计算机视觉基础

第四章 图像语义分割

- 4.1 概述
- 4.2 基于聚类的分割方法
- 4.3 基于边缘的分割方法
- 4.4 基于区域的分割方法
- 4.5 基于图论的分割方法
- 4.6 基于深度学习的分割方法

4.1 概 述

图像分割是计算机视觉中的基本任务，目的是根据图像所显示的内容标记特定区域。

所谓图像分割指的是根据灰度、颜色、纹理和形状等特征把图像划分成若干互不交迭的区域，并使这些特征在同一区域内呈现出相似性，而在不同区域间呈现出明显的差异性。

简单来说分割就是为了解决"这张图片里有什么，它在图片什么位置"的问题。

4.1 概述

给分割后图像中的每一类物体添加语义标签，用不同的颜色代表不同类别的物体，即形成了图像的语义分割。

➢ 实例：

- 左图为原始图像
- 右图为分割任务的真实标记（ground truth）：1号区域表示语义为“person”的区域，2号区域表示语义为“motorbike”的区域，3号区域表示“background”，白色（边）则表示未标记区域。

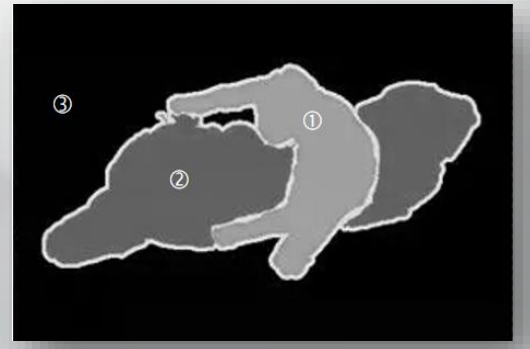


图 4-1 图像语义分割实例

4.1 概 述

图像语义分割也称为“图像语义标注”（image semantic labeling）、“像素语义标注”（semantic pixel labeling）或“像素语义分组”（semantic pixel grouping）。

由于图像语义分割的目标是用相应的类别来表示图像的每个像素，所以通常也称为密集预测（dense prediction）。

4.1 概述

目前语义分割主要包含以下应用领域：

1. 地理信息系统：可以通过训练神经网络让机器输入卫星遥感影像，自动识别道路，河流，庄稼，建筑物等，并且对图像中每个像素进行标注。

2. 无人车驾驶：语义分割也是无人车驾驶的核心算法技术，车载摄像头，或者激光雷达探查到图像后输入到神经网络中，后台计算机可以自动将图像分割归类，以避让行人和车辆等障碍。

3. 医疗影像分析：随着人工智能的崛起，将神经网络与医疗诊断结合也成为研究热点，智能医疗研究逐渐成熟。在智能医疗领域，语义分割主要应用于肿瘤图像分割、龋齿诊断等。

4.2 基于聚类的分割方法

聚类即确定图像中的像素的自然归属类别。

聚类分割技术按照样本间的相似性把集合划分为若干子集，划分结果使某种表示聚类质量的准则为最大。

当用距离来表示两个样本间的相似度时，结果就是把特征空间划分为若干区域，每一个区域相当于一个类别。

4.2 基于聚类的分割方法

一般来说，基于聚类的分割方法通用的步骤如下：

1. 初始化一个粗糙的聚类
2. 使用迭代的方式将颜色、亮度、纹理等特征相似的像素点聚类到同一超像素，迭代直至收敛，从而得到最终的图像分割结果。

基于像素聚类的代表方法有：

- K-Means（K均值）
- 谱聚类
- Meanshift
- SLIC等。

4.2 基于聚类的分割方法

4.2.1 K-Means

■ K-Means是最简单的聚类算法之一

- K-Means聚类算法的实现步骤大致表示如下：

1. 随机选取K个初始聚类中心；
2. 计算每个样本到各聚类中心的距离，将每个样本归到其距离最近的聚类中心所对应的簇中；
3. 对每个簇，以所有样本的均值作为该簇新的聚类中心；
4. 重复第2-3步，直到聚类中心不再变化或达到设定的迭代次数。

4.2 基于聚类的分割方法

4.2.1 K-Means

■ K-Means聚类的优点

- 可收敛性。每次迭代都朝着全局最优靠近。

□K-Means聚类的缺点

1. 聚类簇数K没有明确的选取准则，但是在实际应用中K一般不会设置很大。
2. 从K-Means算法框架可以看出，该算法的每一次迭代都要遍历所有样本，计算每个样本到所有聚类中心的距离，因此当样本规模非常大时，算法的时间开销是非常大的。
3. K-Means算法是基于距离的划分方法，只适用于分布为凸形的数据集，不适合对于图4-2所示的聚类非凸形状的类簇。

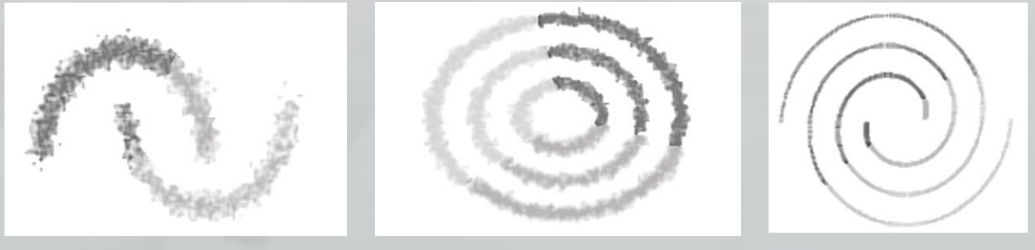


图4-2 非凸形状数据分布

4.2 基于聚类的分割方法

4.2.2 谱聚类

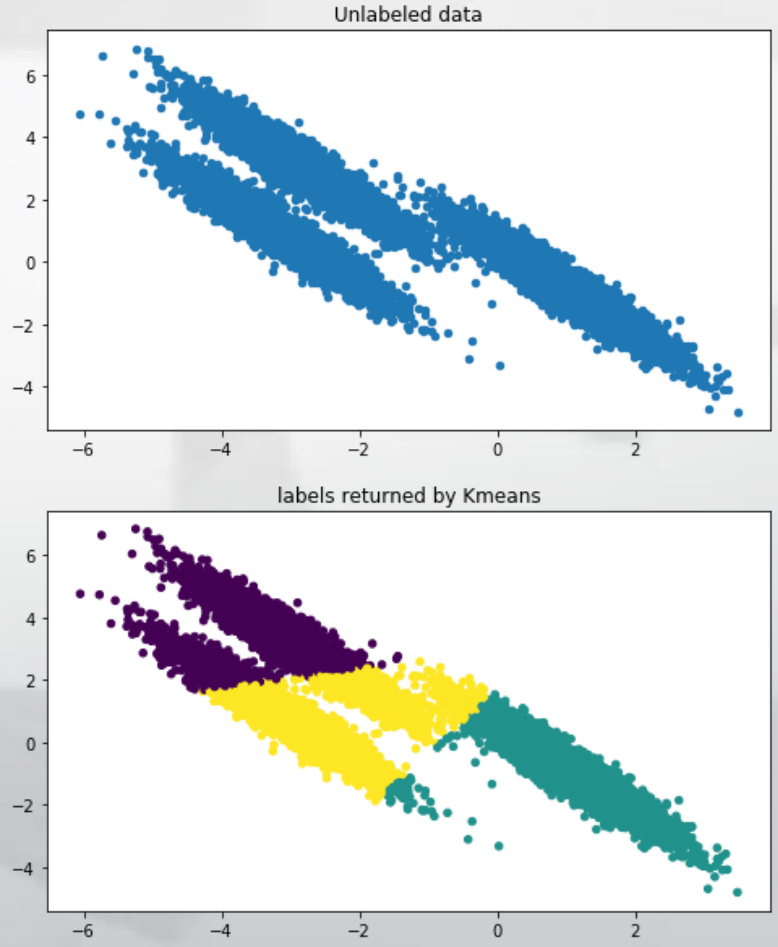
谱聚类(Spectral Clustering)是一种基于图论的聚类方法——将带权无向图划分为两个或两个以上的最优子图，使子图内部尽量相似，而子图间距离尽量较远，以达到常见的聚类的目的。

- 与传统的聚类算法相比，该算法能在任意形状的样本空间上执行并且收敛于全局最优，这个特点使得它对数据的适应性非常广泛。
- 与 K-Means 算法相比谱聚类方法能够对高维度、非常规分布的数据进行聚类。

