

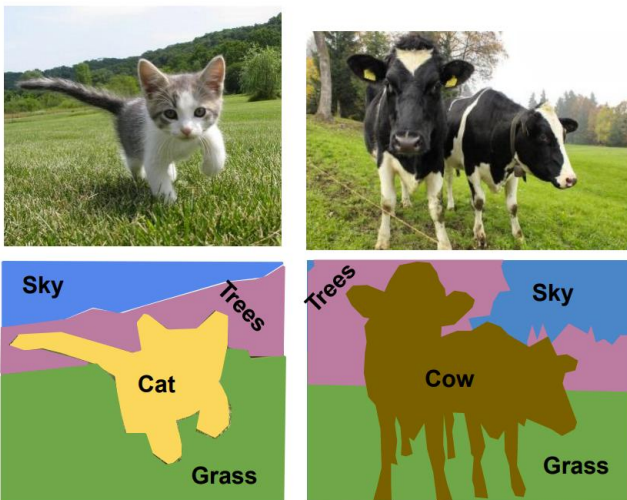
计算机视觉

第6章 语义分割

陈飞: chenfei314@fzu.edu.cn



图像分割任务



像素



特征

边缘



找不同

区域



找相同

边缘 围起来 → 构成区域

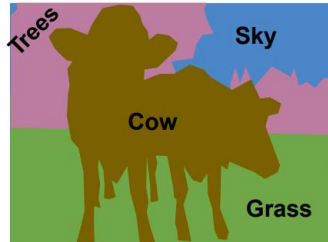
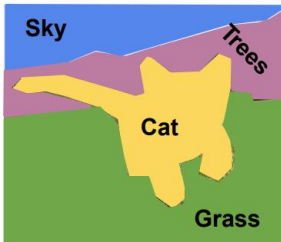
区域 的边界 → 就是边缘

1. 语义分割
2. 实例分割
3. 全景分割

讨论



题目：在图像分割中，模型应该“认识物体”还是“看清边界”？
当前技术更擅长哪一个，哪一个才是真正的瓶颈？



3

本章内容



- 图像分割概述
- 语义分割
- 语义分割思路
- 语义分割发展
- 常用数据集
- 评价标准

4



6.1 图像分割概述

令集合 R 代表整个图像区域，对 R 的分割可看做将 R 分成若干个满足下述条件的非空的子集（子区域）

R_1, R_2, \dots, R_n :

1. $\bigcup_{i=1}^n R_i = R$
2. 对 $\forall i \neq j$, 有 $R_i \cap R_j = \emptyset$
3. 每个子区域 R_i 都是连通的；
4. 对于各个子区域，有均匀性测度度量 P 为真；但对其中任意两个和两个以上相邻子区域之并，其均匀性测度度量 P 为假。即：

$$P(R_i) = \text{TRUE}, \text{ 且 } P(R_i \cup R_j) = \text{FALSE}$$

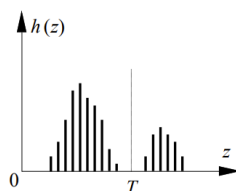
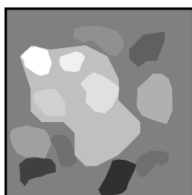
5

单阈值分割图像



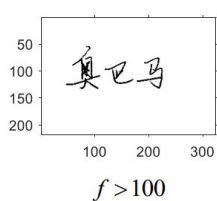
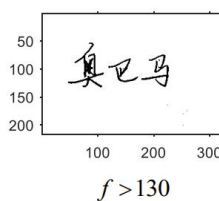
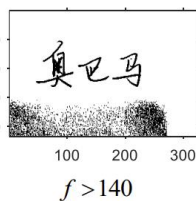
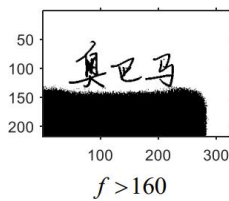
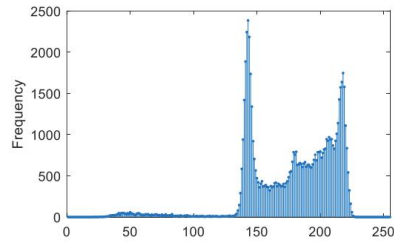
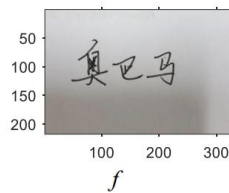
对灰度图（取值在 g_{\min} 和 g_{\max} 之间）确定一个灰度阈值 T
 $(g_{\min} < T < g_{\max})$

$$g(x, y) = \begin{cases} 1 & \text{如 } f(x, y) > T \\ 0 & \text{如 } f(x, y) \leq T \end{cases}$$



6

实例：生成电子签名

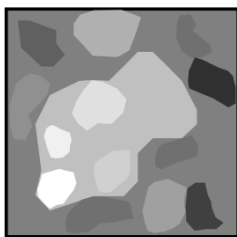


7

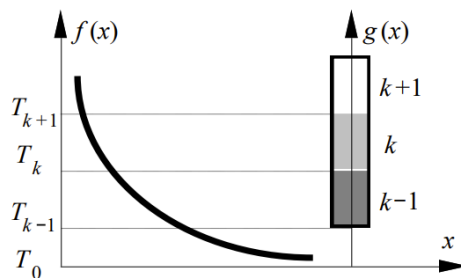
多阈值分割图像



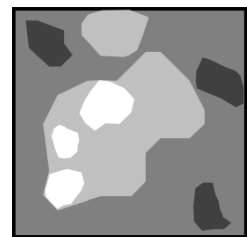
$$g(x, y) = k \quad \text{如} \quad T_k < f(x, y) \leq T_{k+1} \quad k = 0, 1, 2, \dots, K$$



(a)



(b)



(c)

8



最优阈值的选取Otsu's 方法

- 假设图像像素可分为两类，选择分割阈值，使得类内方差最小。

给定一个图像 I ，计算其归一化的直方图 $P(i)$

选择阈值 t ，将像素分为两类，每类概率：

$$q_1(t) = \sum_{i=1}^t P(i) \quad q_2(t) = \sum_{i=t+1}^T P(i)$$

每类像素的灰度均值：

$$\mu_1(t) = \sum_{i=1}^t \frac{i \cdot P(i)}{q_1(t)} \quad \mu_2(t) = \sum_{i=t+1}^T \frac{i \cdot P(i)}{q_2(t)}$$

9

Otsu's 方法



每类像素的灰度方差：

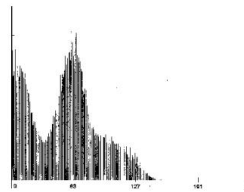
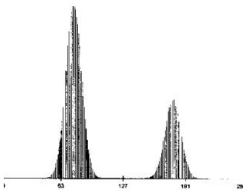
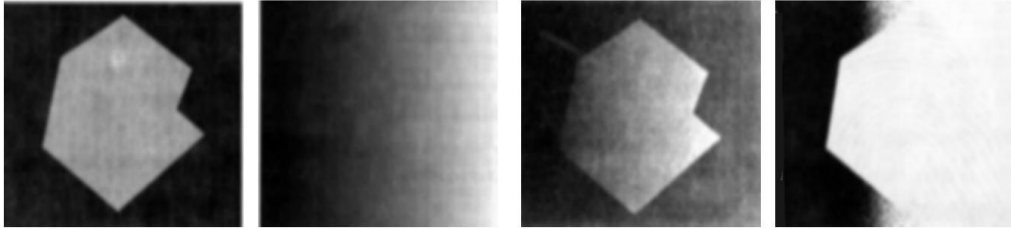
$$\sigma_1^2(t) = \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \quad \sigma_2^2(t) = \sum_{i=t+1}^T [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}$$

