# 图像分割

董秋雷
中国科学院自动化研究所
qldong@nlpr.ia.ac.cn

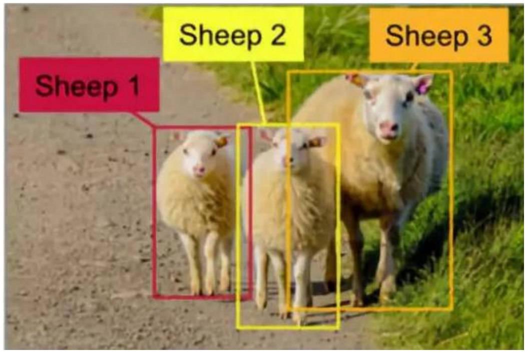
# 提纲

- 概述
- 早期的图像分割方法
- 基于特定理论的方法
- 基于深度神经网络的图像分割

# 什么是图像分割

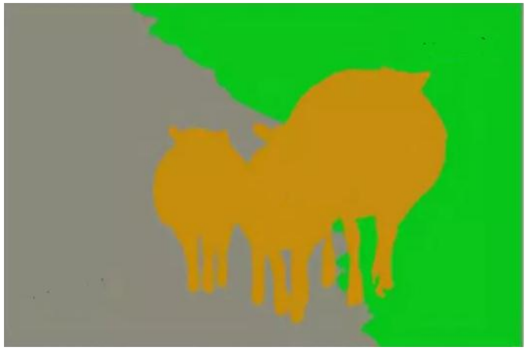
图像分割：把图像分成互不重叠的区域并提取出感兴趣目标的技术和过程。

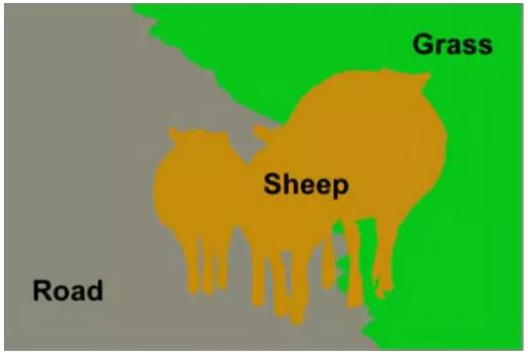
# 什么是图像分割



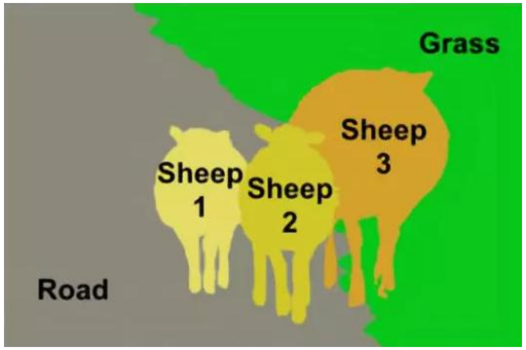

目标检测

图像分割：
- ■ 基本的图像分割
- ■ 语义分割（semantic segmentation）
- ■ 实例分割（instance segmentation）





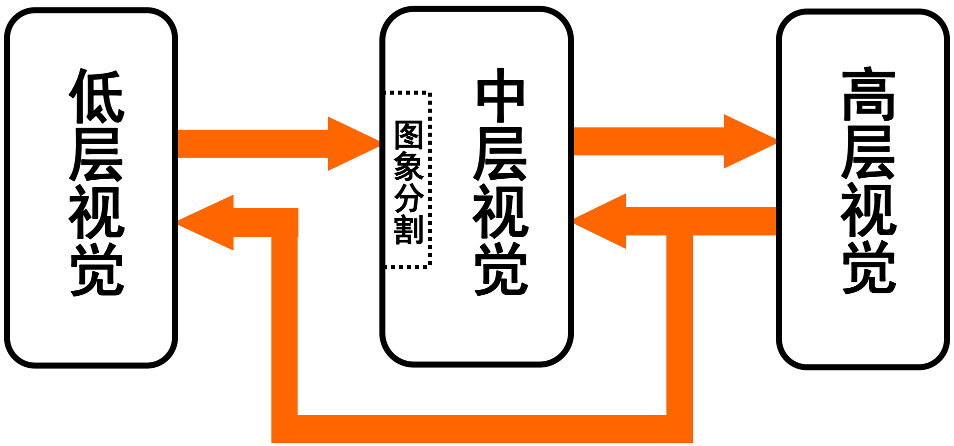

语义检测

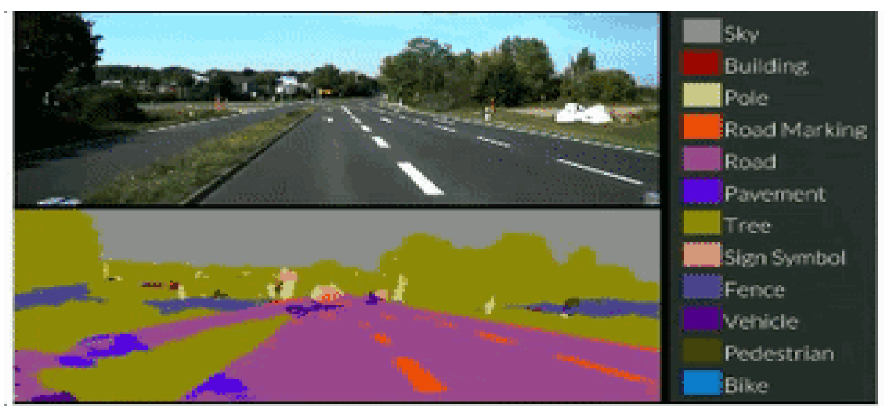



实例检测

# 为什么要图像分割

图像分割是由图像处理进到图像分析的关键步骤。它是目标表达的基础，使得更高层的图像分析和理解成为可能。

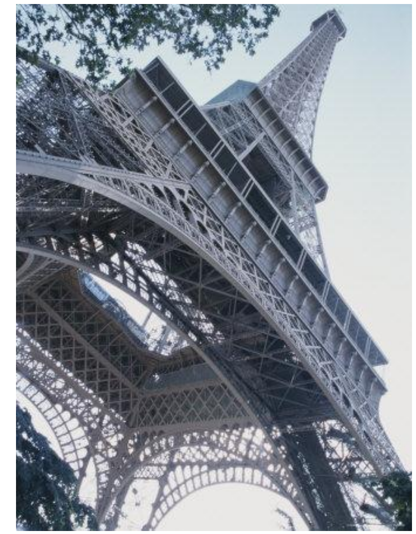


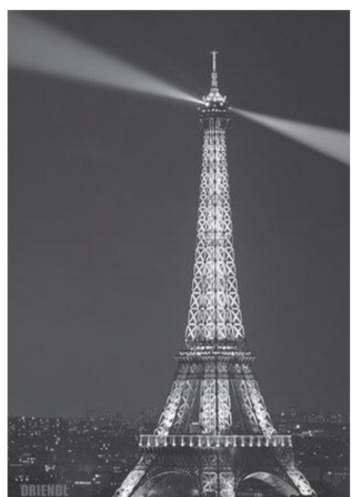



# 图像分割的应用领域

- 医学图像处理
- 遥感图像处理
- 目标跟踪
- 生物特征识别
- 等等

# 图像分割的难点





La Tour Eiffel 埃菲尔铁塔

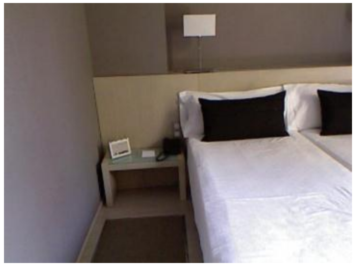
- 图像分割是中层视觉中的最基本问题，也是计算视觉和图像理解中的最基本问题之一。它还是该领域国际学术界公认的将会长期存在的最困难的问题之一。
- 从一般意义上来说，只有对图象内容的彻底理解，才能产生完美的分割。