4.2 基于聚类的分割方法

4.2.2 谱聚类

- Kmeans聚类的问题

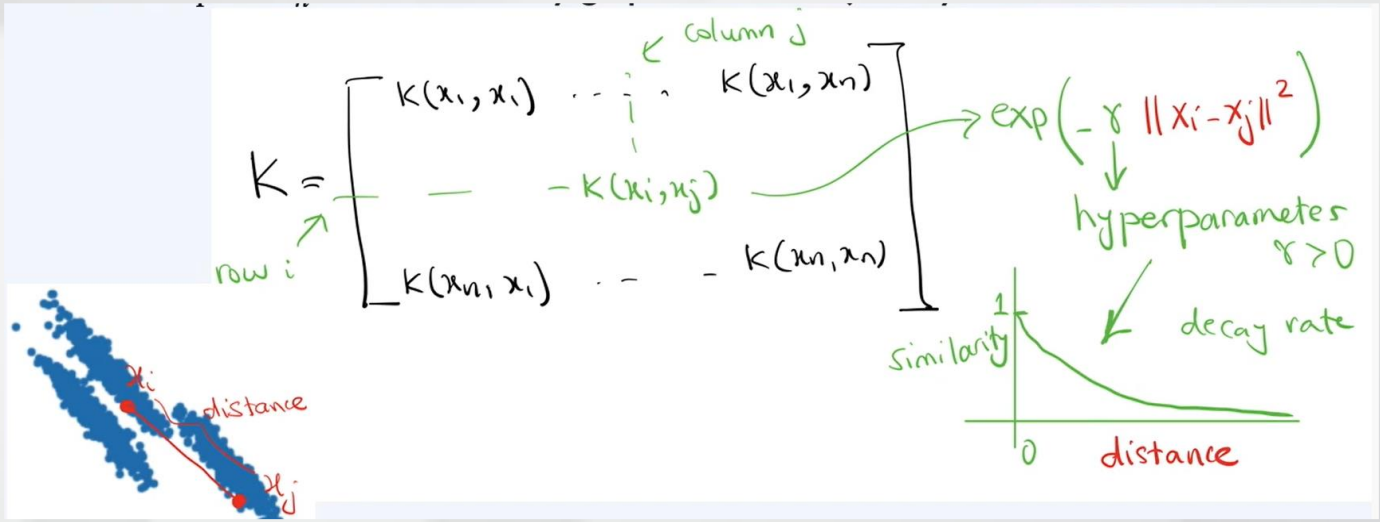


4.2 基于聚类的分割方法

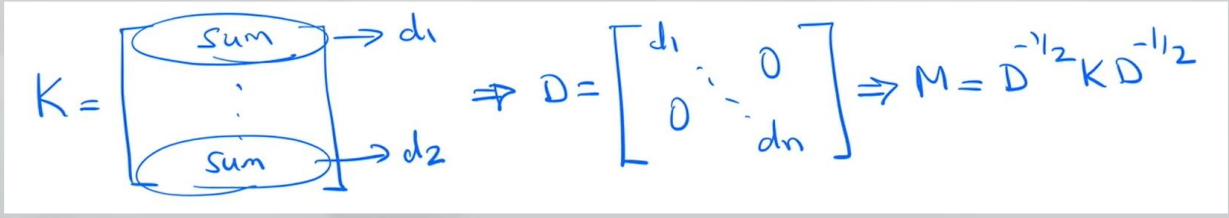
4.2.2 谱聚类

1. 计算相似度矩阵

- 相似度矩阵：



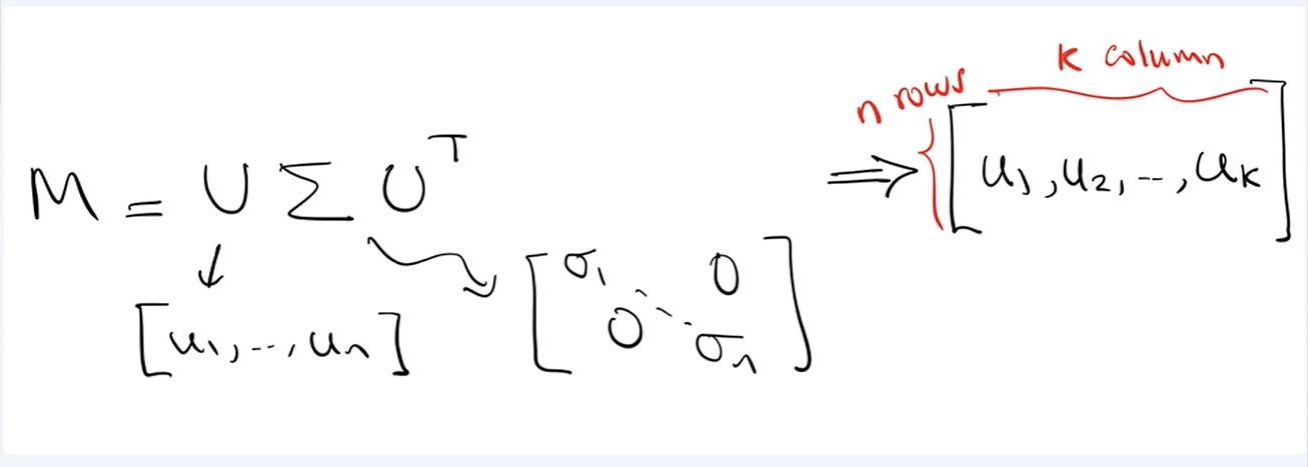
- 相似度矩阵标准化



4.2 基于聚类的分割方法

4.2.2 谱聚类

2. 计算特征值

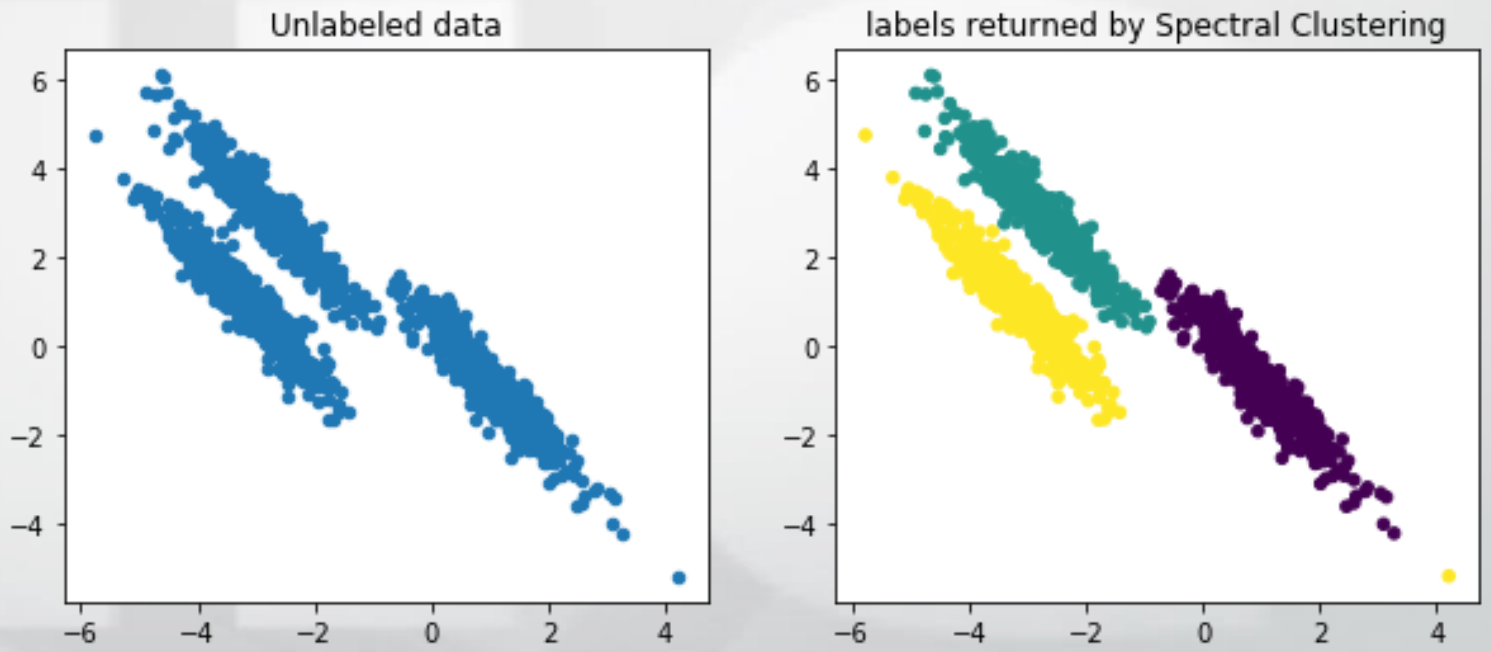


3. 采用Kmeans进行聚类

4.2 基于聚类的分割方法

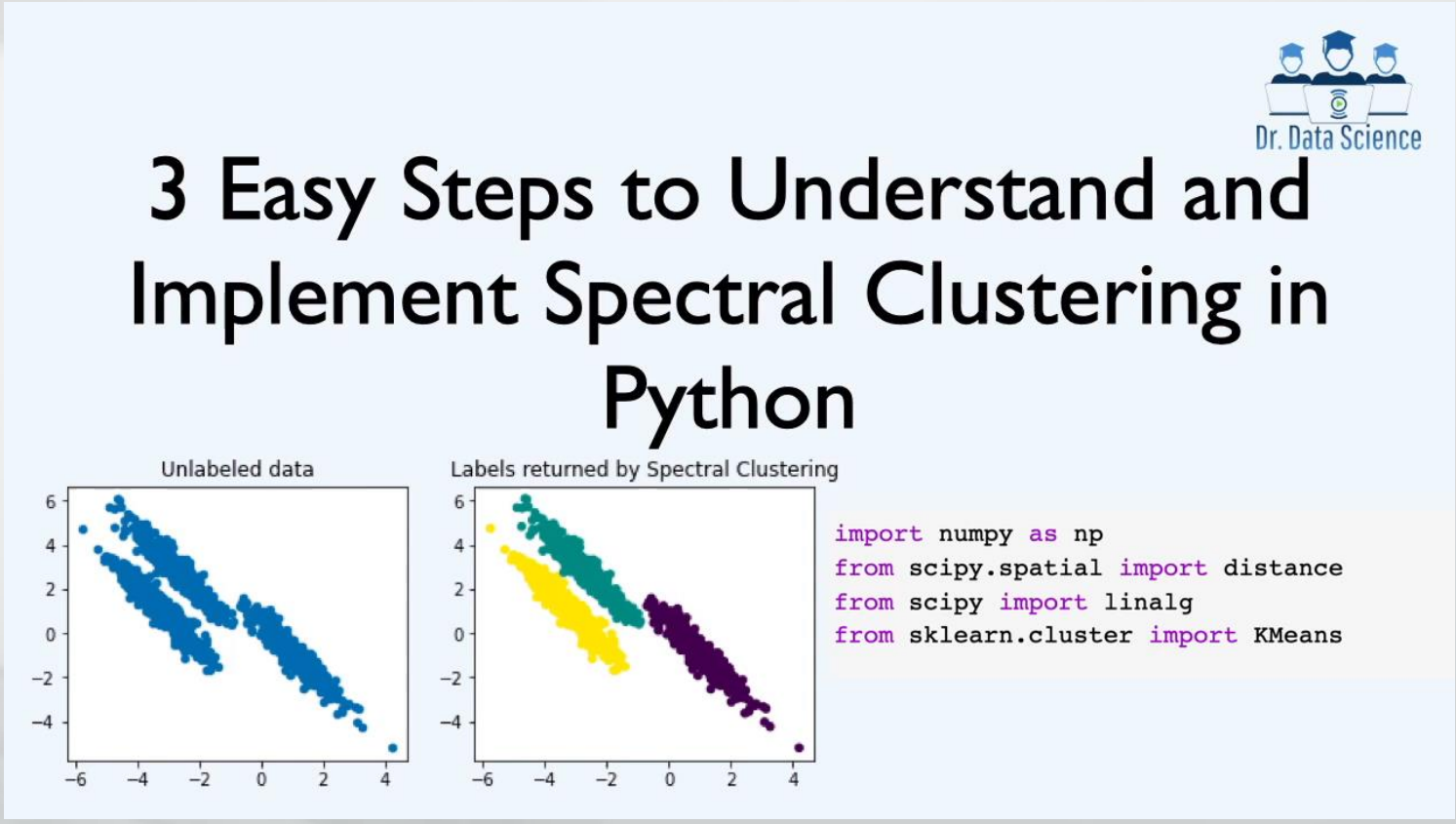
4.2.2 谱聚类

- 谱聚类结果



4.2 基于聚类的分割方法

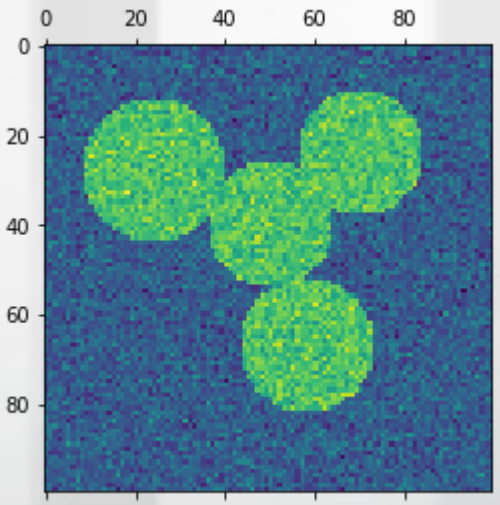
4.2.2 谱聚类

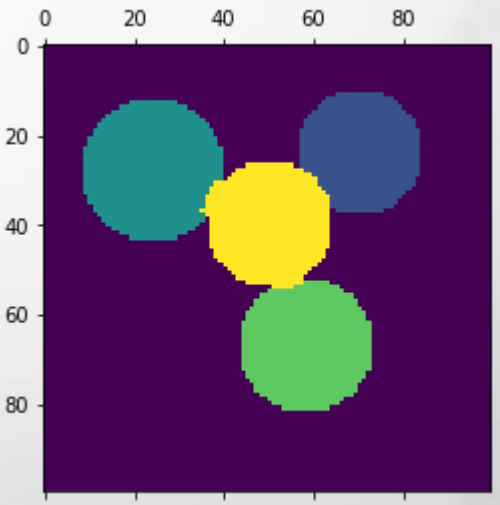


4.2 基于聚类的分割方法

4.2.2 谱聚类