- 加权类内方差： $\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$
- 最优阈值选择： $\tau = \arg \min_t \sigma_w^2(t)$

- 二值化图像分割： $B(i, j) = \begin{cases} 1 & I(i, j) \geq \tau \\ 0 & I(i, j) < \tau \end{cases}$

10

光照不均匀对分割的影响

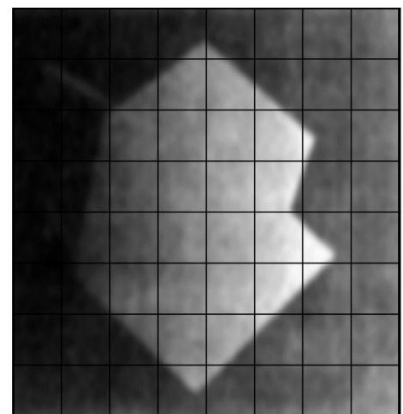


11

变化阈值法

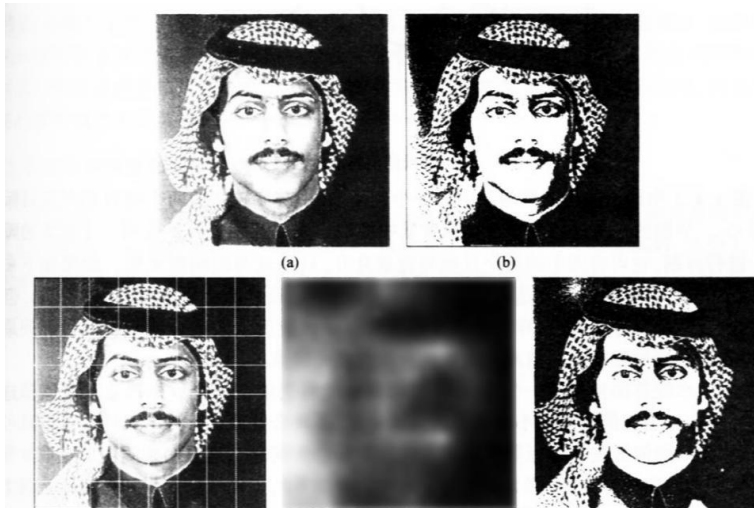


1. 将整幅图像分成一系列互相之间有50%重叠的子图像；
2. 做出每个子图像的直方图；
3. 检测各个子图像的直方图是否为双峰的。如是，则采用最优阈值法确定一个阈值；否则，不进行处理；
4. 根据对直方图为双峰的子图像得到的阈值通过插值得到所有子图像的阈值；
5. 根据各子图像的阈值，再通过插值得到所有像素的阈值，然后对图像进行分割



12

分块取阈值

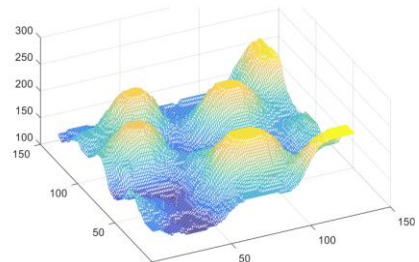
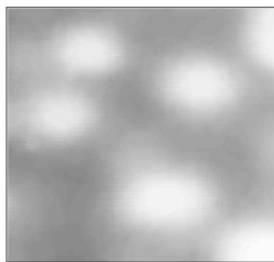


13

分水岭分割算法



- 分水岭（watershed，也称分水线/水线）
- 把图象看成3-D地形的表示，即2-D的地基（对应图像空间）加上第3维的高度（对应图像灰度）
- 图像的梯度图也可以视为3D地形（图像梯度图3D地形中的峰岭对应目标的边界）

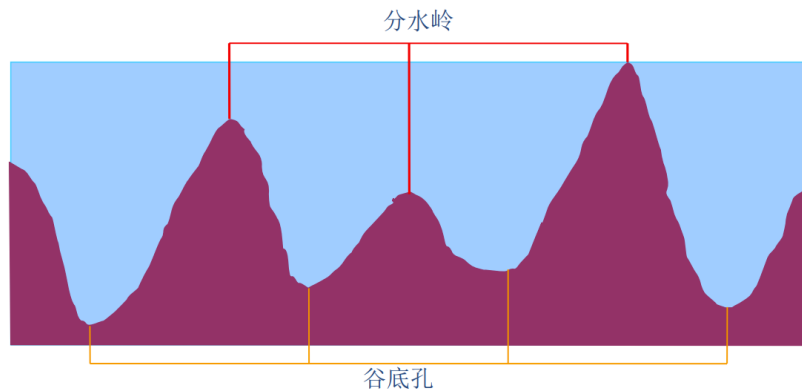


14

分水岭分割算法



- 分水岭算法原理（建立不同目标间的分水岭）
- 区域内部灰度值接近，区域间灰度值差别较大，所以一般使用梯度图的分水岭



15

基本原理和步骤

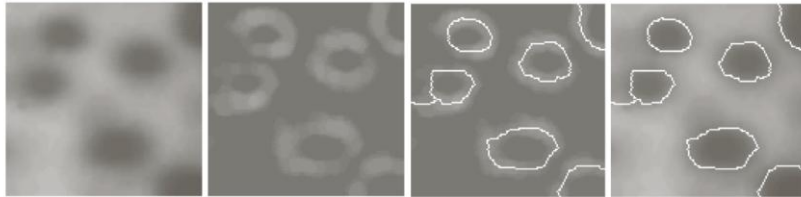


1. 待分割图像 $f(x, y)$ ，其梯度图像为 $g(x, y)$
2. 用 M_1, M_2, \dots, M_R 表示 $g(x, y)$ 中各局部极小值的像素位置， $C(M_i)$ 为与 M_i 对应的区域中的像素坐标集合。
3. 用 n 表示当前灰度阈值， $T[n]$ 代表记为 (u, v) 的像素集合， $g(u, v) < n$ ，
4. 对 M_i 所在的区域，其中满足条件的坐标集合 $C_n(M_i)$ 可看作一幅二值图像：

$$C_n(M_i) = C(M_i) \cap T[n]$$

16

分水岭分割算法实例

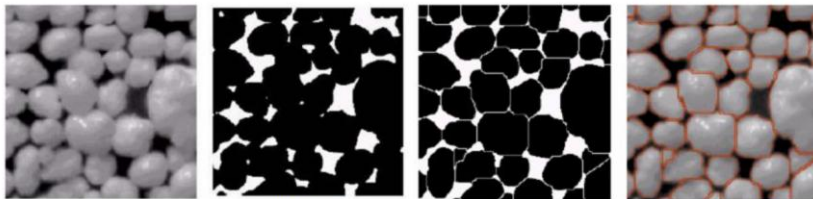


原始图

梯度图

梯度图上的分水岭

原图上的分水岭



原始图

阈值分割

分水岭

叠加轮廓

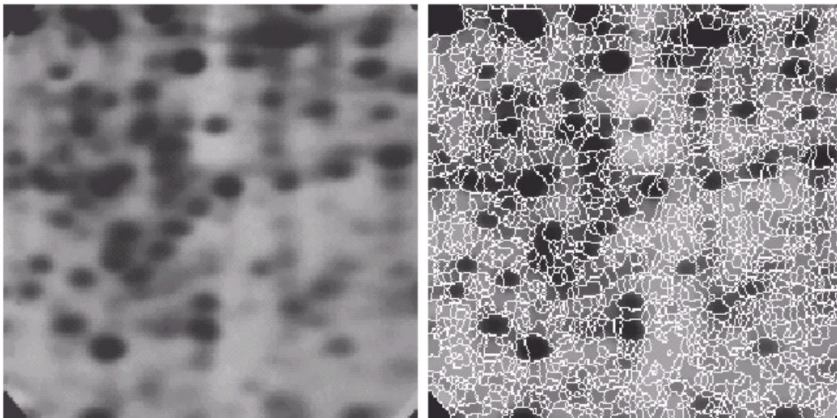
17

过分割 (over-segmentation)



