征进行匹配的结果

2.3 传统计算机视觉方法基础知识

2. 特征提取和表示

2.1 尺度不变特征变换 (SIFT)

2.1.1 多尺度空间极值检测

a) 计算图像的高斯拉普拉斯算子
(全称Laplacian of Gaussian, 缩写LOG)

- LOG的主要作用是在不同的尺度上进行区域监测
- 因为LOG计算量比较大, 通常都是用高斯差分 (全称Difference of Gaussian, 缩写DOG) 计算

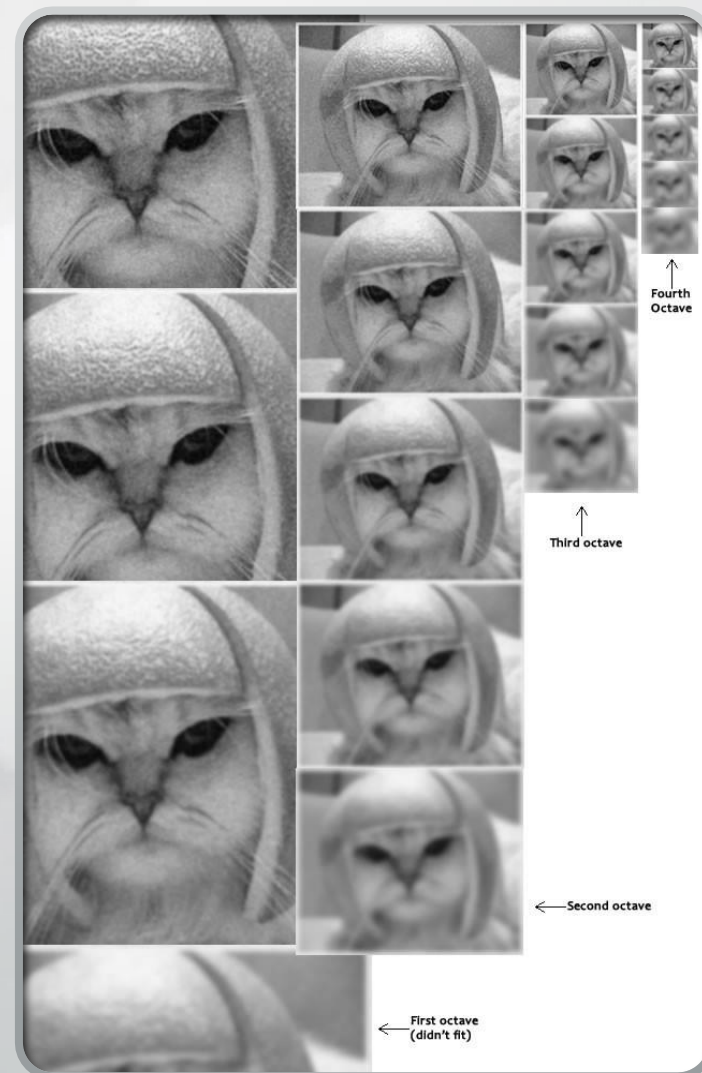
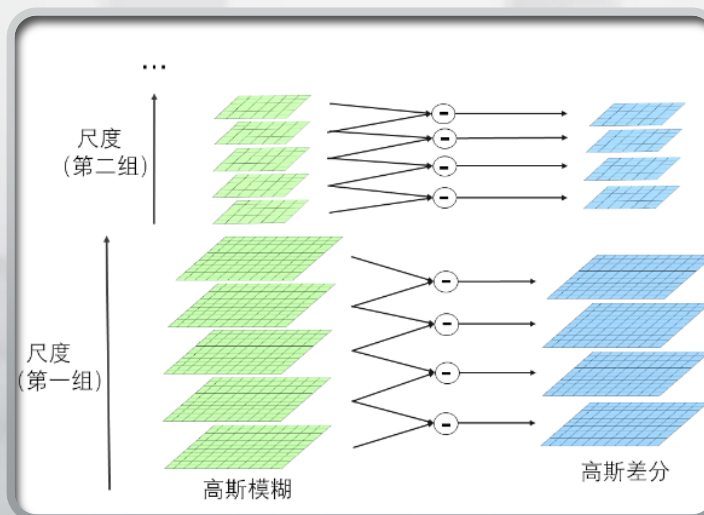
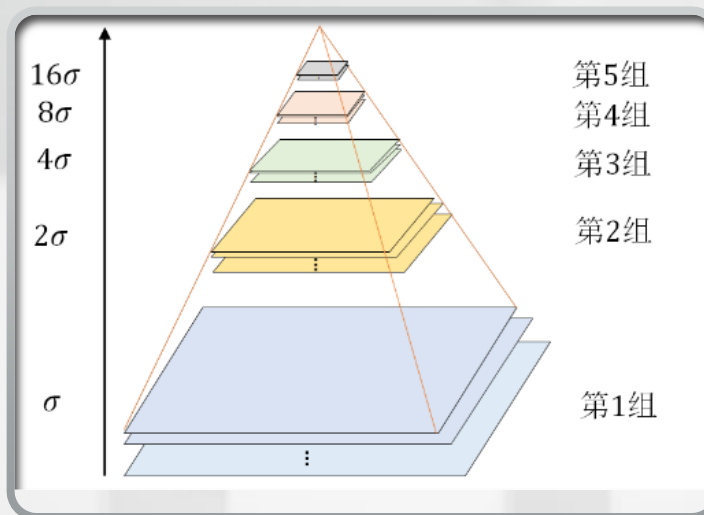
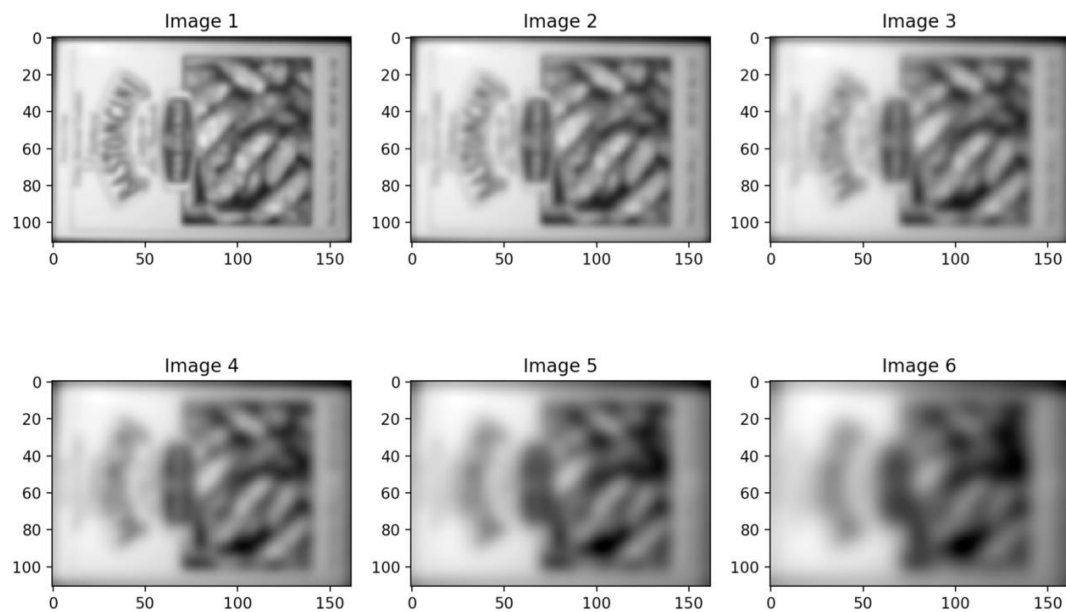
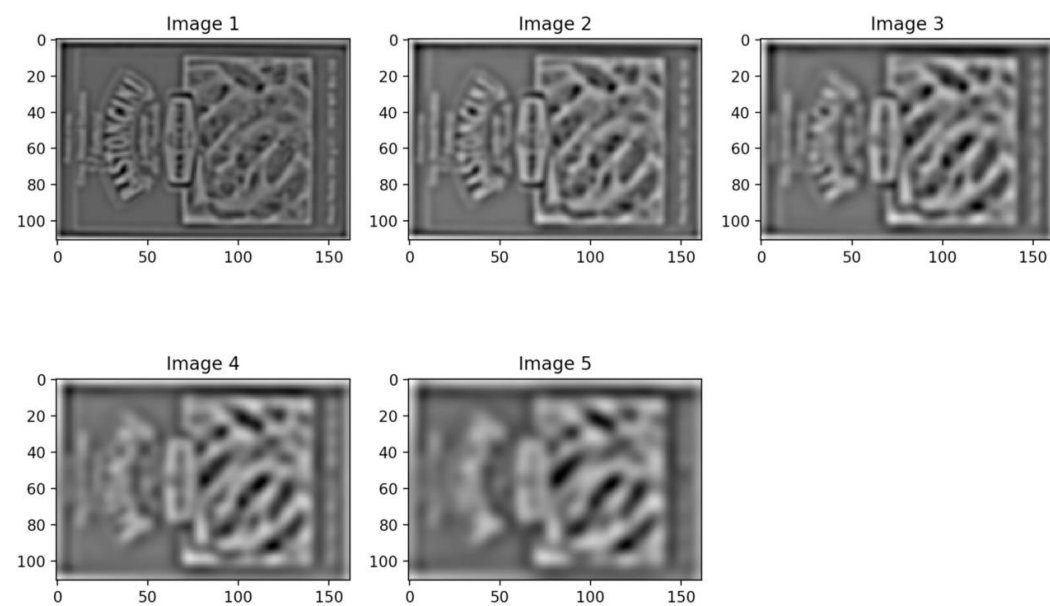