- 基于谱聚类的图像分割





4.2 基于聚类的分割方法

4.2.3 Mean Shift

Mean Shift算法，又称均值漂移算法，是一种基于核密度估计的爬山算法。

可用于聚类、图像分割、跟踪等。

- 它的工作原理基于质心，这意味着它的目标是定位每个簇（类）的质心，即先算出当前点的偏移均值，将该点移动到此偏移均值位置，然后以此为新的起始点，继续移动，直到满足最终的条件（找出最密集的区域）。

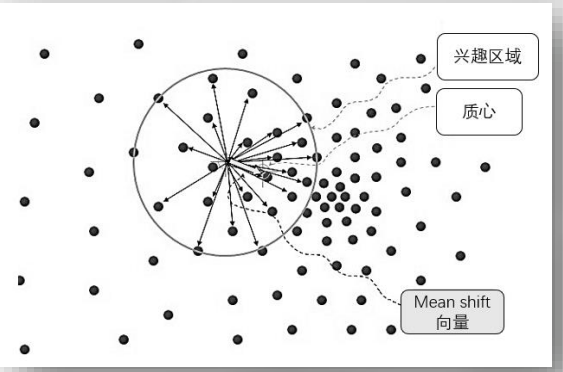
4.2 基于聚类的分割方法

4.2.3 Mean Shift

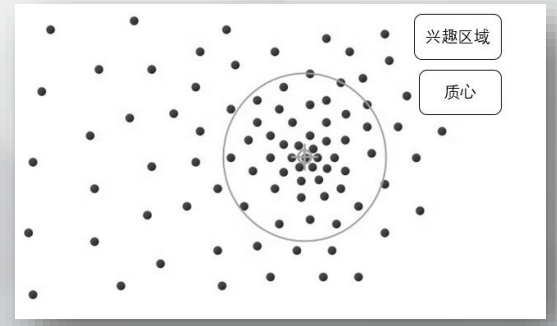
4.2 基于聚类的分割方法

4.2.3 Mean Shift

1. 首先在d 维空间中，任选一点作为圆心，以h为半径做圆。圆心和圆内的每个点都构成一个向量。将这些向量进行矢量加法操作，得到Mean Shift向量(图a)
2. 继续以Mean Shift 向量的终点为圆心做圆，得到下一个Mean Shift 向量，通过有限次迭代计算（图a）
3. 最终Mean Shift 算法一定可以收敛到图中概率密度最大的位置，即数据分布的稳定点，称为模点（图b）