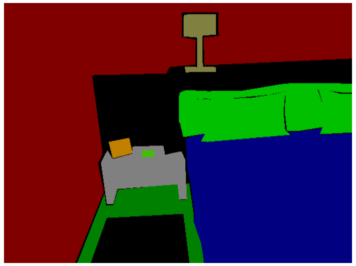
# 图像分割的基本依据

基本依据

1. 区域内的一致性

2. 区域间的不一致性





# 提纲

- 概述
- 早期的图像分割方法
- 基于特定理论的方法
- 基于深度神经网络的图像分割

# 早期的图像分割方法

早期的图像分割方法：

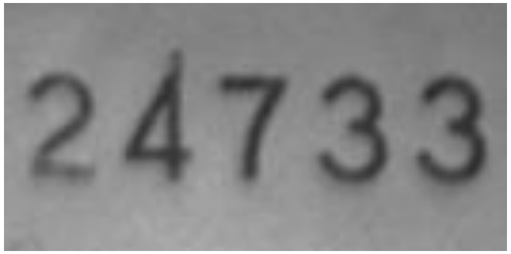
1. 阈值法
2. 区域生长法
3. 分裂合并法
4. 基于边缘的分割方法
5. 等等

# 阈值法

基本原理是:

- 通过设定不同的特征阈值，把图像像素点分为若干类．常用的特征包括：灰度、彩色特征、由原始灰度或彩色值变换得到的特征。

$$g(x,y)=\begin{cases}1, & f(x,y)\geq T\\ 0, & f(x,y)<T\end{cases}$$




# 阈值法

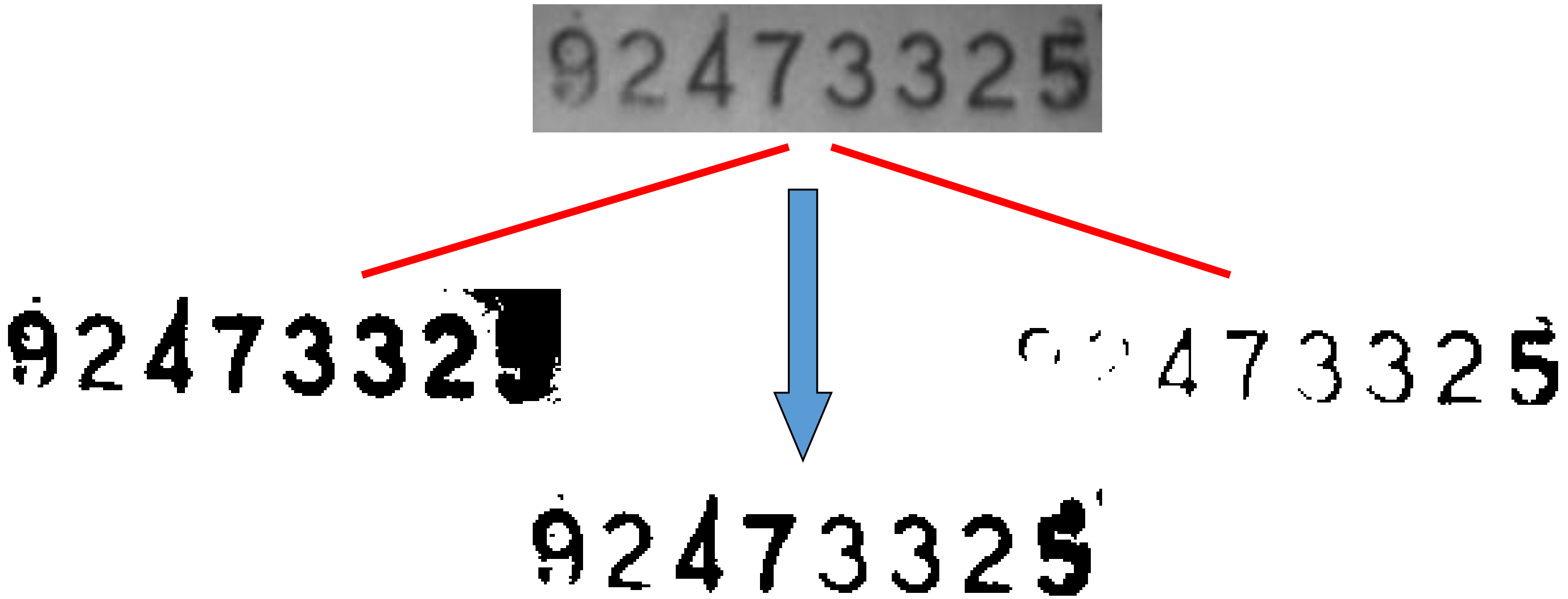
- 阈值分割法的关键是如何选取合适的阈值！
- 如果阈值选取过高，则过多的目标点被错误的归为背景；阂值选得过低，则会出现相反的情况。
- 相应地，大量基于阈值的分割方法涌现出来。

$$g(x,y)=\begin{cases}1, & f(x,y)\geq T\\ 0, & f(x,y)<T\end{cases}$$

# 局部阈值法

基本原理：

- 将图象分块，分别用全局阈值方法分割，最后再综合。



# 局部阈值法

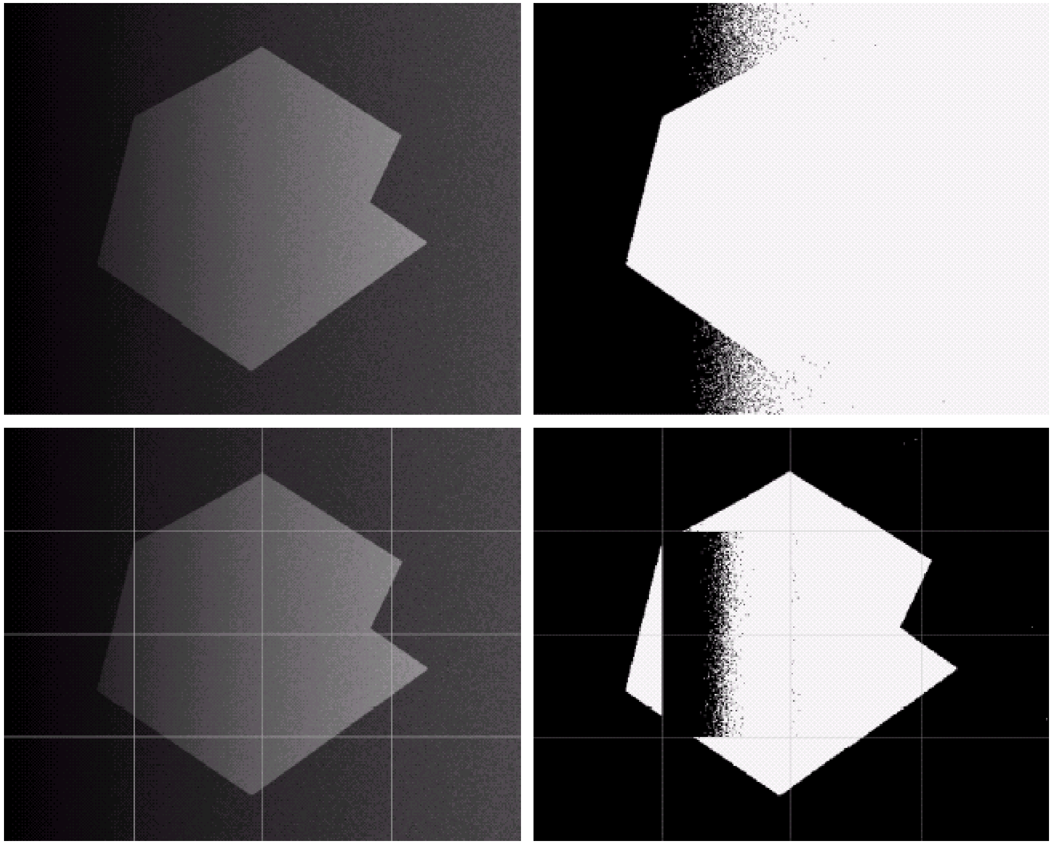


FIGURE 10.30 (a) Original image. (b) Result of global thresholding. (c) Image subdivided into individual subimages. (d) Result of adaptive thresholding.