图2-7 高斯差分计算示意图。左上和右图：高斯模糊，左下：高斯差分。

2.3 传统计算机视觉方法基础知识-SIFT-DoG



1. 高斯图金字塔第三层



2. 计算得到的第三层DoG

高斯差分计算结果图

2.3 传统计算机视觉方法基础知识

2. 特征提取和表示

2.1 尺度不变特征变换 (SIFT)

2.1.1 多尺度空间极值检测

- 计算图像的高斯拉普拉斯算子 (全称 Laplacian of Gaussian, 缩写LOG)
- 对所有尺度的所有图像空间进行搜索寻找局部极大值
 - 像素点值比它同尺度的8邻域点的像素值大, 且比它的上一尺度对应的9个像素点值及其下一尺度对应的9个像素点值都大

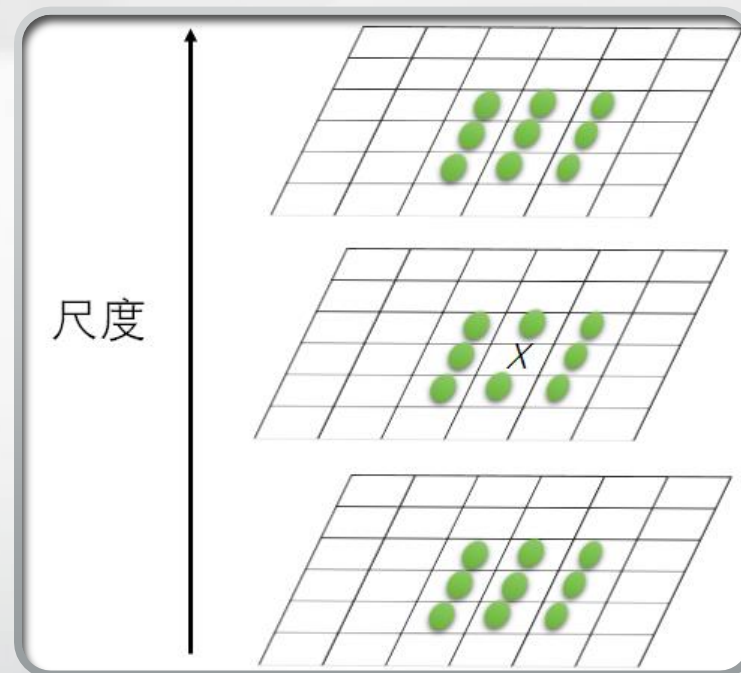


图2-7 局部极值计算示意图

2.3 传统计算机视觉方法基础知识

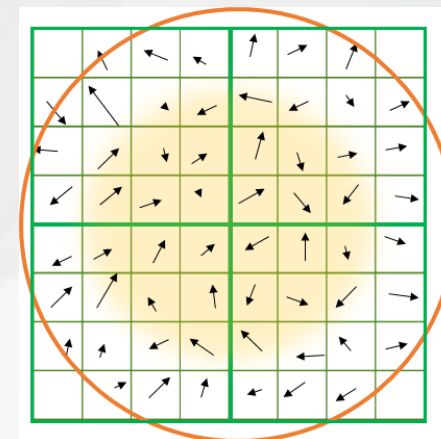
2. 特征提取和表示

2.1 尺度不变特征变换 (SIFT)

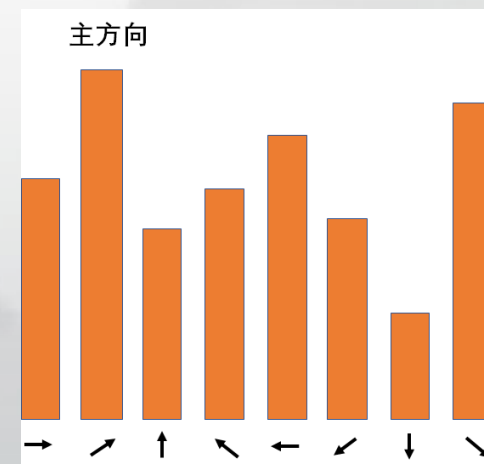
2.1.1 多尺度空间极值检测

2.1.2 关键点方向分配

- 从某个关键点的邻域（子图（a）最大的绿色正方形包含的区域）计算所有采样点（每个绿色小正方形代表的位置）的梯度和方向
- 每个方向都乘以其对应的梯度
- 整个邻域利用一个中心在该关键点的高斯（用橙色圆圈表示）进行加权
- 每个点的方向将其统计到对应方向区域的直方图中（子图（b））
- 取直方图的最大值 $\max$ 对应的方向区间和所有超过 $0.8 \times \max$ 的方向区间来计算该关键点的最终方向



(a) 每个采样点的方向和梯度



(b) 直方图

图2-8 关键点周围采样点方向分配示意图

2.3 传统计算机视觉方法基础知识

2. 特征提取和表示

2.1 尺度不变特征变换 (SIFT)

2.1.1 多尺度空间极值检测

2.1.2 关键点方向分配

2.1.3 关键点描述符

- 取关键点周围 16×16 大小的邻域，将这块区域划分成 16 个 4×4 的子块
- 每个子块计算一个 8 个方向区间的直方图，总共产生 128 个区间
- 这个 128 个区间的直方图就是该关键点的描述符

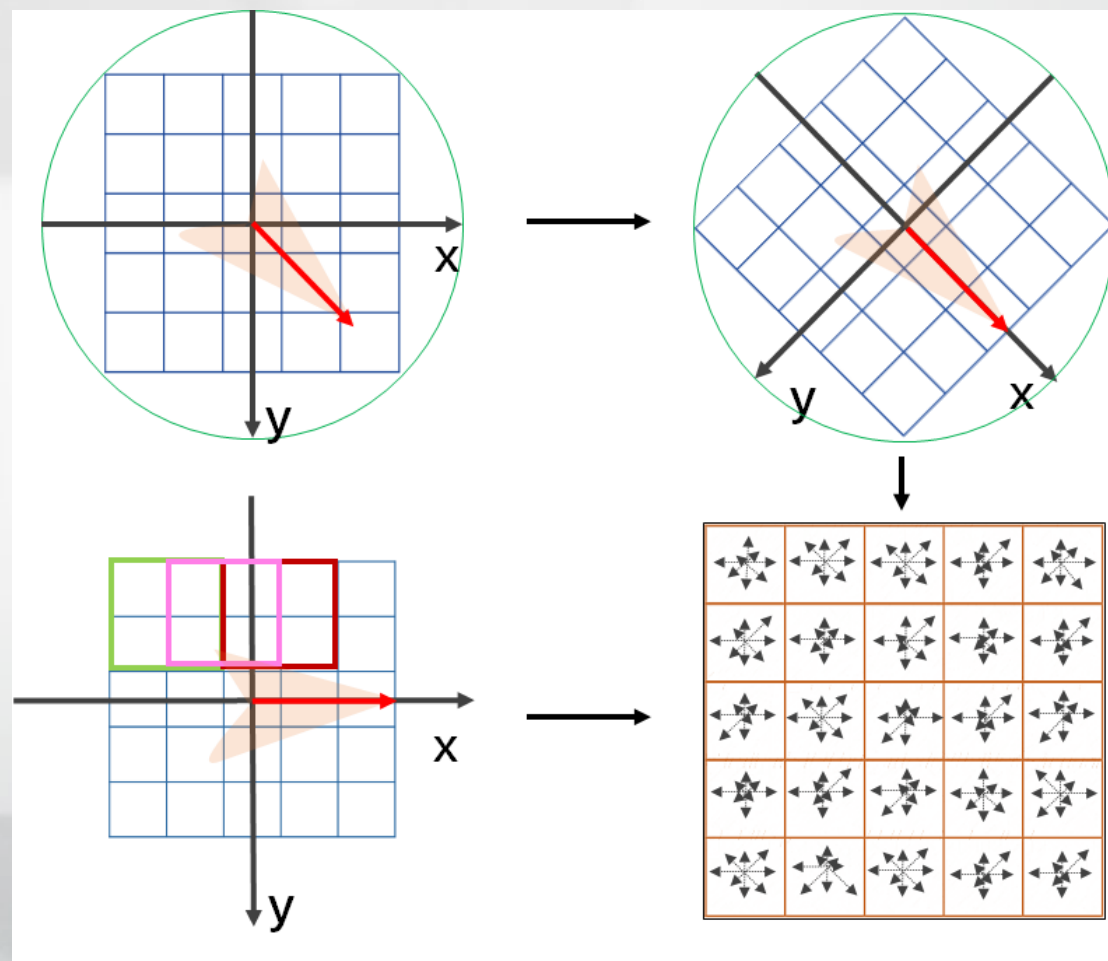


图2-9 关键点描述符计算过程

2.3 传统计算机视觉方法基础知识

2. 特征提取和表示

2.1 尺度不变特征变换 (SIFT)