（a）聚类过程



（b）聚类结果

图4-3 Mean Shift算法工作流程。

4.2 基于聚类的分割方法

4.2.3 Mean Shift

- Mean Shift 算法具有较好的稳定性与鲁棒性。
- 与K-Means算法相比，Mean Shift不需要实现定义聚类数量，因为这些都可以在计算偏移均值时得出。
- 同时，算法推动聚类中心在向密度最大区域靠近的效果也非常令人满意，这一过程符合数据驱动型任务的需要，而且十分自然直观。
- 但是Meanshift对于高维球区域选择不同半径r可能会产生高度不同的影响。
- 另外，当分割对象所包含的语义信息较少时使用本算法分割效果不理想。
- 且算法运行速度较慢，不适用于实时处理任务。

4.2 基于聚类的分割方法

4.2.4 SLIC

SLIC（Simple Linear Iterative Clustering），是一种思想简单、实现方便的算法，将彩色图像转化为CIE Lab颜色空间和XY坐标下的5维特征向量，然后对5维特征向量构造距离度量标准，对图像像素进行局部聚类的过程。

- SLIC算法能生成紧凑、近似均匀的超像素，在运算速度、物体轮廓保持、超像素形状方面具有较高的综合评价，比较符合人们期望的分割效果。
- 在平常图像处理任务中，处理的最小单位是像素，这就是像素级（pixel-level）；而把像素级的图像划分成为区域级（district-level）的图像，把区域当成是最基本的处理单元，这就是超像素。

4.2 基于聚类的分割方法

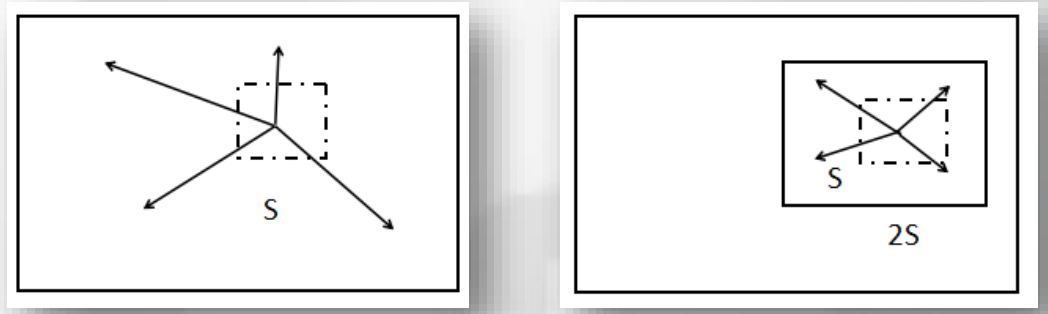
4.2.4 SLIC

4.2 基于聚类的分割方法

4.2.4 SLIC

- SLIC的具体实现包含以下步骤：

1. 将图像转换为CIE Lab颜色空间，对应每个像素的（L, a, b）颜色值和（x, y）坐标组成一个5维向量V[L, a, b, x, y]。
2. 初始化K个种子点（聚类中心），在图像上平均撒落K个点，K个点均匀的占满整幅图像。假设图片总共有 N 个像素点，预分割为K个相同尺寸的超像素，那么每个超像素的大小为N/K，则相邻种子点的距离（步长）近似为S=sqrt(N/K)。
3. 对种子点在内的n×n（一般为3×3）区域计算每个像素点梯度值，为了防止种子点落在了轮廓边界上，选择值最小（最平滑）的点作为新的种子点。
4. 对种子点周围 2S×2S的方形区域内的所有像素点计算距离度量D′，对于K-Means算法是计算整张图的所有像素点，而SLIC的计算范围是2S×2S，所以SLIC算法收敛速度更快。
5. 每个像素点都可能被几个种子点计算距离度量，选择其中最小的距离度量对应的种子点作为其聚类中心。



（a）K-means在整个图片空间进行搜索

（b）SLIC在某个局部空间进行搜索聚类

图4-4 K-means与SLIC方法对比

4.2 基于聚类的分割方法

4.2.4 SLIC

- SLIC算法主要有优点有：
  1. 生成的超像素一般比较紧凑整齐且邻域特征明显；
  2. 可对彩色图和灰度图进行分割；
  3. 只需设置一个预分割的超像素参数；
  4. 与其他超像素分割方法比较，运算速度、紧凑整齐度优于其它算法。
- SLIC算法的主要缺点是：
  - 对边缘的保持使用位置限制，导致超像素和图像边缘的契合度变差。

4.3 基于边缘的分割方法

基于边缘检测的图像分割算法试图通过检测包含不同区域的边缘来解决分割问题。

- 通常不同区域的边界上像素的灰度值变化比较剧烈，如果将图片从空间域通过傅里叶变换到频率域，边缘就对应着高频部分，这是一种非常简单的边缘检测算法。

边缘检测技术通常可以按照处理的技术分为：

- 串行边缘检测。串行边缘检测是要想确定当前像素点是否属于检测边缘上的一点，取决于先前像素的验证结果
- 和并行边缘检测。并行边缘检测是确定一个像素点是否属于检测边缘上的一点，取决于当前正在检测的像素点以及与该像素点临近的一些临近像素点。

4.3 基于边缘的分割方法

基于边缘分割的一般流程如下：

1. 确定起始边界点；
2. 选择搜索策略，并根据一定的机理依次检测新的边界点；
3. 设定终止条件，当搜索进程结束时使之停下来。

常用的一阶微分算子有：

- Roberts
- Sobel
- Prewitt
- Canny等算子

常用的二阶微分算子有：

- Laplace
- Kirsh
- LOG等算子

4.3 基于边缘的分割方法

微分算子

- 常用的一阶算子：
  - Roberts算子 $\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}\begin{bmatrix}0 & 1\\-1 & 0\end{bmatrix}$
  - Sobel算子 $\begin{bmatrix}-1 & -2 & -1\\0 & 0 & 0\\1 & 2 & 1\end{bmatrix}\begin{bmatrix}-1 & 0 & 1\\-2 & 0 & 2\\-1 & 0 & 1\end{bmatrix}$
  - Prewitt算子 $\begin{bmatrix}-1 & -1 & -1\\0 & 0 & 0\\1 & 1 & 1\end{bmatrix}\begin{bmatrix}-1 & 0 & 1\\-1 & 0 & 1\\-1 & 0 & 1\end{bmatrix}$
  - Canny算子
- 常用的二阶微分算子：
  - Laplace算子
  - Kirsh算子
  - LOG算子 $\begin{Vmatrix}0 & -1 & 0\\-1 & 4 & -1\\0 & -1 & 0\end{Vmatrix}$ 或 $\begin{Vmatrix}-1 & -1 & -1\\-1 & 8 & -1\\-1 & -1 & -1\end{Vmatrix}$

（a）梯度算法　（b）Roberts算法

（c）Sobel算法　（d）Prewitt算法

图4-5 不同算法的处理结果比较

4.4 基于区域的分割方法

基于区域的分割方法是将图像按照相似性准则分成不同的区域，主要包括：

1. 基于阈值的分割方法
2. 种子区域生长法
3. 区域分裂合并法
4. 分水岭法等

4.4 基于区域的分割方法

4.4.1 基于阈值的分割方法

- 阈值法可以解决将图像分为背景与目标两部分的分割问题。
- 阈值法的基本思想是基于图像的灰度特征来计算一个或多个灰度阈值，并将图像中每个像素的灰度值与阈值作比较，最后将像素根据比较结果分到合适的类别中。