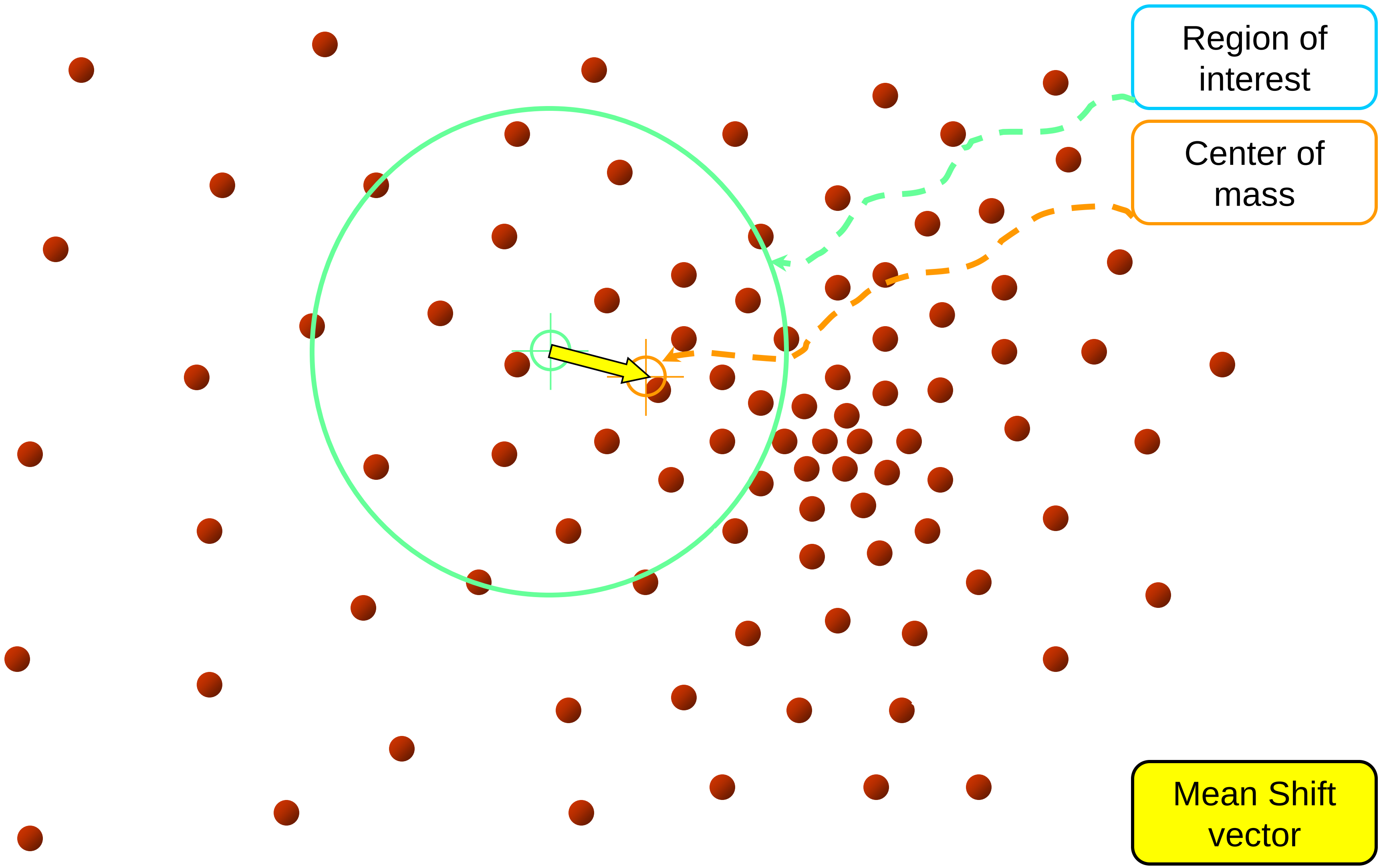
# 提纲

- 概述
- 早期的图像分割方法
- 基于特定理论的方法
- 基于深度神经网络的图像分割

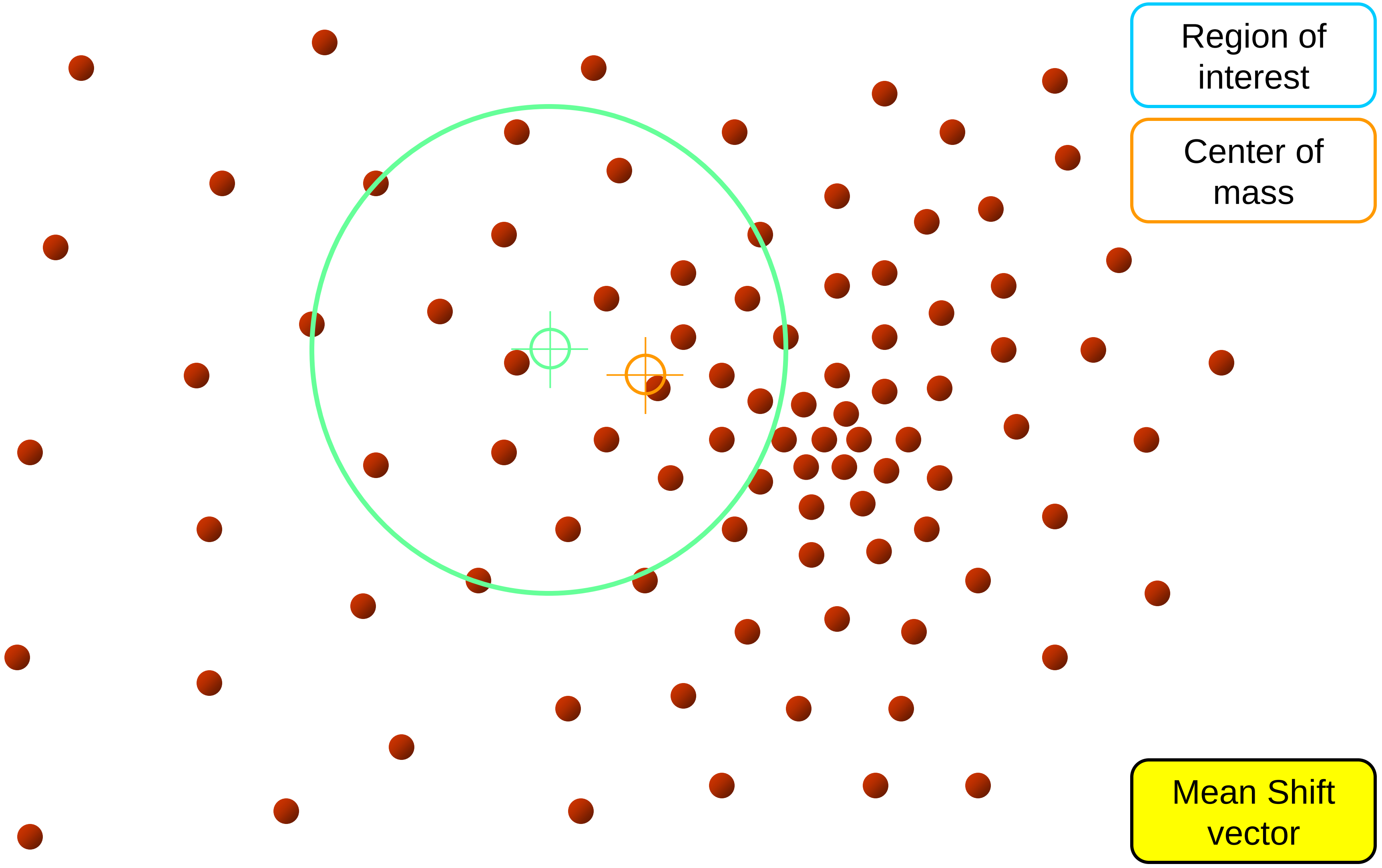
# 基于“均值移动”的图象分割方法

- 1975年，Fukunaga和Hostetle提出了一种基于一般核函数的非参数密度梯度的估计算法，并给出了保证估计值与真实值之间渐近无偏、一致和均匀连续时核函数应满足的条件。
- 1999年，Comaniciu将均值移动应用于图像分析。
- 核心思想:找到概率密度梯度为零的采样点，并以此作为特征空间聚类的模式点.