2.1.1 多尺度空间极值检测

2.1.2 关键点方向分配

2.1.3 关键点描述符

2.1.4 关键点匹配

a) SIFT特征点的一个最直接的应用是进行图像配准 (Image Registration)

- 将不同条件下获取的两幅或者多幅图像进行匹配，以便进行叠加或者拼接

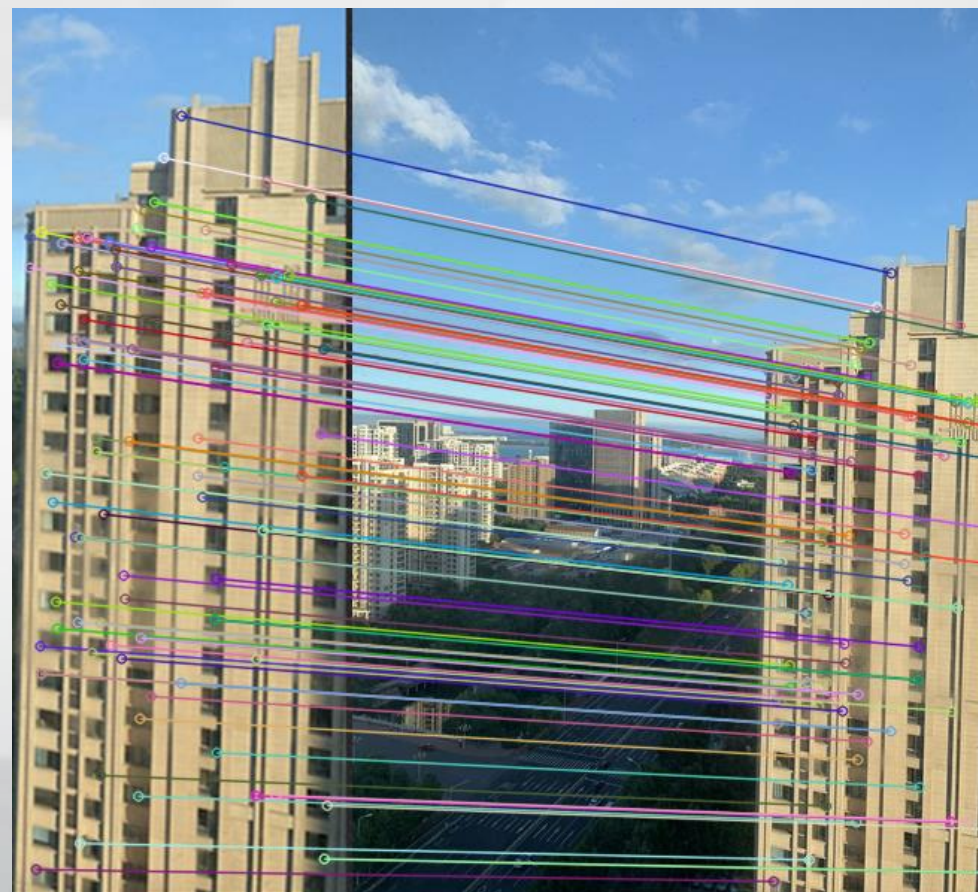


图2-10 SIFT关键点匹配结果示例

2.3 传统计算机 视觉方法基础 知识

2. 特征提取和表示

2.2 方向梯度直方图 (HOG)

2.2.1 梯度图像

2.2.2 像子块的方向梯度直方图

2.2.3 HOG特征描述符

2. 特征提取和表示

2.2 方向梯度直方图 (HOG)

2.3 传统计算机 视觉方法基础 知识

2.3 传统计算机视觉方法基础知识

2. 特征提取和表示

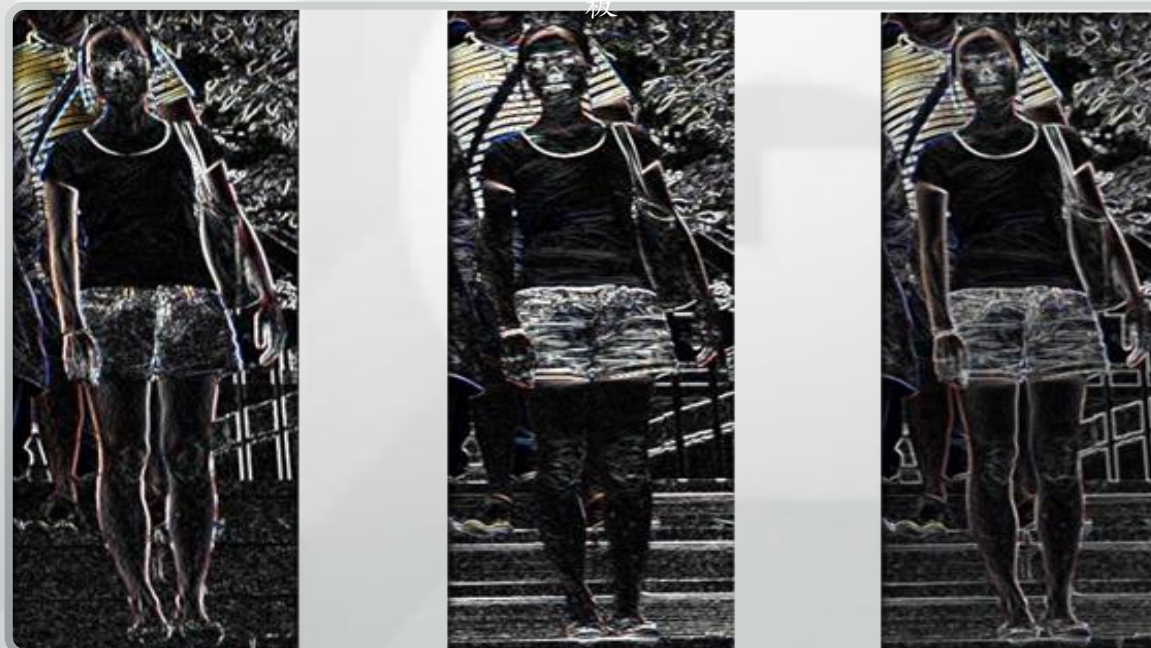
2.2 方向梯度直方图 (HOG)