✓该方法最为关键的一步就是按照某个准则函数求解最佳灰度阈值。

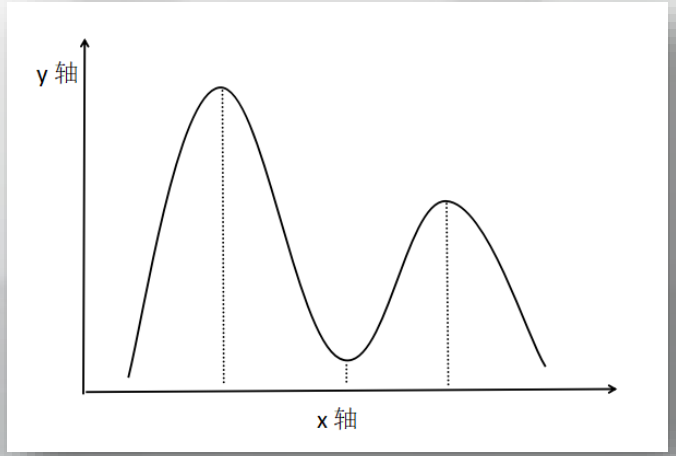


图4-6 以谷底为阈值分割

4.4 基于区域的分割方法

4.4.1 基于阈值的经典分割算法：

1. 固定阈值分割：思路简单，即固定某像素值为分割点对图像进行分割。

2. 直方图双峰法：假设图像中有明显的目标和背景，则其灰度直方图呈双峰分布，当灰度级直方图具有双峰特性时，选取两峰之间波谷对应的灰度级作为阈值。

3. 迭代阈值图像分割：该算法先假定一个阈值，然后计算在该阈值下的前景和背景的中心值，当前景和背景中心值的平均值和假定的阈值相同时，则迭代中止，并以此值为阈值进行二值化。

4. 自适应阈值图像分割：有时候物体和背景的对比度在图像中不是处处一样的，普通阈值分割难以起作用。这时候可以根据图像的局部特征分别采用不同的阈值进行分割。

5. 大津法 OTSU（最大类间方差法）：背景和目标之间的类间方差越大，说明构成图像的两部分的差别越大，当部分目标错分为背景或部分背景错分为目标都会导致两部分差别变小。因此，使类间方差最大的分割意味着错分概率最小。

4.4 基于区域的分割方法

4.4.1 基于阈值的经典分割算法：

6. 均值法：把图像分成m×n块子图，求取每一块子图的灰度均值就是所有像素灰度值之和除以像素点的数量，这个均值就是阈值。

7. 最佳阈值：阈值选择需要根据具体问题来确定，一般通过实验来确定。如对某类图片，可以分析其直方图等。

阀值分割方法计算简单，效率较高，

但由于只考虑像素点灰度值本身的特征，一般不考虑空间特征，因此对噪声比较敏感，鲁棒性不高。

4.4 基于区域的分割方法

4.4.2 种子区域生长法

- 基本思想：将具有相似性质的像素集合起来构成区域，即从一组代表不同生长区域的种子像素开始，将种子像素邻域里符合条件的像素合并到种子像素所代表的生长区域中，并将新添加的像素作为新的种子像素继续合并，直到找不到符合条件的新像素为止。

- 该方法的关键是选择合适的初始种子像素以及合理的生长准则。

- 区域生长算法需要解决的三个问题：

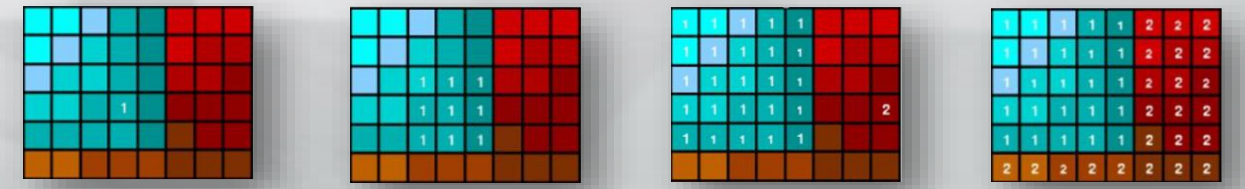
  1. 选择或确定一组能正确代表所需区域的种子像素；
  2. 确定在生长过程中能将相邻像素包括进来的准则；
  3. 指定让生长过程停止的条件或规则。

4.4 基于区域的分割方法

4.4.2 种子区域生长法

- 方法过程：

1. 随机选取图像中的一个像素作为种子像素，并将其表示出来，如：标签1。
2. 检索种子附近的未被标记的像素点，如果差值在所规定的阈值内（阈值需要根据不同的情况进行设置），则合并到分割区域中。
3. 重复步骤2，直到区域停止扩张，并在此时再次随机选择非选定区域的一个像素做为种子像素。
4. 重复上述步骤，直到图像中的每个像素均被分配到不同的区域中。

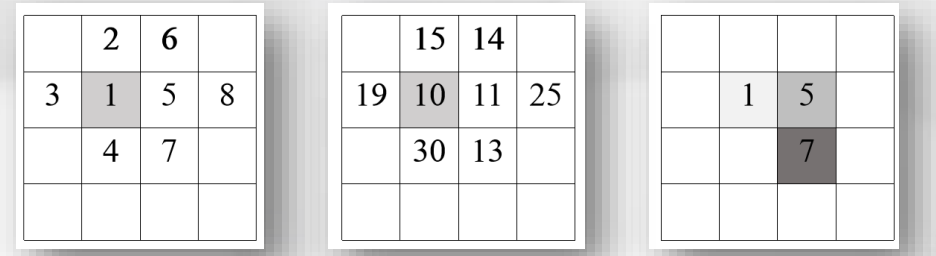


步骤1　步骤2　步骤3　步骤4

4.4 基于区域的分割方法

4.4.2 种子区域生长法

➢ 示例：



(a) 像素点标号　(b) 像素点灰度值　(c) 区域生长方向

图4-7 区域生生长法过程示意图

种子区域生长法的优点是计算简单，对于较均匀的连通目标有较好的分割效果；

缺点是需要人为确定种子点，对噪声敏感，可能导致区域内有空洞。另外，它是一种串行算法，当目标较大时，分割速度较慢，因此在设计算法时，要尽量提高效率。

4.4 基于区域的分割方法

4.4.3 区域分裂合并法

- 基本思想：首先将图像任意分成若干互不相交的区域，然后再按照相关准则对这些区域进行分裂或者合并从而完成分割任务。

✓是区域生长逆过程

✓该方法既适用于灰度图像分割也适用于纹理图像分割。

4.4 基于区域的分割方法

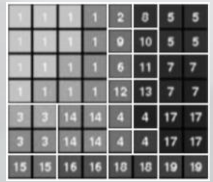
4.4.3 区域分裂合并法

- 方法过程：

1. 首先将图像中的所有像素进行单独的标识；
2. 对于相同标识的区域，如果像素特征值不在同一个范围内，则将该区域分成四个象限；
3. 重复步骤2，直到所有的子区域都有相似的特征值；
4. 对比相邻的子区域，将相似的区域进行合并，重复上述过程，直到所有子区域均没有相邻的相似区域。

步骤1

步骤2



步骤3

步骤4

4.4 基于区域的分割方法

4.4.3 区域分裂合并法

区域分裂合并算法对于复杂图像具有良好的分割效果，

但是算法复杂，计算量大，分裂过程中有可能破坏区域的边界。

在实际应用当中通常将区域生长算法和区域分裂合并算法结合使用，该类算法对某些复杂物体定义的复杂场景的分割或者对某些自然景物的分割等类似先验知识不足的图像分割效果较为理想。

4.4 基于区域的分割方法

4.4.4 分水岭法

- 分水岭法是一种基于拓扑理论的数学形态学的分割方法
- 基本思想：把图像看作是测地学上的拓扑地貌，图像中每一点像素的灰度值表示该点的海拔高度，每一个局部极小值及其影响区域称为集水盆，而集水盆的边界则形成分水岭。
- 算法模拟：

✓可以模拟成洪水淹没的过程，图像的最低点首先被淹没，然后水逐渐淹没整个山谷。当水位到达一定高度的时候将会溢出，这时在水溢出的地方修建堤坝，重复这个过程直到整个图像上的点全部被淹没，这时所建立的一系列堤坝就成为分开各个盆地的分水岭。