- 分水岭对于图像的变化非常敏感

分得太细、切得太碎



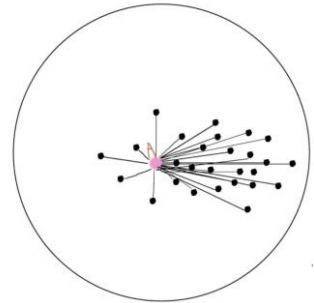
18

Meanshift分割



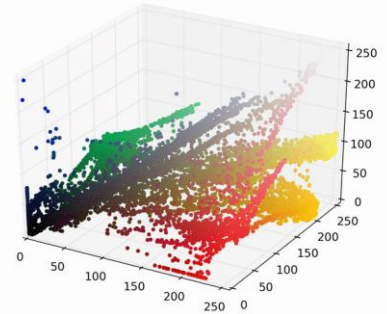
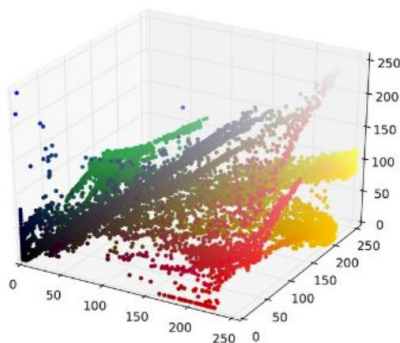
均值漂移 (Mean Shift) 分割是一种基于**密度聚类**的非参数图像分割算法。它的核心思想是：将图像中的每个像素看作一个多维空间（包含空间坐标和颜色信息）中的点，通过迭代地“漂移”到数据密度最大的区域（即**模式**），最终将收敛到同一模式的像素归为一类，实现分割。

- 对于每一个点，算法会定义一个以该点为中心、半径为固定值（即**带宽**）的“窗口”。
- 然后，计算这个窗口内所有点的“均值”（即窗口的中心）。
- 接着，将窗口中心移动到新计算出的均值点位置。
- 重复这个过程，直到窗口的中心不再移动或移动距离小于设定阈值。最终的位置，就是这个窗口内所有点集的“**模式**”，即局部密度最大处。



19

Meanshift分割



20

传统的图像分割方法存在的问题



- 基于阈值的分割：难以选择合适的阈值
- 边缘检测：边缘没有闭合，难以得到完整目标区域

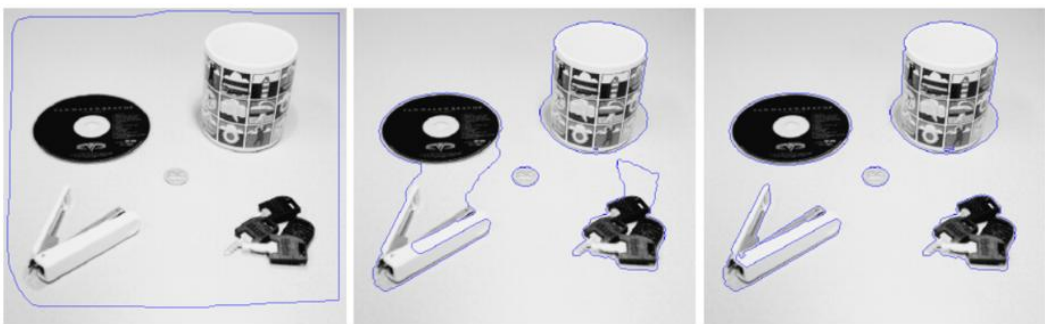


21

基于主动轮廓模型的图像分割



- 基本思想：通过在图像中演化一组动态曲线，自适应地控制曲线的演化速度，使得其停驻在物体边缘处，从而检测出图像中感兴趣的物体

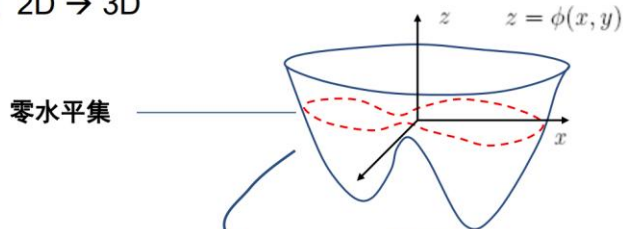


22

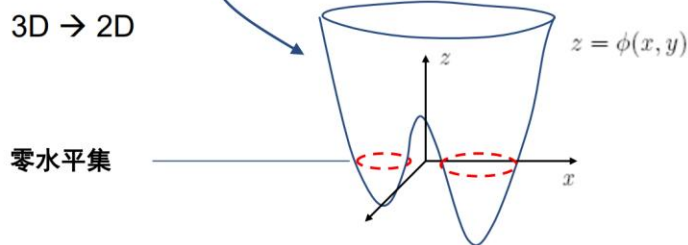
水平集描述



□ 2D → 3D



□ 3D → 2D

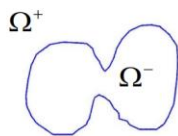


23

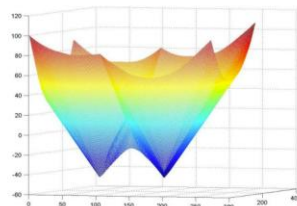
符号距离函数 (Signed distance function)



轮廓 C



符号距离函数



$$\text{符号距离函数定义: } \phi(\mathbf{x}) = \begin{cases} \text{dist}(\mathbf{x}, C) & \text{if } \mathbf{x} \text{ is outside } C \\ 0 & \mathbf{x} \in C \\ -\text{dist}(\mathbf{x}, C) & \text{if } \mathbf{x} \text{ is inside } C \end{cases}$$

可以证明: $|\nabla\phi(\mathbf{x})| = 1$

24



特殊的演化方程（了解）

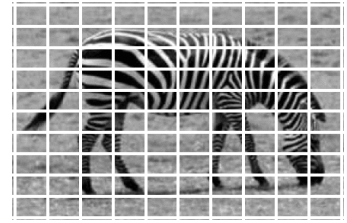
□ 平均曲率运动 $\frac{\partial \mathcal{C}}{\partial t} = \kappa \mathcal{N}$ \longrightarrow $\frac{\partial \phi}{\partial t} = \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) |\nabla \phi|$

□ 常速率运动 $\frac{\partial \mathcal{C}}{\partial t} = c \mathcal{N}$ \longrightarrow $\frac{\partial \phi}{\partial t} = c |\nabla \phi|$

□ 对流运动 $\frac{\partial \mathcal{C}}{\partial t} = \langle \bar{V}, \mathcal{N} \rangle \mathcal{N}$ \longrightarrow $\frac{\partial \phi}{\partial t} = -\langle \bar{V}, \nabla \phi \rangle$