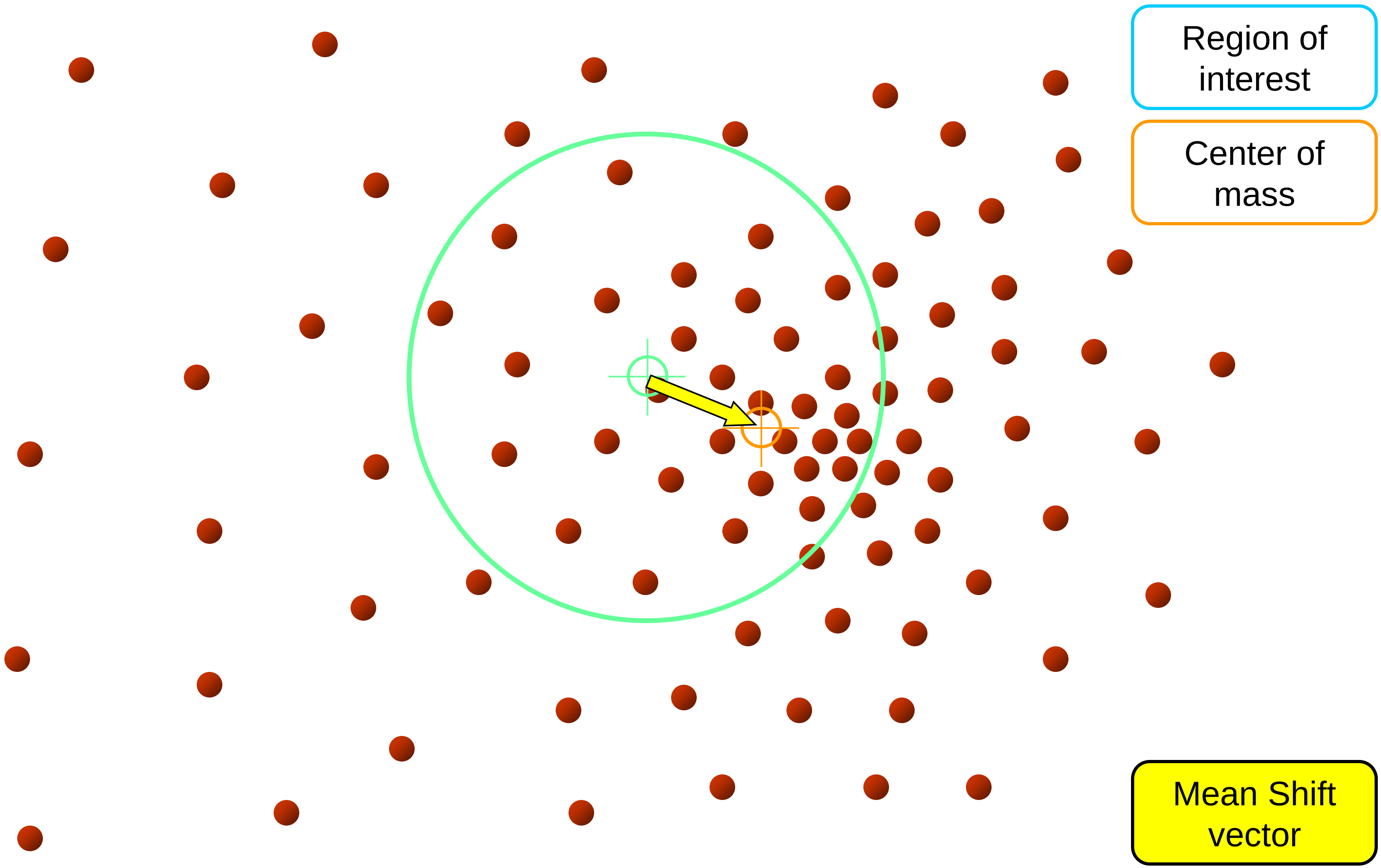
# Intuitive Description



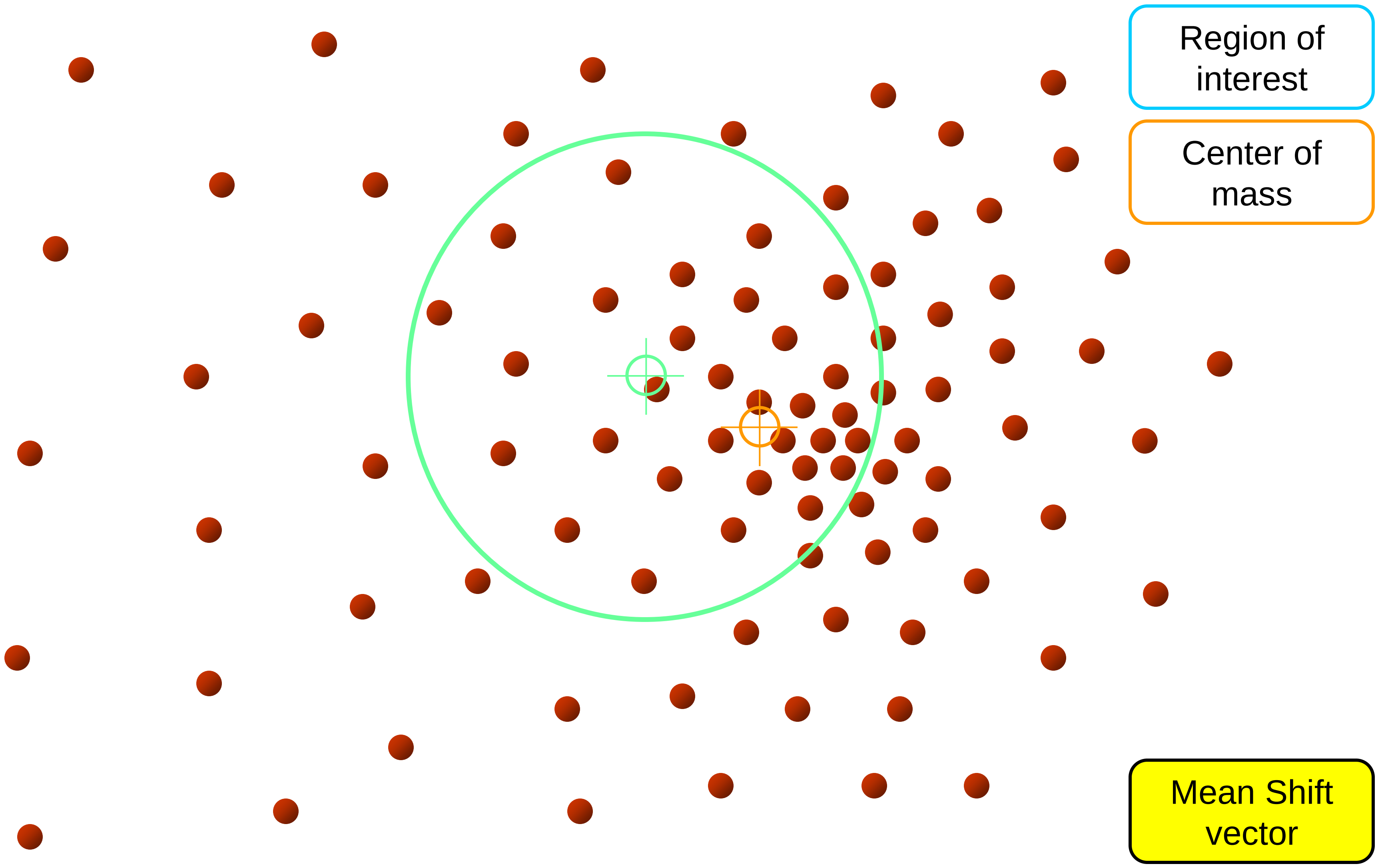
# Intuitive Description



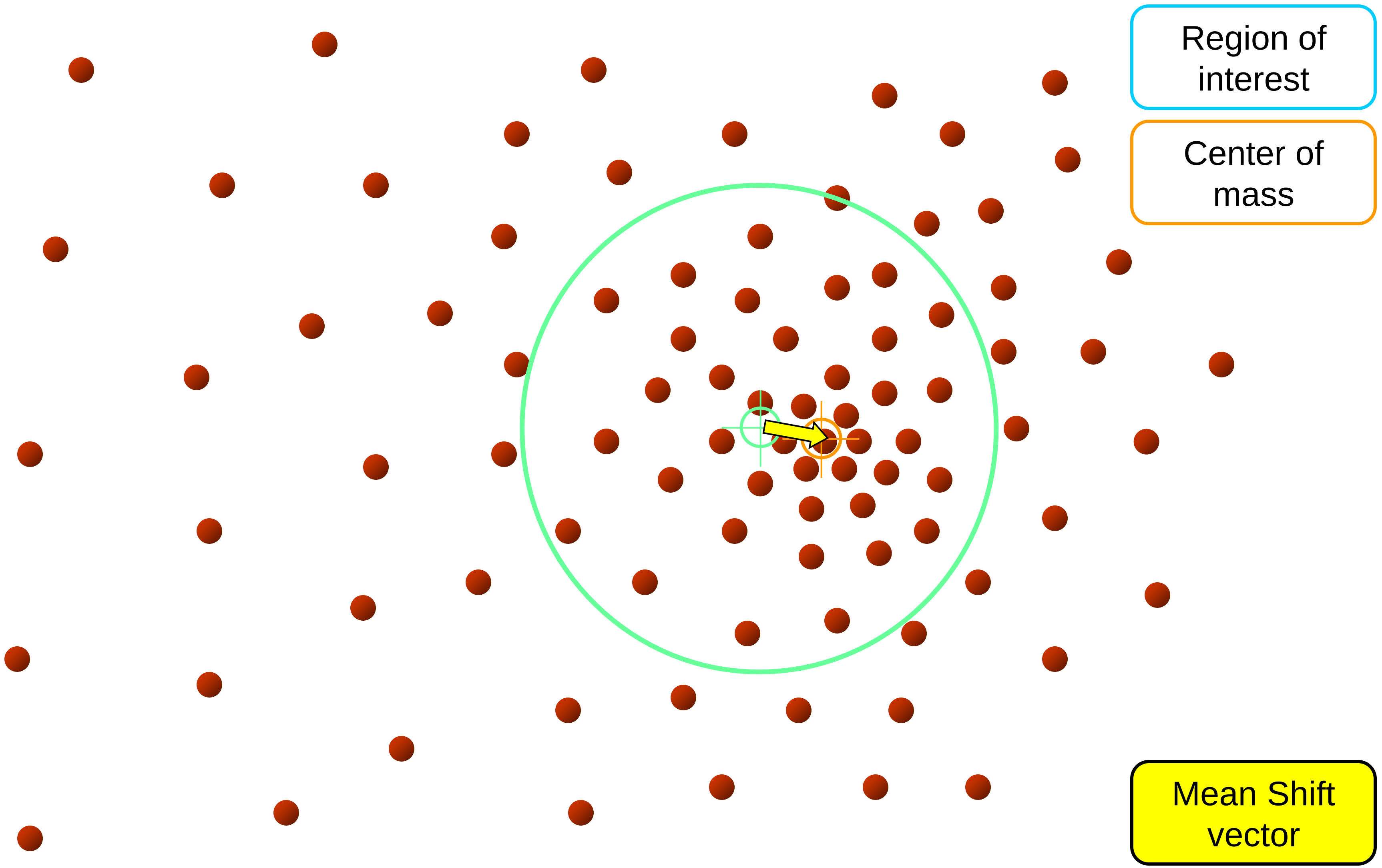
# Intuitive Description



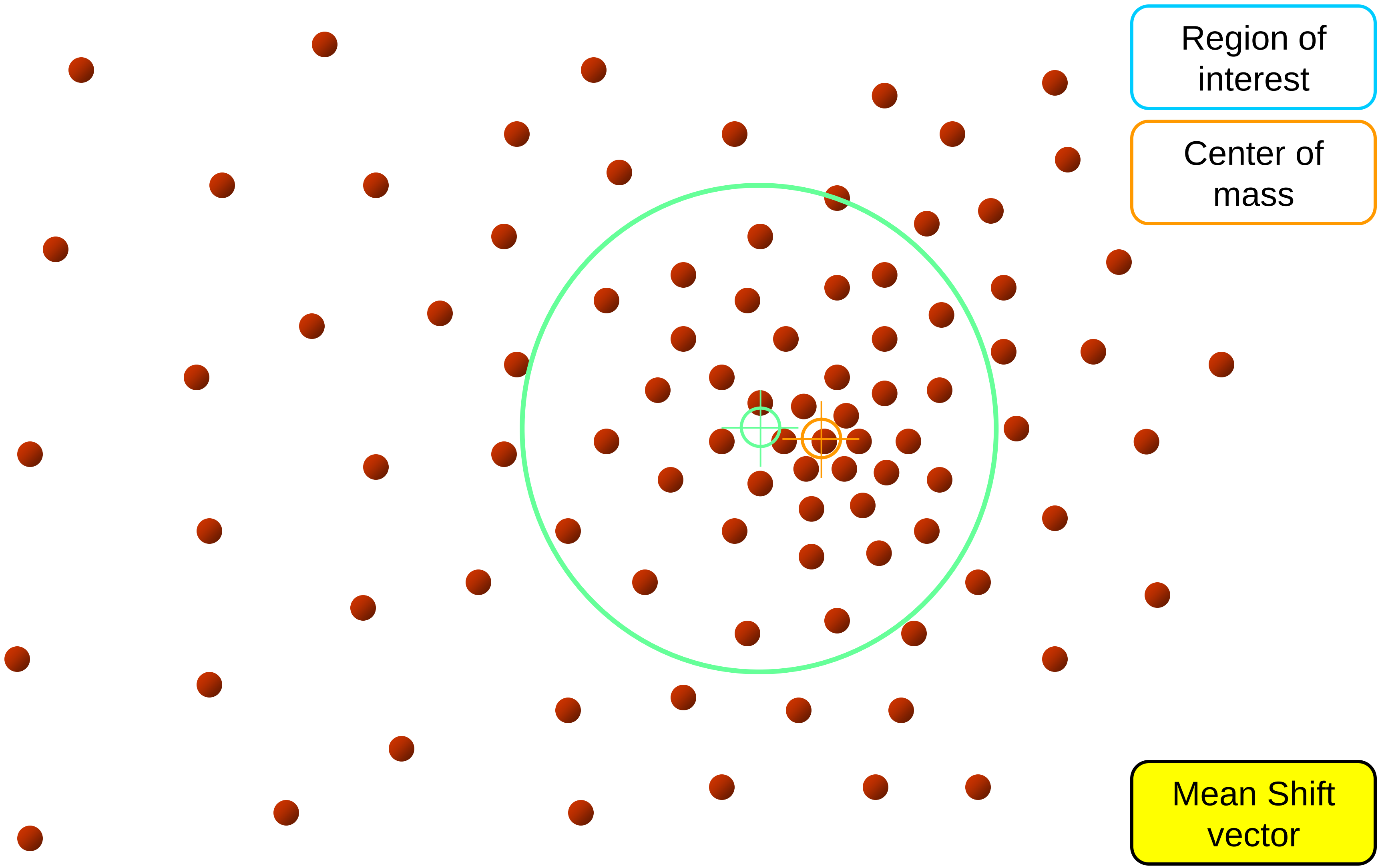