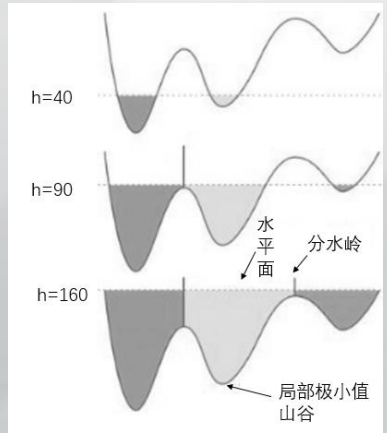


图4-8 分水岭模型

4.5 基于图论的分割方法

目前使用较多的基于图论的方法有：

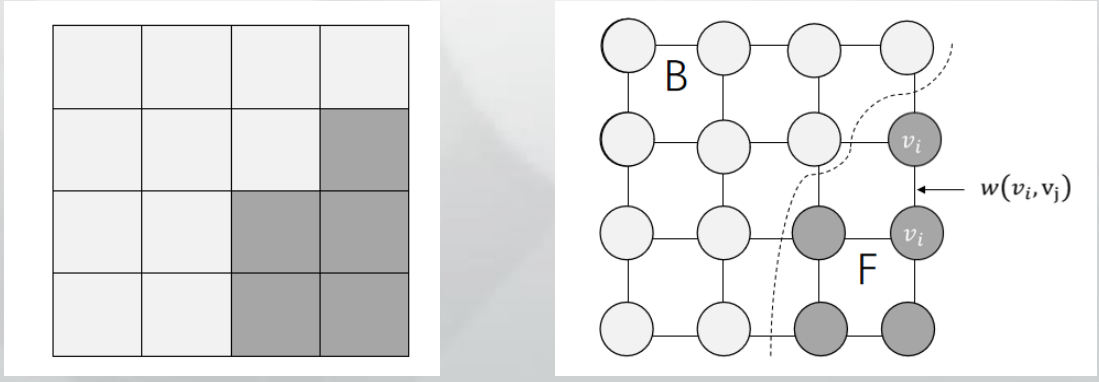
1. Normalized Cut
2. Graph Cuts
3. Grab Cut等

图4-9 图与基于图论的分割方法的映射关系

4.5 基于图论的分割方法

4.5.1 Normalized Cut

- 最小化分割（min-cut)：图G分成两个子集A、B，$A \cup B = V$，$A \cap B = \emptyset$，$cut(A,B) = \sum \mu \in A, \nu \in B \omega(\mu, \nu)$，其中（是权重(weight)。现在需要把它分成两个部分，最小割就是让上式的值最小，
- 例子：我们把图4-10（a）看成一个整体，中间的黑色虚线切割的边就是最小化分割，如图4-10（b）所示。



(a)原始图像　(b)图像对应的图G=(V,E)

图4-10 从图像计算图的过程

4.5 基于图论的分割方法

4.5.1 Normalized Cut

- 最小化分割的问题：因为min-cut目的是使cut(A，B)的值最小，而边缘处cut值确实是最小，因此最小化切割时会有偏差。
- Normalized Cut的设计思路是对分割进行标准化处理：

$$Normalize\quad Cut = \frac{cut(A,B)}{volA} + \frac{cut(A,B)}{volB}$$

- 效果对比：

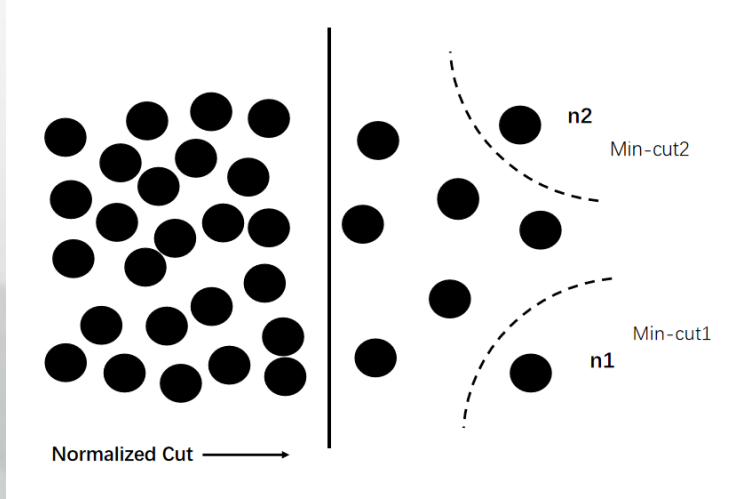


图4-11 Normalized Cut与 Min-Cut效果对比

4.5 基于图论的分割方法

4.5.2 Graph Cuts

- Graph Cuts是一种能量优化算法
- 在计算机视觉领域普遍应用于前背景分割（Image segmentation）、立体视觉（Stereo vision）、抠图（Imagematting）等

4.5 基于图论的分割方法

4.5.2 Graph Cuts

- 方法过程：

1. 构造网络图。首先用一个无向图G=<V，E>表示要分割的图像，V和E分别是顶点和边的集合。而Graph Cuts图是在普通图的基础上多了2个终端顶点S和T，其它所有的顶点都必须和这2个顶点相连形成边集合中的一部分。如图4-12，Graph Cuts中存在两种顶点和边：

① 普通顶点对应于图像中的每个像素。每两个邻域顶点（对应于图像中每两个邻域像素）的连接就是一条边。这种边也叫n-links。

② 除图像像素外的两个终端顶点S和T，每个普通顶点和这2个终端顶点之间都有连接，组成第二种边。这种边也叫t-links。

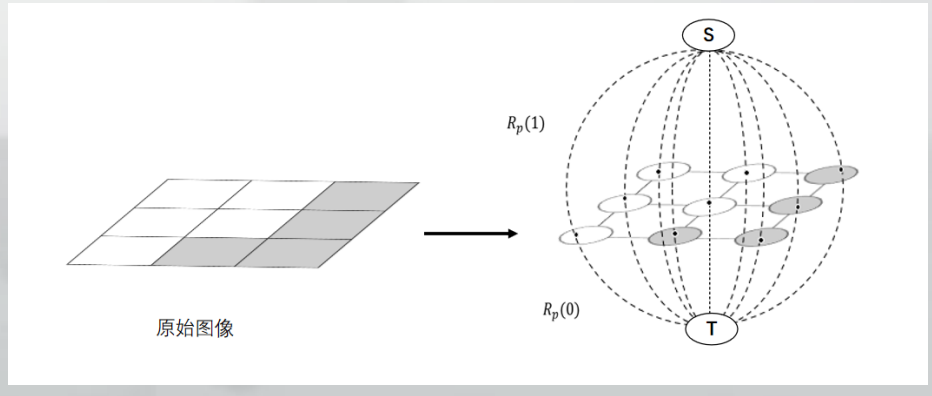


图4-12 图模型建立

4.5 基于图论的分割方法

4.5.2 Graph Cuts

- 方法过程：

1. 构造网络图。
2. 构造能量函数。

用每一个割对应一个能量函数的值，当割处在前景和背景之间时，能量函数达到最优值。能量函数由区域项和边界项构成：

$$E(A) = \lambda \cdot R(A) + B(A)$$

式中R(A)是先验惩罚项，B(A)是区域相似度惩罚项，是平衡因子。

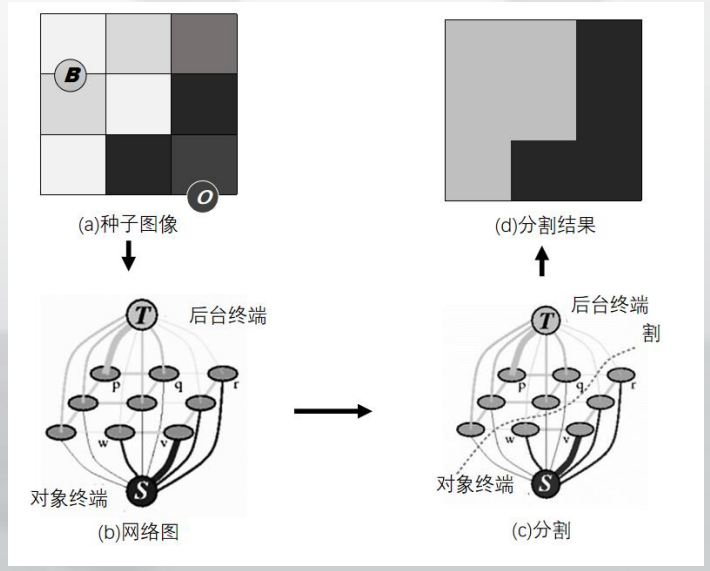


图4-13 Graph Cuts分割过程

4.5 基于图论的分割方法

4.5.2 Graph Cuts

- Graph Cuts的缺陷包括：
  - 算法只适用于灰度图；
  - 需要人工标注至少一个前景点和一个背景点；
  - 分割结果为硬分割，未考虑边缘介于0到1之间的透明度。