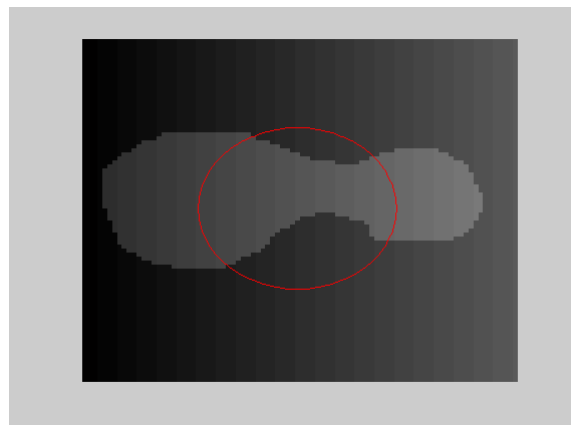
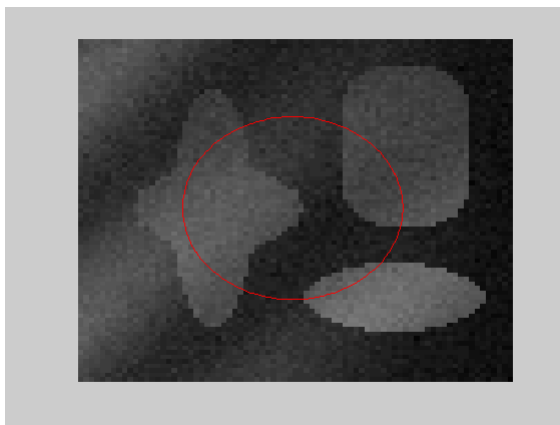
□ 测地线主动轮廓 $\frac{\partial \mathcal{C}}{\partial t} = (c + g) \kappa \mathcal{N} - \langle \nabla g, \mathcal{N} \rangle \mathcal{N}$

$$\frac{\partial \phi}{\partial t} = \left(c + \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right) |\nabla \phi| + \langle \nabla g, \nabla \phi \rangle$$



25

水平集分割实例

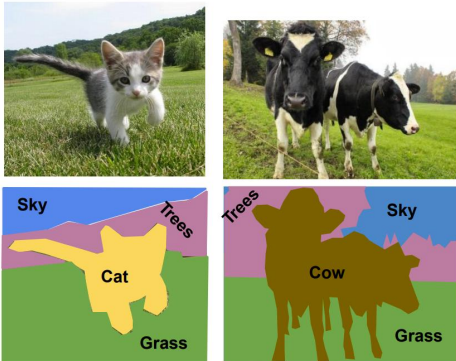


26



6.2语义分割概述

- 给每个像素分配类别标签
- 不区分实例，只考虑像素类别



Input

0:background
1:person
2:bottle

Segmented



Labels

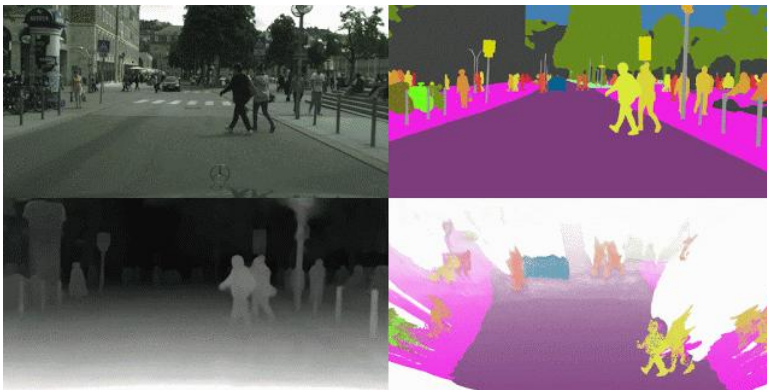


Visual



Ground truth

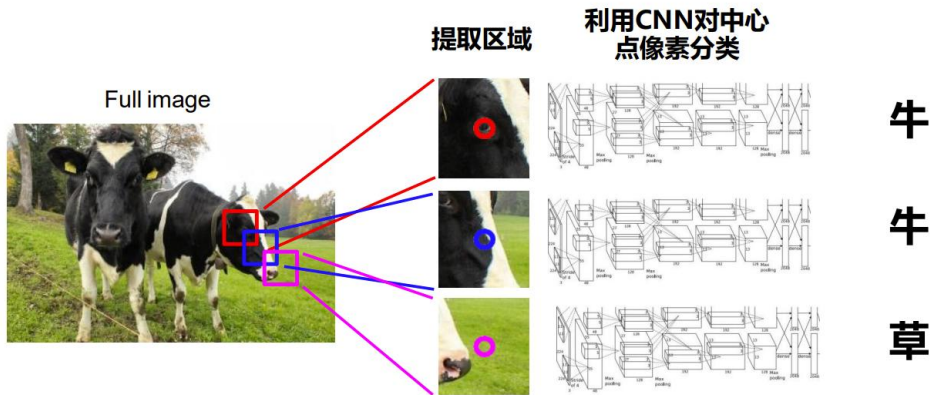
自动驾驶汽车



在自动驾驶汽车中，环境感知是一项至关重要的技术，它赋予汽车实时分析和理解周围道路场景的能力。这个过程中，语义分割技术发挥着重要作用。图为自动驾驶过程中实时分割道路场景。



6.3语义分割思路：滑动窗口

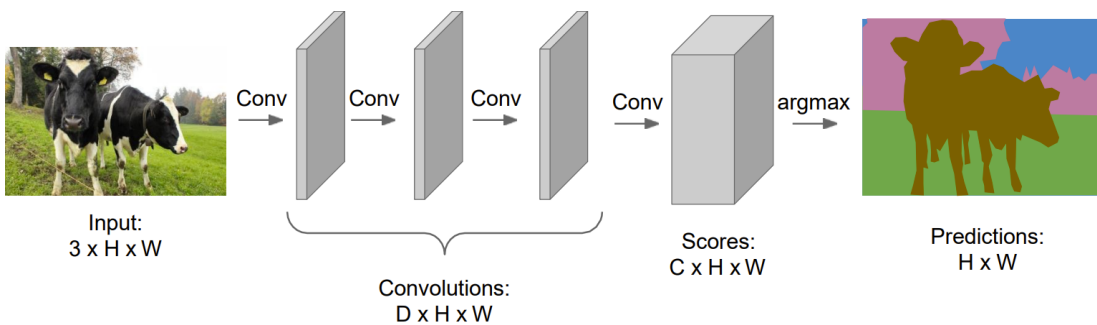


问题: 效率太低! 重叠区域的特征反复被计算

Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013
Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

29

6.3语义分割思路：全卷积



解决方案: 让整个网络只包含卷积层, 一次性输出所有像素的类别预测。

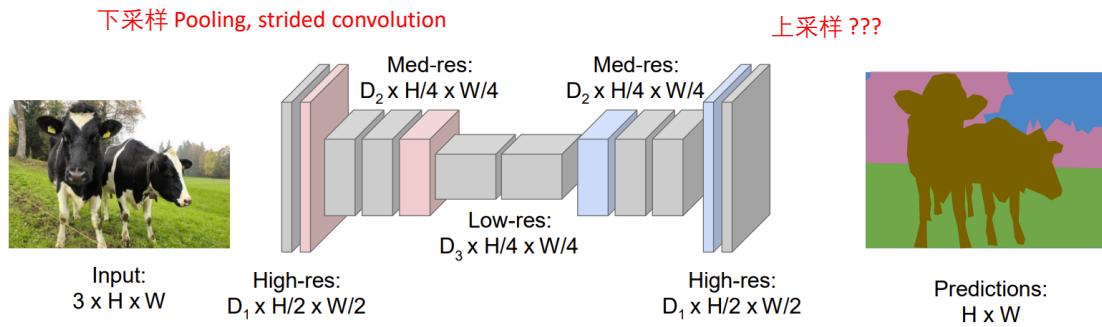
问题: 处理过程中一直保持原始分辨率, 对于显存的需求会非常庞大...