2.2.1 梯度图像

- a) 利用图2-11中的模板计算图像中的水平和垂直梯度
- 图2-12给出了通过模板滤波计算得到的水平和垂直梯度以及梯度级数示意图

			-1
-1	0	1	0
			1

图2-11 计算水平和垂直梯度的模板



(a) 水平梯度 (或x梯度) (b) 垂直梯度 (或y梯度) (c) 梯度级数

图2-12 水平和垂直梯度计算结果

2.3 传统计算机视觉方法基础知识

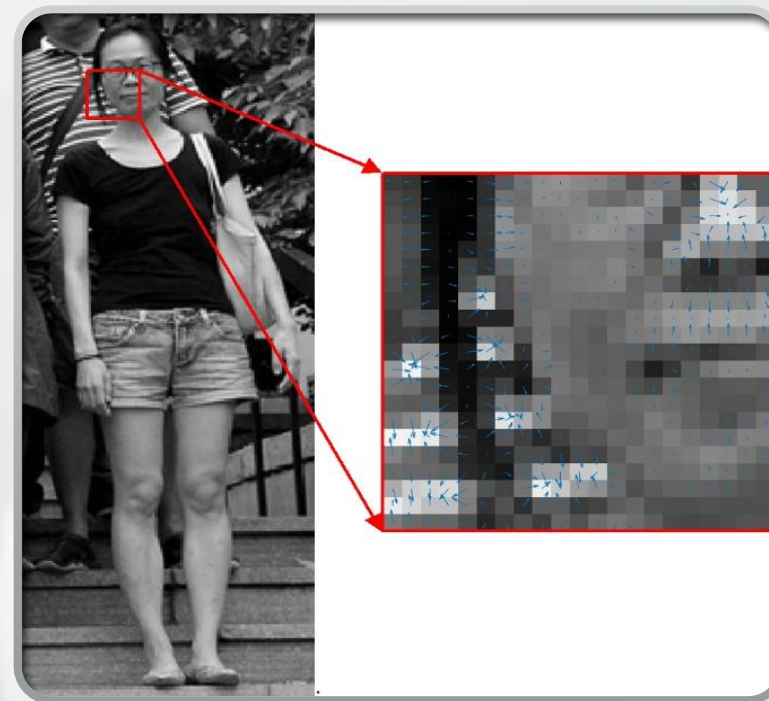
2. 特征提取和表示

2.2 方向梯度直方图 (HOG)

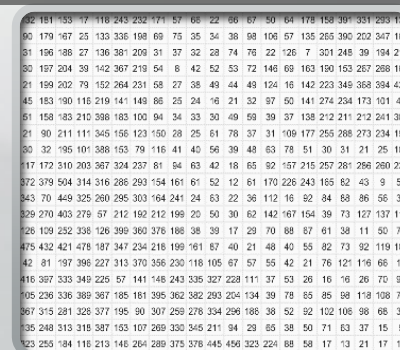
2.2.1 梯度图像

2.2.2 图像子块的方向梯度直方图

- 从人的图像中取出一块图像子块，将其与计算得到的梯度叠加得到图2-13 (a)
 - ✓ 梯度用带蓝色箭头的线段表示，箭头指向的方向表示像素强度变化方向，线段长度表示梯度值的大小
- 图2-13 (b) 显示了该图像子块的梯度值
 - ✓ 梯度方向的范围是0到180度而不是-180到180度，这个梯度称为无符号梯度，因为-180到0度的梯度方向取绝对值后被表示成其对应的正值



(a) 从包含人的图中取出的RGB图像小块和用箭头表示的梯度



(b) 梯度的数值图

图2-13 HOG特征示意图

2.3 传统计算机视觉方法基础知识

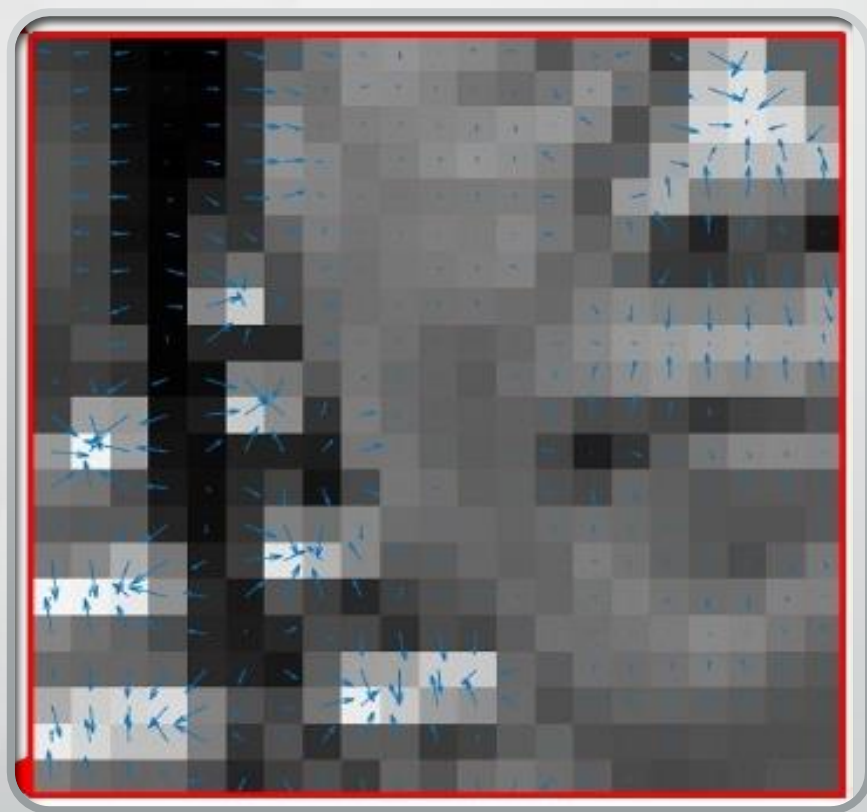
2. 特征提取和表示

2.2 方向梯度直方图 (HOG)