# Intuitive Description



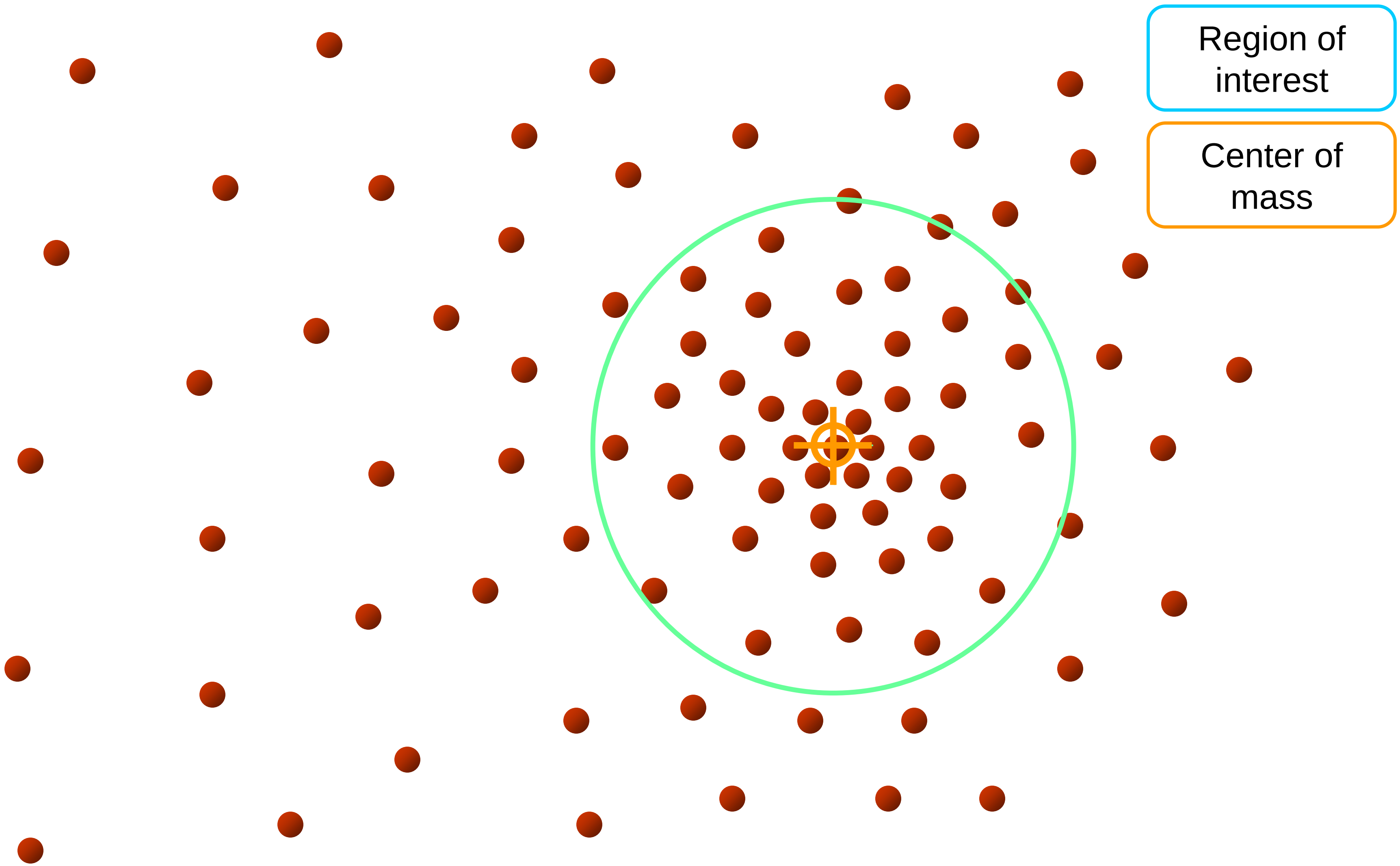
# Intuitive Description



# Intuitive Description



# Intuitive Description



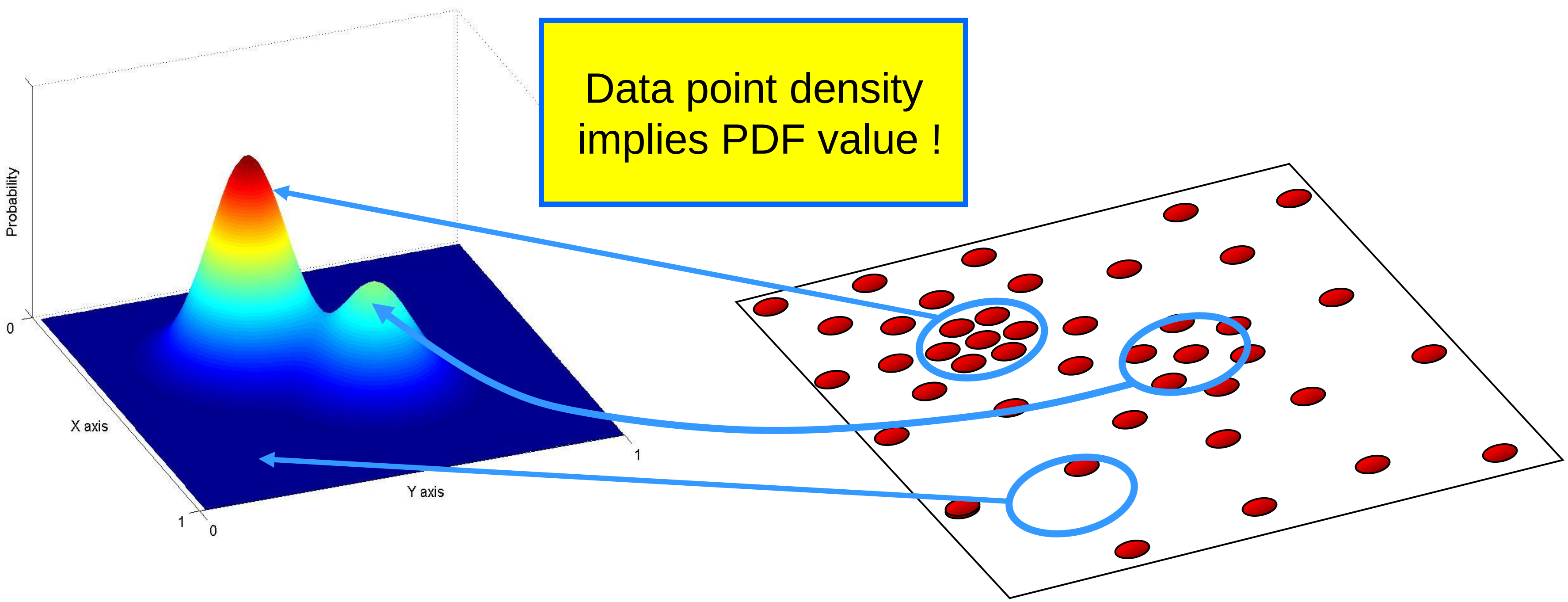