30

6.3语义分割思路：全卷积



解决方案：让整个网络只包含卷积层，并在网络中嵌入下采样与上采样过程。



Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

31

反池化操作：“Unpooling”



Nearest Neighbor

1	2
3	4

Input: 2×2

1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

Output: 4×4

“Bed of Nails”

1	2
3	4

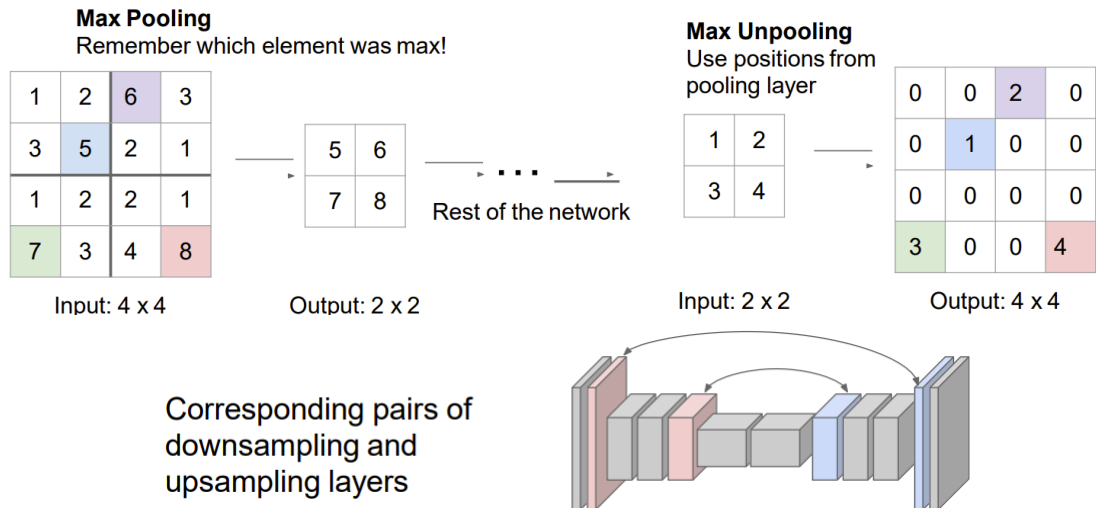
Input: 2×2

1	0	2	0
0	0	0	0
3	0	4	0
0	0	0	0

Output: 4×4

32

反池化操作：“Max Unpooling”

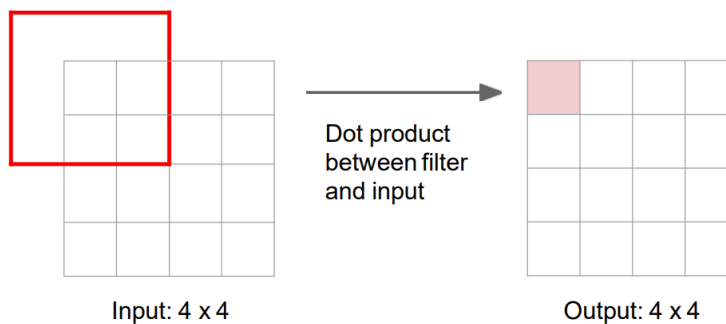


33

转置卷积



- 可学习的上采样: 转置卷积 (Transpose Convolution)
- 回顾: 3 x 3 卷积, 步长 (stride) 1, 零填充 (pad) 1

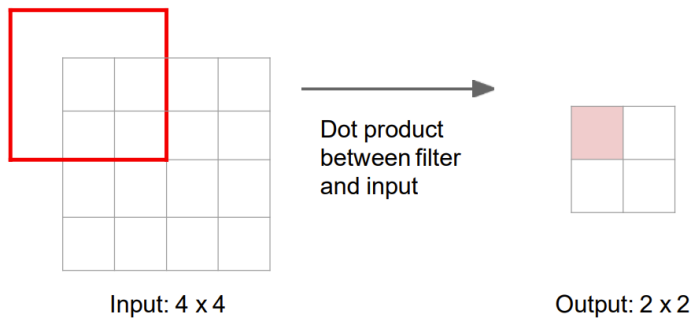


34

转置卷积



- 回顾: 3×3 卷积, 步长 (stride) 2, 零填充 (pad) 1

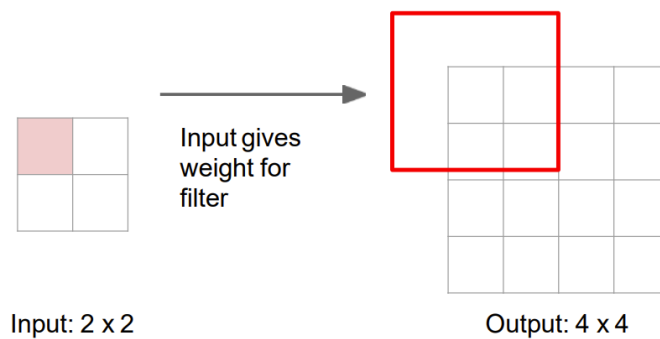


35

转置卷积



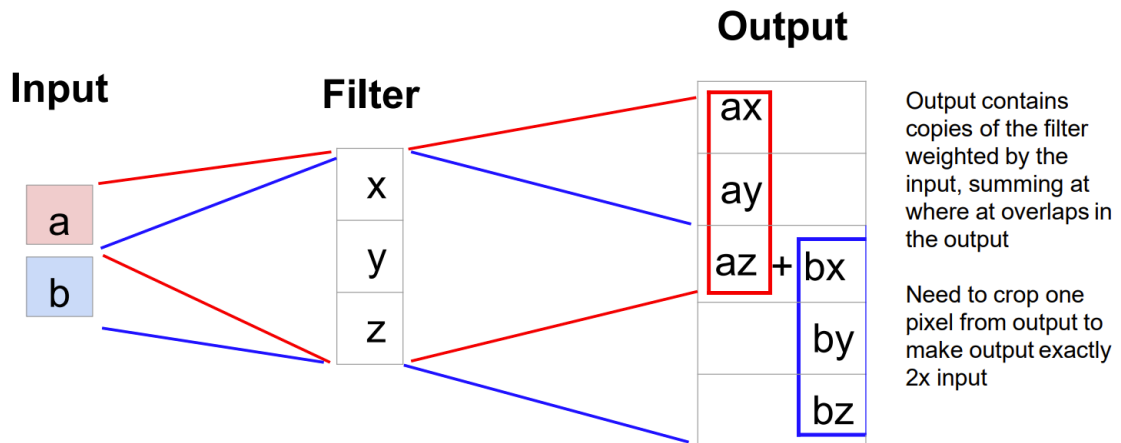
- 可学习的上采样: 转置卷积 (Transpose Convolution)
- 3×3 转置卷积 (transpose convolution), stride 2 pad 1



36



可学习的上采样: 一维例子



37

卷积与矩阵相乘 (一维例子)