4.6 基于深度学习的分割方法

基于深度学习的分割方法有：

1. 基于上采样/反卷积的分割方法
2. 基于提高特征分辨率的分割方法
3. 基于RNN的图像分割
4. 基于特征增强的分割方法
5. 使用CRF/MRF的方法

4.6.1基于上采样/反卷积的分割方法

全卷积神经网络

- 与经典的卷积神经网络在卷积层之后使用全连接层得到固定长度的特征向量进行分类不同，全卷积神经网络采用反卷积层对最后一个卷积层的特征图进行上采样，使它恢复到输入图像相同的尺寸，从而可以对每个像素都产生了一个预测，同时保留了原始输入图像中的空间信息。
- 最后在上采样的特征图上进行逐像素分类，从而实现了语义级别的图像分割。

4.6.1基于上采样/反卷积的分割方法

全卷积神经网络

- 全卷积神经网络的主要操作包含:

1. 全卷积层

- 全卷积神经网络将传统CNN中的全连接层替换成一个个的卷积层。

➢实例（如图4-16所示）：

➢在使用的CNN模型中，前5层是卷积层，第6层和第7层分别是一个长度为4096的一维向量，第8层是长度为1000的一维向量，分别对应1000个类别的概率。

➢卷积神经网络把3个全连接层用卷积层实现，卷积核的大小(通道数，宽，高)分别为（4096，1，1）、（4096，1，1）、（1000，1，1）。所有的层都是卷积层，故称为全卷积网络。

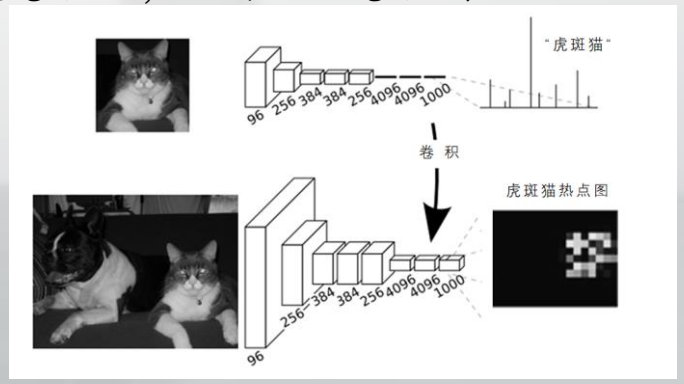


图4-16 传统的卷积神经网络与全卷积神经网络的对比

[1] Long J, Shelhamer E, Darrell T.Proceedings of the IEEE conference on computer vision and pattern recognition, June 8-10,2015[C].Piscataway, NJ: IEEE，2015.

4.6.1基于上采样/反卷积的分割方法

全卷积神经网络

- 全卷积神经网络的主要操作包含:

1. 全卷积层
2. 上采样层

➢ 实例（如图4-17所示）：

➢ 在池化过程中，下采样会使图片不断缩小，例如经过5次卷积和池化以后，图像的分辨率依次缩小了2，4，8，16，32倍，这使得图片中的像素点不能恢复到原图，给像素级别的训练带来困扰，因此需要对特征图进行上采样。

➢ 对于最后一层的输出图像，需要进行32倍的上采样才能得到原图一样的大小，而FCN采用的上采样的方法就是反卷积。

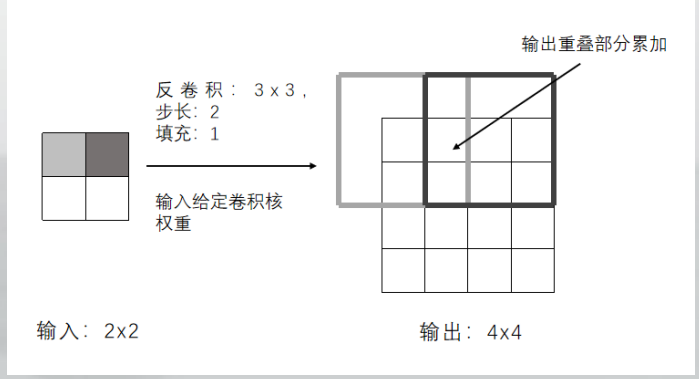


图4-17 反卷积操作示意图

4.6.1基于上采样/反卷积的分割方法

全卷积神经网络

- 全卷积神经网络的主要操作包含:

1. 全卷积层
2. 上采样层
3. 跳跃层

- 跳跃层将不同池化层的结果进行上采样之后来优化输出

➢ 实例（如图4-18所示）：

➢ 对第5层的输出（32倍放大）反卷积到原图大小，但得到的结果还是不够精确，一些细节无法恢复，于是作者将第4层的输出和第3层的输出也依次反卷积，分别进行16倍和8倍上采样，将不同全局步长下的层之间进行连接，结果也就更精细一些。

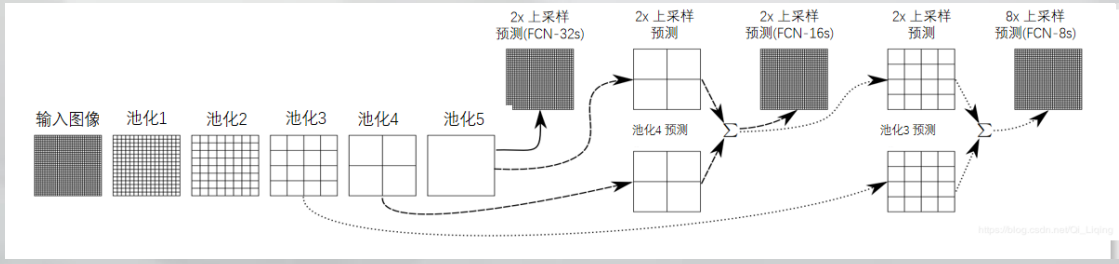


图4-18 FCN全卷积神经网络中的跳跃连接

4.6.2基于提高特征分辨率的分割方法

DeepLab

- DeepLab结合了深度卷积神经网络和概率图模型

✓应用在语义分割的任务上，目的是做逐像素分类

- 之前的语义分割网络结果往往比较粗糙，原因主要有两个：
  - 池化导致丢失信息
  - 没有利用标签之间的概率关系
- 全卷积神经网络先减小图片尺寸（卷积）再增大尺寸（上采样）的过程会造成一定信息损失，DeepLab的提出旨在解决这类问题。
- 其先进性体现在DenseCRFs（概率图模型）和深度卷积神经网络的结合。
  - 将每个像素视为条件随机场的结点，利用远程依赖关系并使用条件随机场推理直接优化深度卷积神经网络的损失函数。

4.6.2基于提高特征分辨率的分割方法

4.6.2.1 DeepLab-V1

■ DeepLab-v1[1]的一大亮点就是使用了空洞卷积

✓ 用带有空洞的卷积核进行采样

✓ 使用空洞卷积扩大了卷积核感受野，使每个卷积输出都包含了较大范围的信息，避免了深度卷积神经网络中重复最大池化和下采样带来的分辨率下降问题。

✓ 通过使用不同采样率的空洞卷积，可以明确控制网络的感受野。

➢ 实例（如图4-22）：

(a) 3x3的1-dilated conv，它和普通的卷积操作是相同的；

(b) 3x3的2-dilated conv，实际卷积核的尺寸还是3x3（黑点），但是空洞为1，其感受野能够达到7x7；

(c) 3x3的4-dilated conv，其感受野已经达到了15x15。

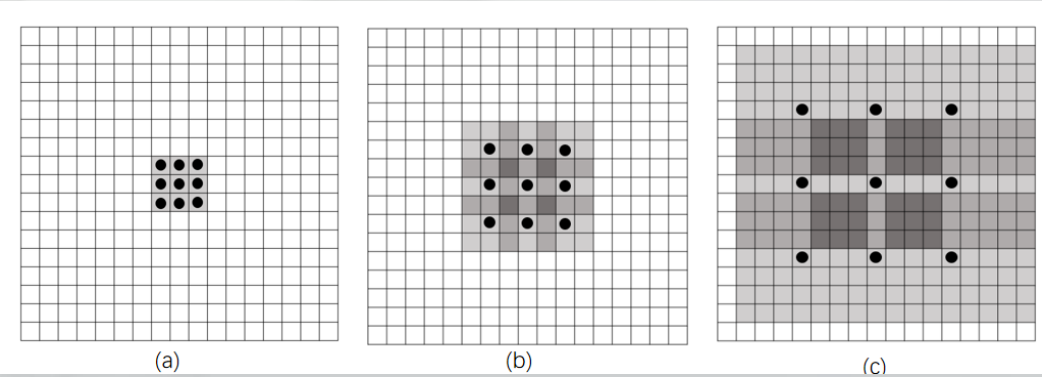


图4-22 不同采样率的空洞卷积示例

[1] Chen L C, Papandreou G, Kokkinos I, et al. Semantic image segmentation with deep convolutional nets and fully connected crfs[J]. arXiv preprint arXiv:1412.7062, 2014