# 均值移动分割

核心思想：

- 找到概率密度梯度为零的采样点，并以此作为特征空间聚类的模式点。

# Non-Parametric Density Estimation

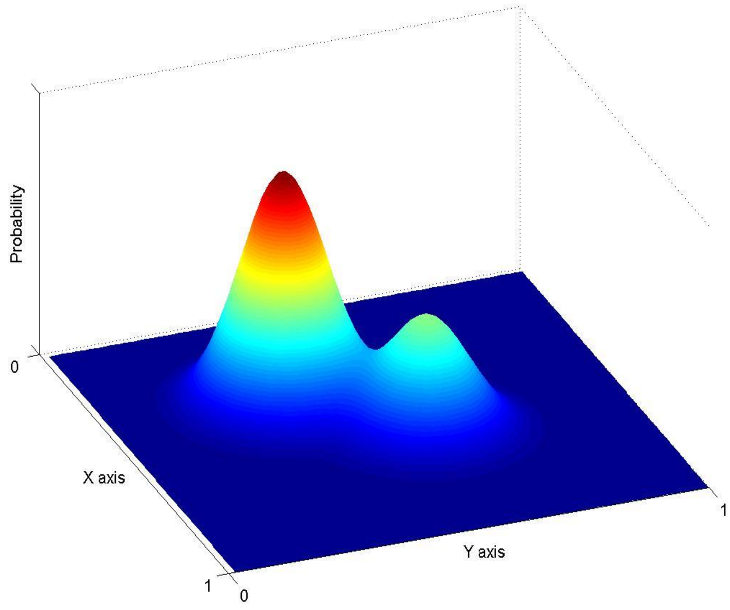
**<u>Assumption</u> : The data points are sampled from an underlying PDF**