- 将卷积写为矩阵乘法

Convolution transpose multiplies by the transpose of the same matrix:

$$\vec{x} * \vec{a} = X\vec{a}$$

$$\begin{bmatrix} x & y & x & 0 & 0 & 0 \\ 0 & x & y & x & 0 & 0 \\ 0 & 0 & x & y & x & 0 \\ 0 & 0 & 0 & x & y & x \end{bmatrix} \begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ay + bz \\ ax + by + cz \\ bx + cy + dz \\ cx + dy \end{bmatrix}$$

$$\vec{x} *^T \vec{a} = X^T \vec{a}$$

$$\begin{bmatrix} x & 0 & 0 & 0 \\ y & x & 0 & 0 \\ z & y & x & 0 \\ 0 & z & y & x \\ 0 & 0 & z & y \\ 0 & 0 & 0 & z \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} ax \\ ay + bx \\ az + by + cx \\ bz + cy + dx \\ cz + dy \\ dz \end{bmatrix}$$

例子: 1D 卷积, 卷积核尺寸=3, 步长=1, 零填充=1

38

卷积与矩阵相乘 (一维例子)



- 将卷积写为矩阵乘法

Convolution transpose multiplies by the transpose of the same matrix:

$$\vec{x} * \vec{a} = X\vec{a}$$

$$\begin{bmatrix} x & y & x & 0 & 0 & 0 \\ 0 & 0 & x & y & x & 0 \end{bmatrix} \begin{bmatrix} 0 \\ a \\ b \\ c \\ d \\ 0 \end{bmatrix} = \begin{bmatrix} ay + bz \\ bx + cy + dz \end{bmatrix}$$

$$\vec{x} *^T \vec{a} = X^T \vec{a}$$

$$\begin{bmatrix} x & 0 \\ y & 0 \\ z & x \\ 0 & y \\ 0 & z \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} ax \\ ay \\ az + bx \\ by \\ bz \\ 0 \end{bmatrix}$$

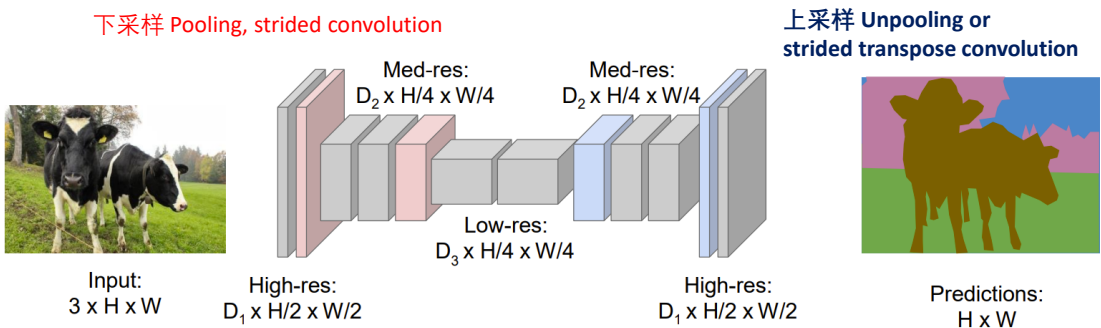
例子: 1D 卷积, 卷积核尺寸=3, 步长=2, 零填充=1

39

语义分割思路: 全卷积



解决方案: 让整个网络只包含卷积层, 并在网络中嵌入下采样与上采样过程。



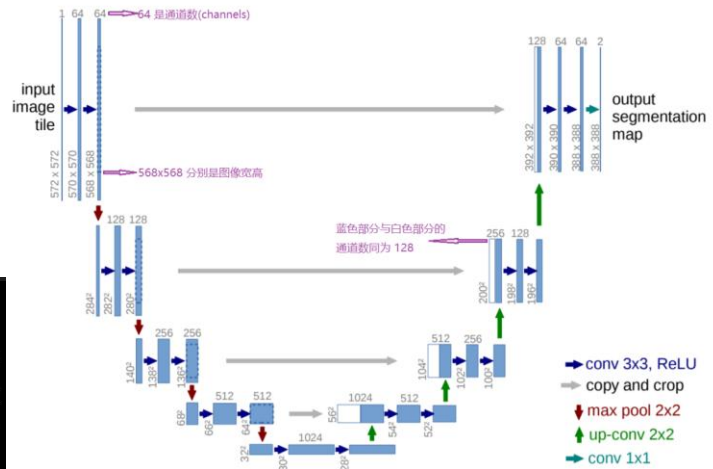
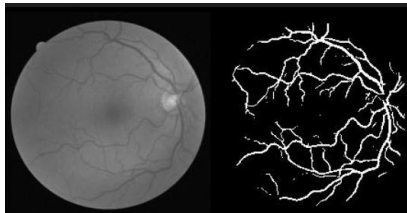
Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
 Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