4.6.2基于提高特征分辨率的分割方法

4.6.2.1 DeepLab-V1

- DeepLab-V1基于VGG-16，并作了以下修改：
  - 将VGG-16的全连接层转为卷积
  - 最后的两个最大池化层去掉了下采样
  - 后续卷积层的卷积核改为了空洞卷积。使用空洞卷积，避免池化带来的信息损失。
  - 使用条件随机场，进一步优化分割精度。

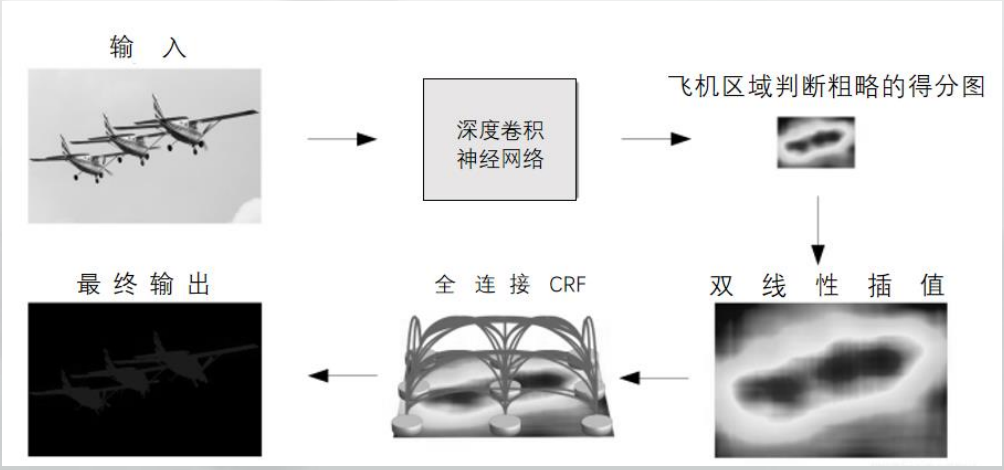


图4-21 DeepLa-V1流程图

4.6.2基于提高特征分辨率的分割方法

4.6.2.2 DeepLab-V2

- DeepLab-V2[2]在DeepLab-v1基础上进行优化:
  - 针对图像中存在多尺度的同一对象问题，模型提出了空洞空间卷积池化金字塔模块（如图），在给定的输入上用多个不同采样率的空洞卷积并行采样，相当于以多个比例捕捉图像的上下文。
- 由于VGG-16表达能力有限，模型将网络框架替换为表达能力更强的ResNet-101，增加了模型的拟合能力。

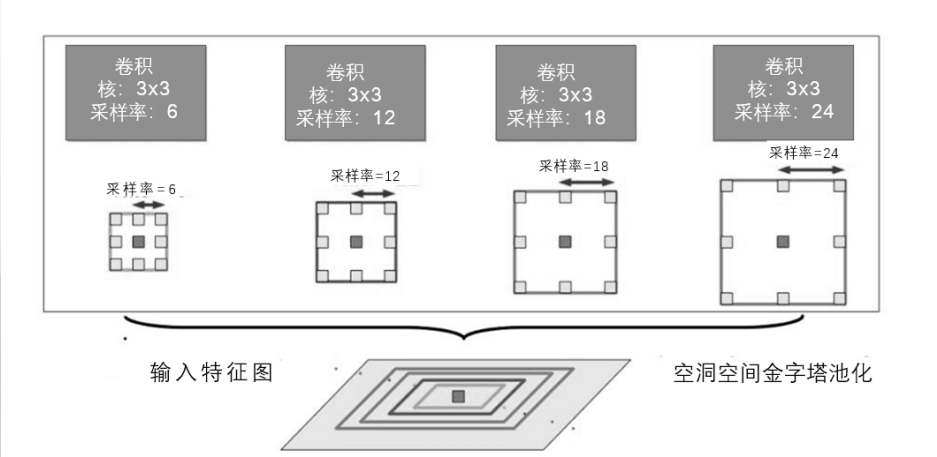


图4-21 DeepLa-V1流程图

[2] Chen L C, Papandreou G, Kokkinos I, et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 40 (4): 834-848

4.6.3基于循环神经网络的图像分割

ReSeg模型

- 尽管基于全卷积神经网络的方法取得了不俗的效果，但分割时仍存在一个重大的不足：没有考虑到局部或者全局的上下文依赖关系，而在语义分割中这种依赖关系是非常有用的。
- ReSeg使用循环神经网络去检索上下文信息，以此作为分割的一部分依据。

4.6.3基于循环神经网络的图像分割

ReSeg模型

- ReSeg是基于图像分割模型ReNet提出的
- ReNet（如图4-28所示）由两层顺序排列的循环神经网络构成。在给定输入图像（或前层）特征后，ReNet对展开结果分别按列、按行扫描。

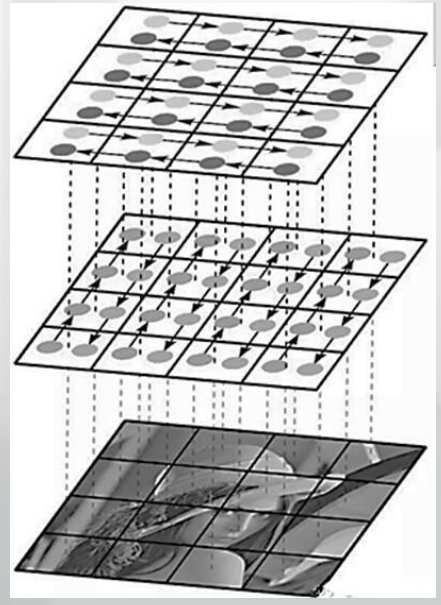


图4-28 ReNet运算示意图

4.6.3基于循环神经网络的图像分割

ReSeg模型

- ReSeg结构（如图4-29）：
  - 3次串联的完整ReNet模块，在这个过程中空间分辨率逐渐减小，目的是将VGG-16提取的特征进行进一步的处理，从而得到对输入图像更复杂的特征描述；
  - 在所有ReNet模块结束后，ReSeg应用了若干层由反卷积组成的上采样层，将特征图的空间分辨率恢复成原始输入图像的空间分辨率；
  - 最后应用softmax实现分割。

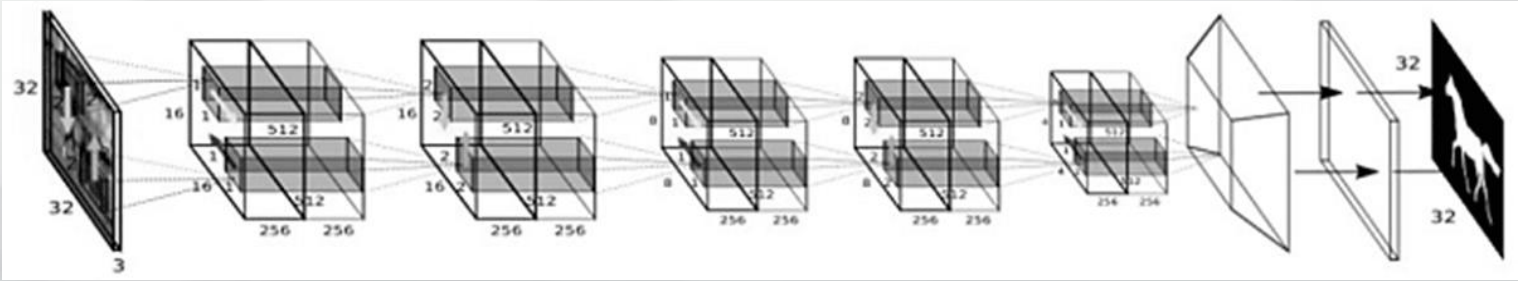


图4-29 ReSeg网络结构