2.2.1 梯度图像

2.2.2 图像子块的方向梯度直方图

- 从人的图像中取出一块图像子块，将其与计算得到的梯度叠加得到图2-13 (a)
 - ✓ 梯度用带蓝色箭头的线段表示，箭头指向的方向表示像素强度变化方向，线段长度表示梯度值的大小
- 图2-13 (b) 显示了该图像子块的梯度值
 - ✓ 梯度方向的范围是0到180度而不是-180到180度，这个梯度称为无符号梯度，因为-180到0度的梯度方向取绝对值后被表示成其对应的正值



2.3传统计算机视觉方法基础知识

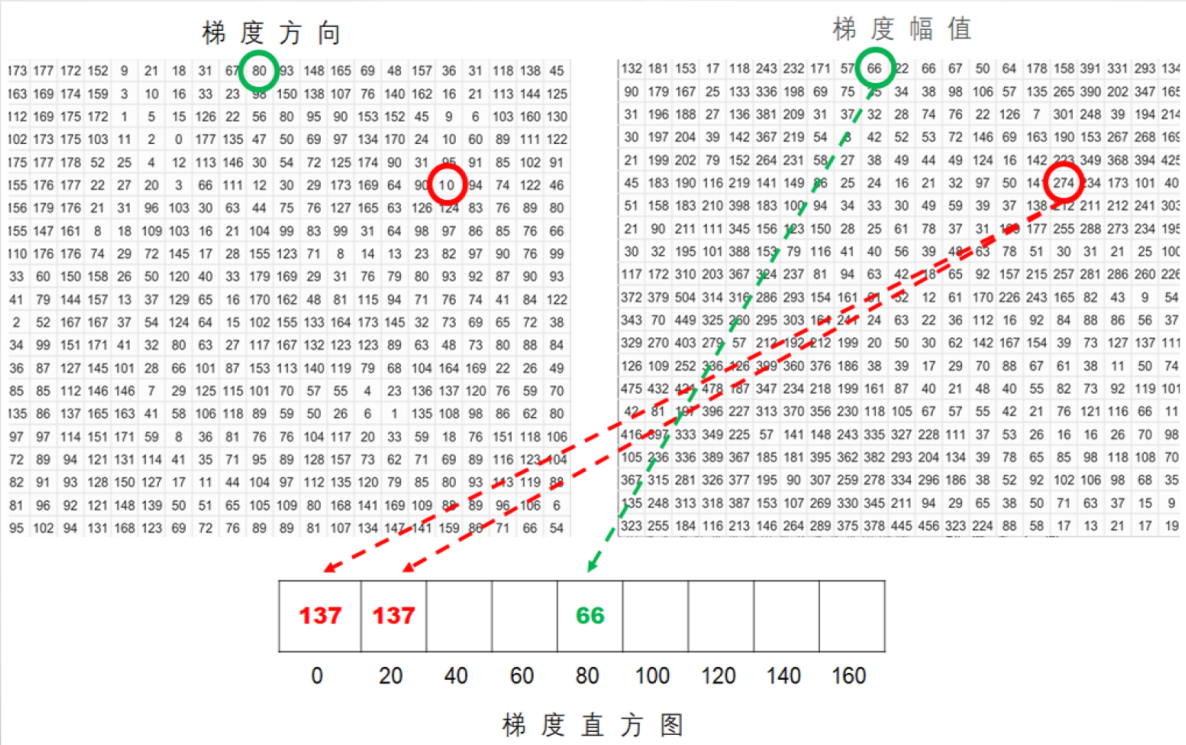
2. 特征提取和表示

2.2 方向梯度直方图 (HOG)

2.2.1 梯度图像

2.2.2 图像子块的方向梯度直方图

- a) 将180度的方向分成9个区间分别对应0,20,40,...,160，根据梯度方向选择区间，表示梯度大小的值放进选好的方向区间
- b) 将整个图像子块的梯度处理完毕之后，可以得到一个直方图的所有值



(a) 水平梯度 (或x梯度) (b) 垂直梯度 (或y梯度) (c) 梯度级数

图2-12 水平和垂直梯度计算结果

2.3 传统计算机视觉方法基础知识

2. 特征提取和表示

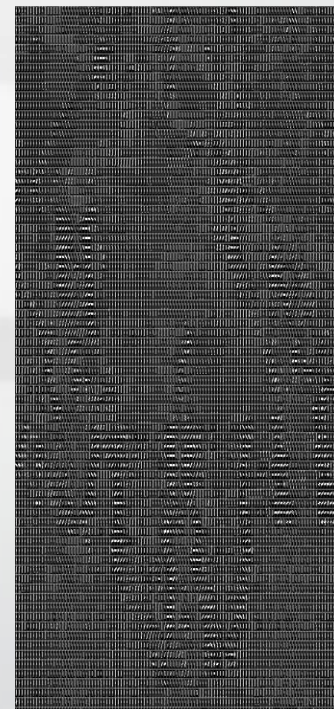
2.2 方向梯度直方图 (HOG)