40

UNet

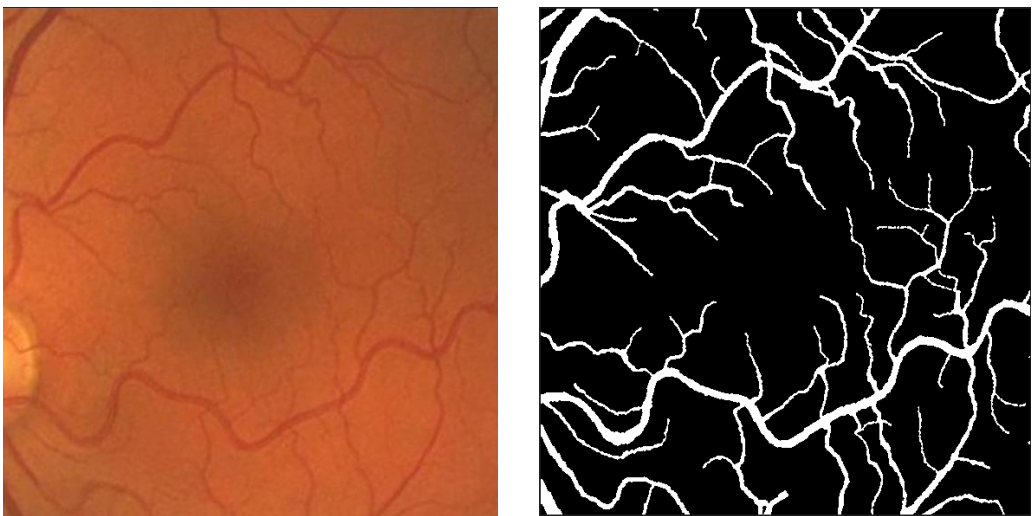


- Unet 属于 FCN 的一种变体。



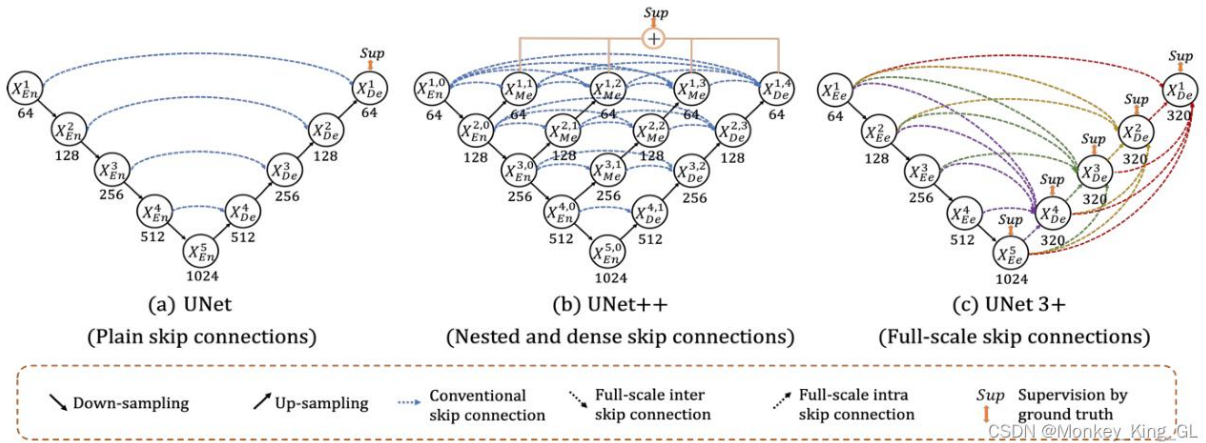
41

UNet



42

UNet++

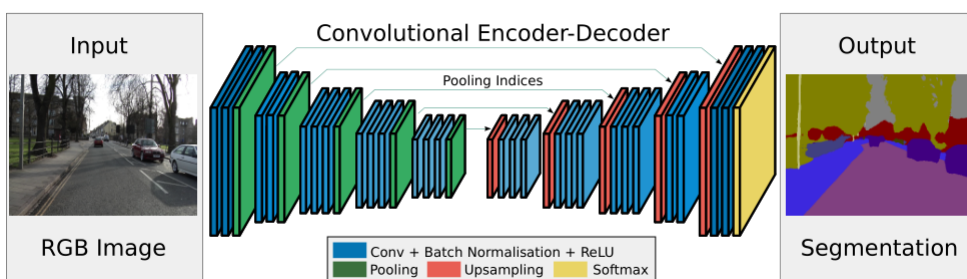


43

6.4 语义分割发展 SegNet



- 主要贡献：将最大池化指数转移至解码器中，改善了分割分辨率。
- 在FCN网络中，通过上卷积层和一些跳跃连接产生了粗糙的分割图，为了提升效果而引入了更多的跳跃连接。然而，FCN网络仅仅复制了编码器特征，而Segnet网络复制了最大池化指数。这使得在内存使用上，SegNet比FCN更为高效。



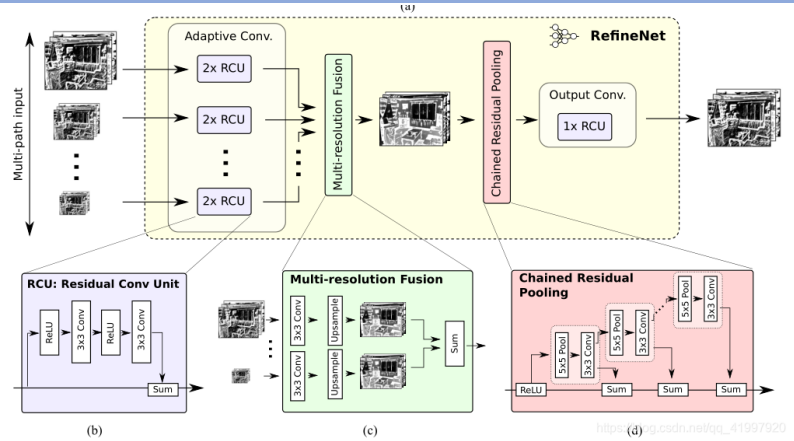
SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation (2015/11/2)

44

RefineNet



- 包含一个能通过对较低分辨率特征进行上采样来融合多分辨率特征的组件，以及一个能基于步幅为1及5×5大小的重复池化层来获取背景信息的组件。
- 遵循恒等映射的思想
- 使用空洞卷积
- 采用残差连接的设计方式。



恒等映射：将模块的输入直接"跳过"中间的计算层，与最终输出相加。

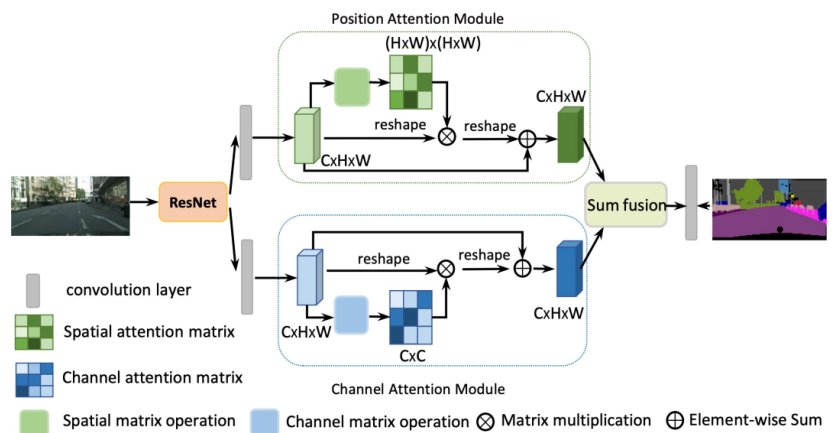
RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation (2016/11/20)

45

DANet (Dual attention network)



- FCN附加两种类型的注意力模块:模拟空间和通道维度中的语义相互依赖性。
- Position Attention** 模块通过对所有位置的特征进行加权和,有选择地聚合每个位置的特征。
- Channel Attention** 模块通过整合所有信道映射中的相关特征来选择性地强调相互依赖的信道映射。



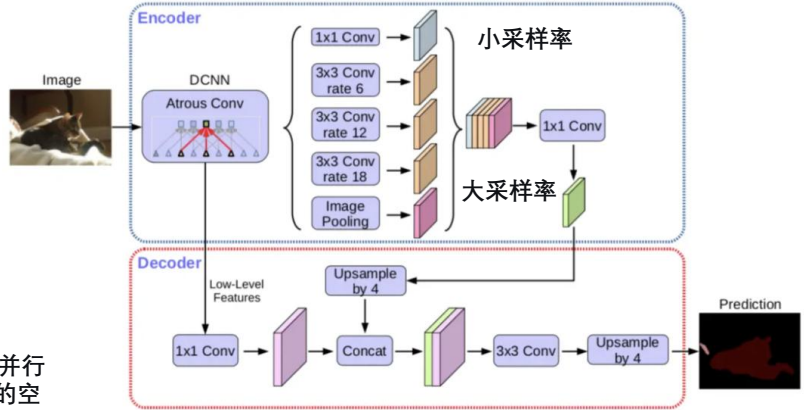
DANet: Dual Attention Network for Scene Segmentation (CVPR2019)

46

DeepLabV3+



- 采用Encoder-Decoder架构。
- Encoder使用类似Xception的结构作为backbone，使用空洞卷积空间金字塔池化，来实现不同尺度的特征融合。
- Decoder再次使用跨层级的concat操作进行高低层次的特征融合。