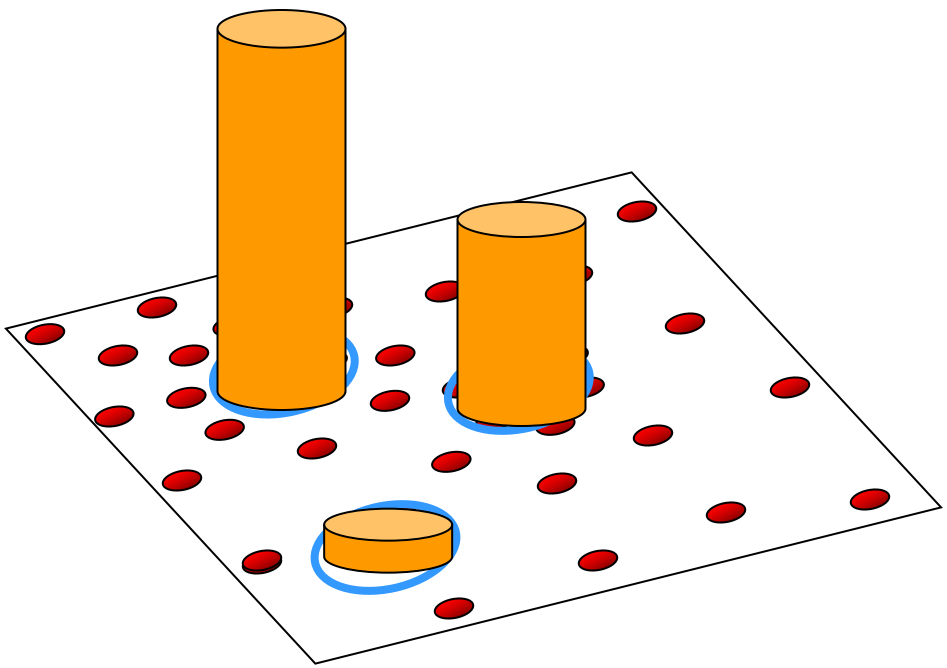

**Assumed Underlying PDF**

**Real Data Samples**

# Non-Parametric Density Estimation

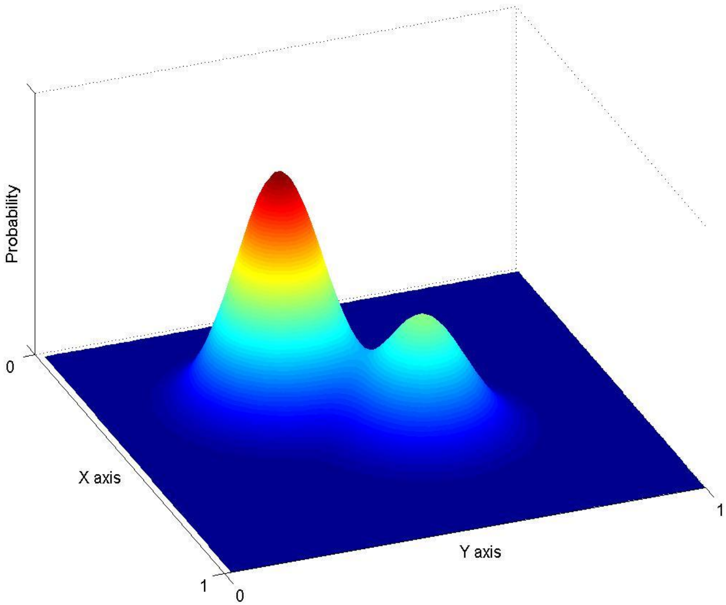



**Assumed Underlying PDF**

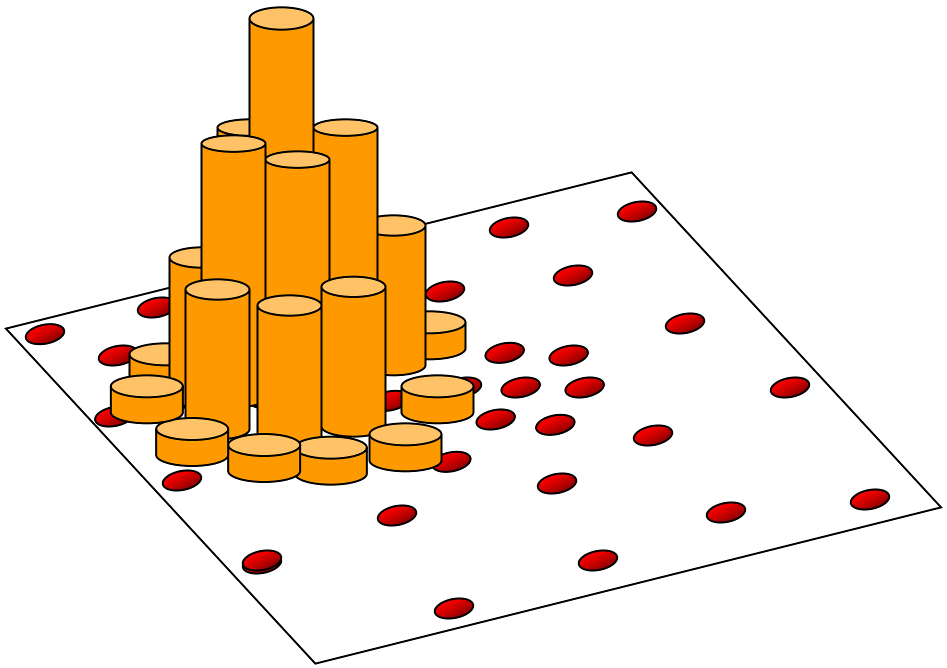


**Real Data Samples**

# Non-Parametric Density Estimation



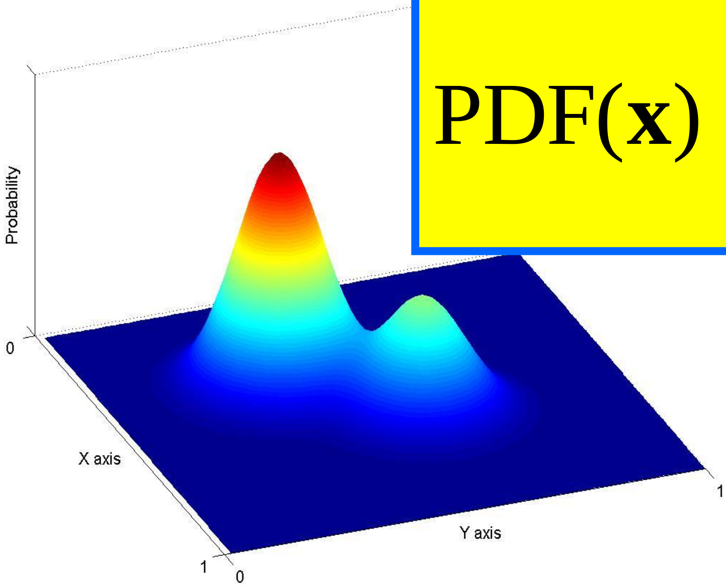

**Assumed Underlying PDF**



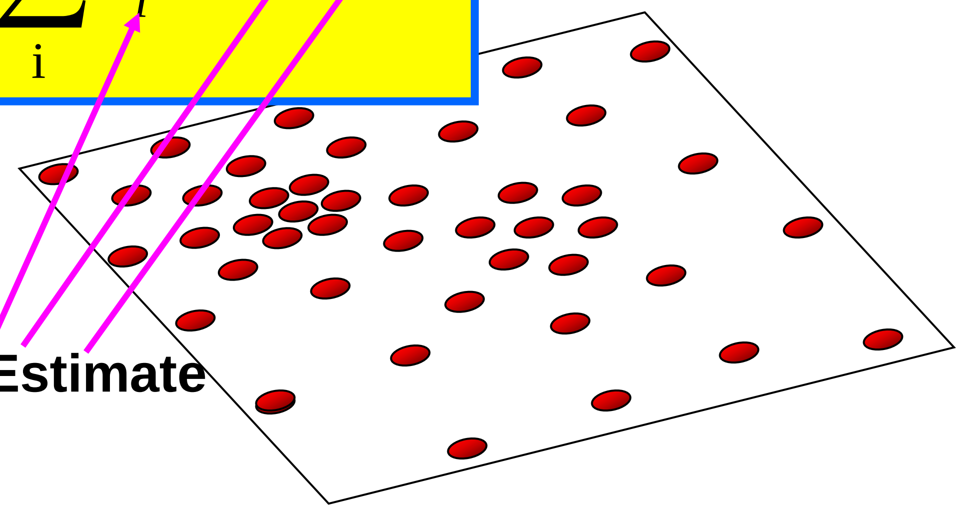
**Real Data Samples**

# *Parametric* Density Estimation

$$\mathrm{PDF}(\mathbf{x}) = \sum_{\mathrm{i}} c_i \cdot e^{-\frac{(\mathbf{x}-\mathbf{u}_i)^2}{2\sigma_i^2}}$$




Estimate



**Assumed Underlying PDF**

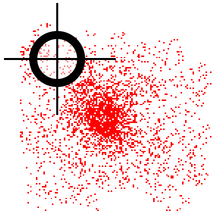
**Real Data Samples**

# Kernel Density Estimation

## Parzen Windows - Function Forms

$$P(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} K(\mathbf{x} - \mathbf{x}_i)$$

A function of some finite number of data points $x_1 \ldots x_n$



Data

In practice one uses the forms:

$$K(\mathbf{x}) = c\prod_{i=1}^{d} k(x_i)$$ or $$K(\mathbf{x}) = ck\left(\|\mathbf{x}\|\right)$$

Same function on each dimension