2.2.1 梯度图像

2.2.2 图像子块的方向梯度直方图

2.2.3 HOG特征描述符

- 将该图划分成 8×16 个图像子块，每个子块提取HOG特征
- 每4个图像子块一起进行归一化处理，处理之后得到了 $4 \times 9 = 36$ 维的直方图
- 在归一化的过程中，每步都取相邻的4个图像子块，计算完一个组合之后，向右下方移动一个图像子块的步长，所以最后得到了 7×15 个归一化的直方图
- 整个HOG特征描述符的长度为3780



(a) 原始图像

(b) 提取的HOG特征

图2-15 计算图像HOG特征示例

2.3 传统计算机 视觉方法基础 知识

3. 传统机器学习模型

3.1 支持向量机

3.2 决策树

3. 传统机器学习模型

3.1 支持向量机

2.3 传统计算机 视觉方法基础 知识

2.4 深度学习基础知识

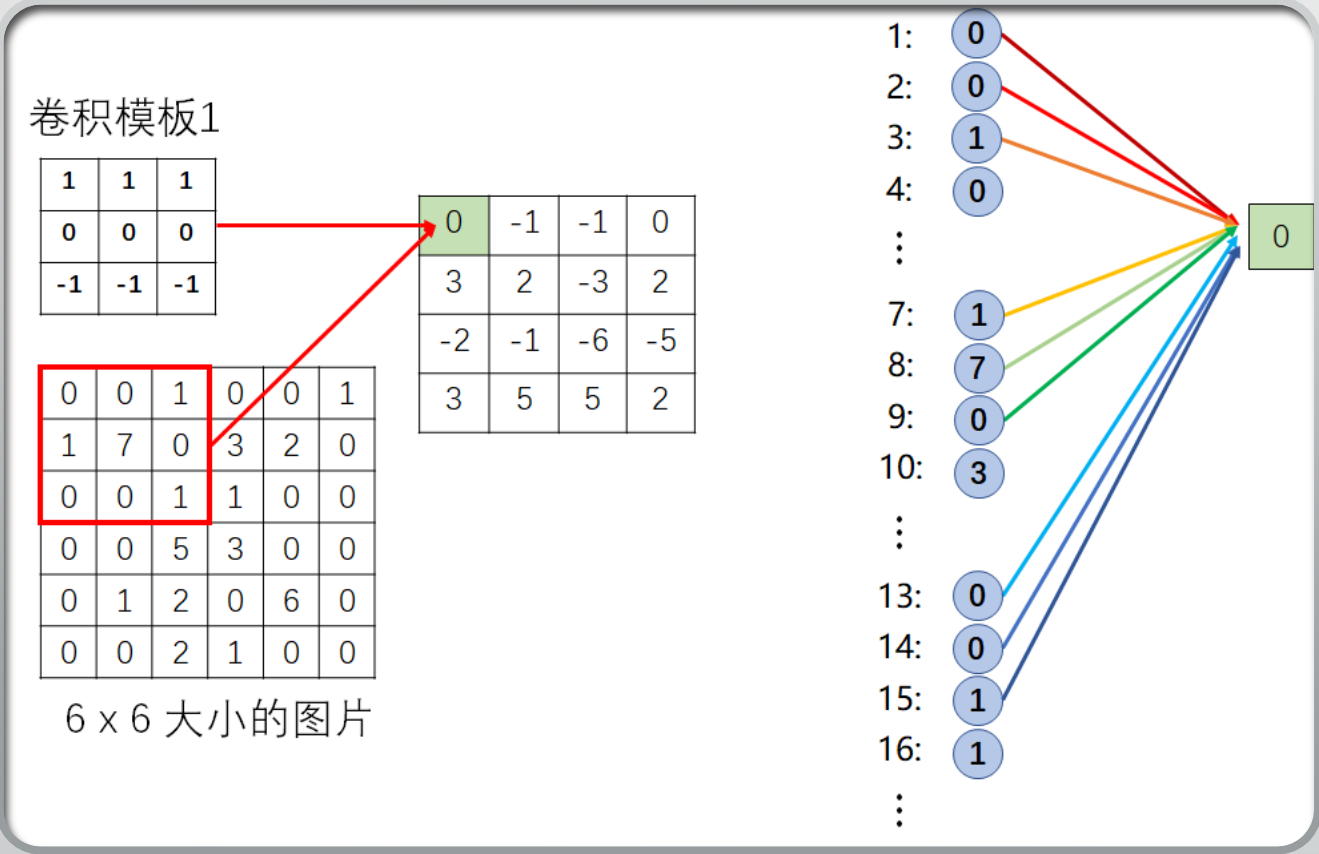
深度学习基本操作

1. 卷积
2. 池化

2.4 深度学习基础知识

1. 卷积

- a) 选取图像中与卷积模板大小相同的区域
- b) 计算并记录卷积结果 $r_{ij} = \sum_{k,l} i_{kl} f_{kl}$ ，其中 i_{kl} 表示目前处理图像区域第 k 行第 l 列位置像素的强度值， f_{kl} 表示卷积模板第 k 行第 l 列位置的值
- c) 从图像中取下一个区域进行处理，直到整幅图像处理完毕