ASPP的核心操作是：将输入的特征图并行输入到多个采用不同采样率（Rate）的空洞卷积层中。

<https://github.com/open-mmlab/msegmentation/tree/master/configs/deeplabv3plus>

47

SegFormer

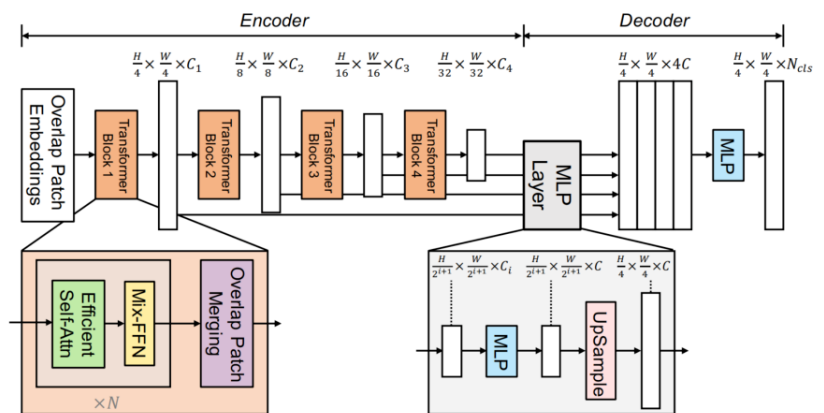


由NVIDIA团队在2021年提出。

将Transformer架构与轻量级MLP解码器结合的统一框架

两大创新：

- (1) 分层结构 (Hierarchical)：多尺度特征
- (2) 无需位置编码 (No Positional Encoding)：3x3 深度可分离卷积



SegFormer专注于高效、轻量的实时语义分割。

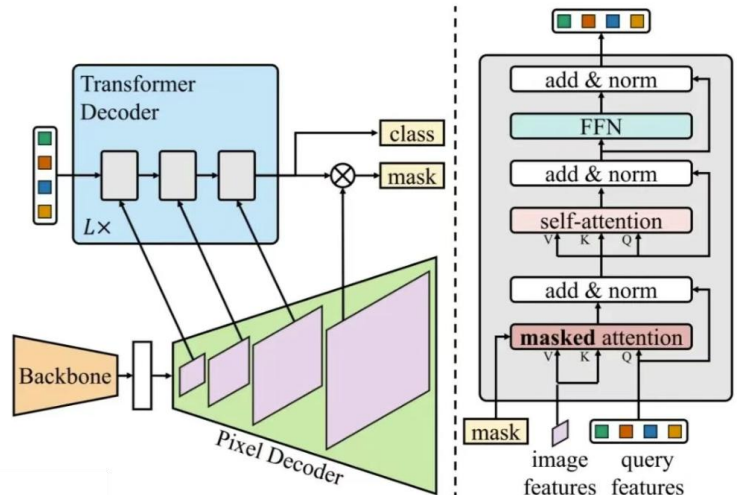
48

Mask2Former



- 首次使用同一个模型、同一套参数和损失函数，在语义、实例和全景三大主流分割任务上均超越了当时专门为各个任务设计的顶尖模型。

1. 特征提取 (Backbone + Pixel Decoder)
2. 初始化查询 (Query 初始化)
3. Transformer 解码器迭代优化 (核心)
4. 分类与掩码输出

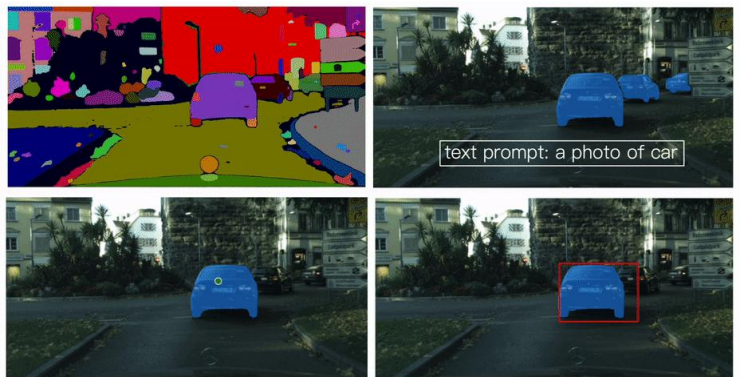


49

SAM (Segment Anything Model)



- Meta AI在2023年推出的一个图像分割基础模型。
- 首次让一个模型无需额外训练，就能通过简单的提示（点击、画框、文字等）来分割图像中的任何物体。
- 构建了迄今为止最大的分割数据集SA-1B(1100万张, 11亿个掩码)



- 2023 SAM 1: 实现图像内“点哪切哪”的交互式通用分割。
- 2024 SAM 2: 将能力从单张图像扩展到视频，实现“一点选择，全视频跟踪分割”。
- 2025 SAM 3: 进化到无需精细点击，仅凭文字描述或示例图片就能分割出所有符合描述的目标

50

投票



- 题目：当分割“一切”时，我们是否也应该分割“隐私”？



匿名投票：你认为SAM应该开源吗？

51

6.5常用数据集



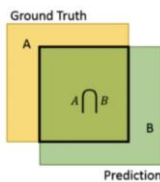
- VOC2012：有 20 类目标，这些目标包括人类、机动车类以及其他类，可用于目标类别或背景的分割
- MSCOCO：是一个新的图像识别、分割和图像语义数据集，是一个大规模的图像识别、分割、标注数据集。它可以用于多种竞赛，与本领域最相关的是检测部分，因为其一部分是致力于解决分割问题的。该竞赛包含了超过80个物体类别
- Cityscapes：50 个城市的城市场景语义理解数据集
- Stanford Background Dataset：至少有一个前景物体的一组户外场景。
- Pascal Context：有 400 多类的室内和室外场景

52

6.6评价指标



- IOU:用于评估语义分割算法性能的标准指标是平均 IOU（Intersection Over Union，交并比）
- mIoU:基于类进行计算的IoU就是将每一类的IoU计算之后累加，再进行平均，得到的就是基于全局的评价，也就是均交并比（mean IoU）
- pixcal-accuracy（PA，像素精度）：基于像素的精度计算是评估指标中最为基本也最为简单的指标，指预测正确的像素占总像素的比例。



$$\text{IoU} = (A \cap B) / (A \cup B)$$

$$mIoU = \frac{1}{k+1} \sum_{i=0}^k \frac{P_{ii}}{\sum_{j=0}^k P_{ij} + \sum_{j=0}^k P_{ji} - P_{ii}}$$

$$\text{PixAcc} = \frac{\sum_{i=0}^k P_{ii}}{\sum_{i=0}^k \sum_{j=0}^k P_{ij}}$$

53

案例



- <https://fzu.eduplus.net/home/index>

课程 计算机视觉

2023-2024第二学期 | 2024-03-26 至 2024-06-30 | 课时: 32课时 | 学院: 福州大学计算机与大数据学院

课程章节

作业

实验

讨论

资料

公告

大纲/教案

成绩统计

默认班级

请选择实验环境

实验2: 细胞分割 (UNet)

环境: 云沙箱 - 公有云

实验时间: 2024-04-13 19:00 至 2024-04-13 23:00

提交截止时间: 2024-04-14 23:59

实验1: 鲜花分类 (数据增强)

环境: 云沙箱 - 公有云

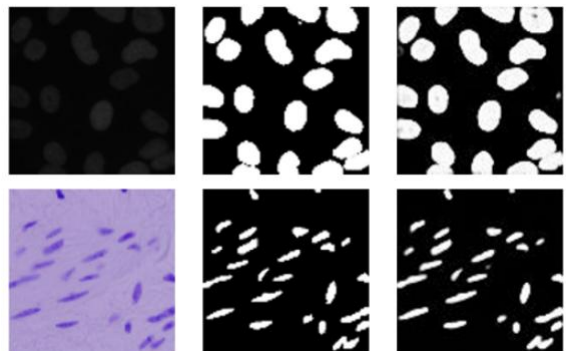
实验时间: 2024-04-01 20:00 至 2024-04-01 22:00

提交截止时间: 2024-04-09 23:59

原图

标签

分割结果



54



Thank you!