(a) 卷积计算过程 (b) 神经网络结构中的处理示意

图2-16 对6×6大小的图像进行处理得到的结果

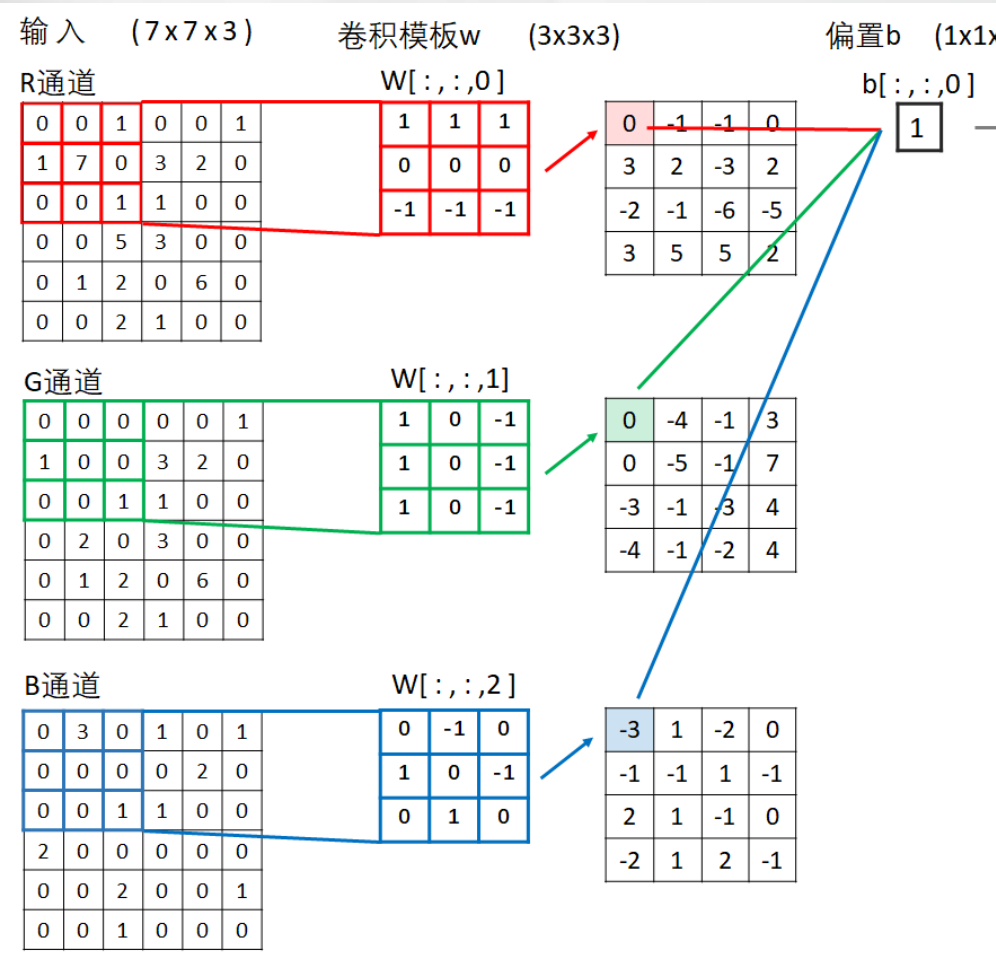
2.4 深度学习基础知识

1. 卷积

➤ RGB彩色图像

a) 对RGB三个颜色通道分别计算卷积

b) 对三个通道卷积的结果进一步处理： $o = \sum_i w_i x_i + b$ ，得到最终结果



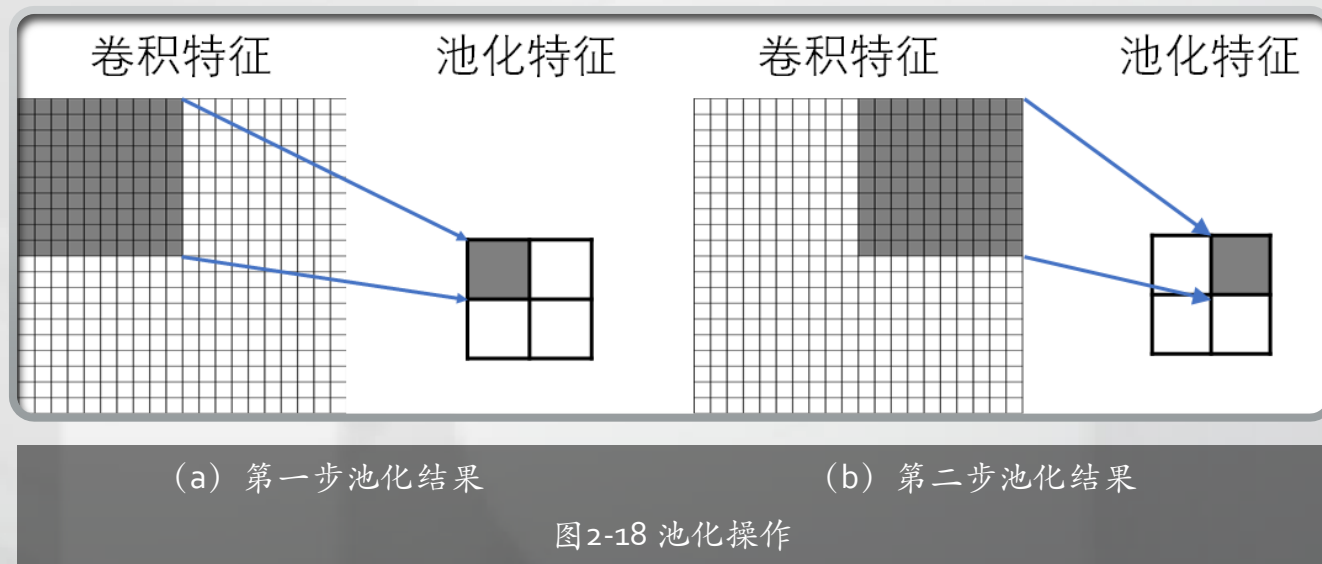
(a) RGB图像 (b) 卷积模板 (c) 卷积结果

图2-17 RGB彩色图像卷积的计算过程

2.4 深度学习基础知识

2. 池化

- 目的：降低特征空间的维度
- 类别：最大池化、平均池化等
- 特点：每步池化操作之间取的图像区域之间没有交集
- 例子：最大池化
 - ✓ 2-18 (a) 中的深灰色区域取最大值得到的值将放到 4×4 结果矩阵中的 $[0,0]$ 的位置
 - ✓ 图2-18 (b) 也是类似的处理过程



计算机视觉的主要分支有：

图像分类

- 通用图像分类（如 ImageNet）
- 手写数字识别（如 Minist）
- 人脸识别
- 场景识别
- 路标分类等

图像分割

- 图像编辑
- 图像修复
- Photoshop等

物体检测

- 行人检测
- 车辆检测
- 车牌检测等

本门课程第3、4、5章分别介绍了这三部分的内容