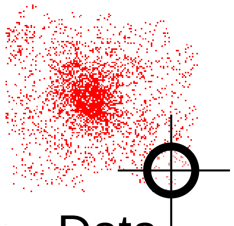
Function of vector length only

# Kernel Density Estimation

## Various Kernels

$$P(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} K(\mathbf{x} - \mathbf{x}_i)$$

A function of some finite number of data points $x_1 \ldots x_n$

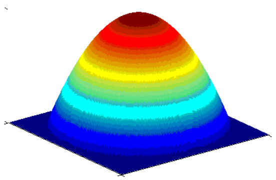


Data

Examples:

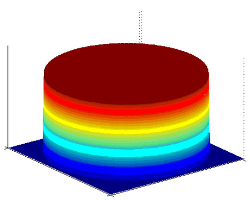
- Epanechnikov Kernel

$$K_E(\mathbf{x}) = \begin{cases} c\left(1 - \|\mathbf{x}\|^2\right) & \|\mathbf{x}\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$



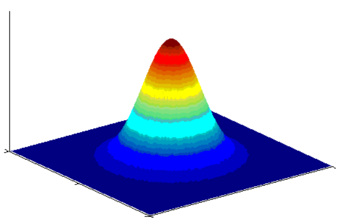
- Uniform Kernel

$$K_U(\mathbf{x}) = \begin{cases} c & \|\mathbf{x}\| \le 1 \\ 0 & \text{otherwise} \end{cases}$$



- Normal Kernel

$$K_N(\mathbf{x}) = c \cdot \exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right)$$



# Kernel Density $\underbrace{\text{Estimation}}_{\text{Gradient}}$

$$\nabla P(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \nabla K(\mathbf{x} - \mathbf{x}_i)$$

Give up estimating the PDF !
Estimate **<u>ONLY</u>** the gradient

Using the Kernel form:

$$K(\mathbf{x} - \mathbf{x}_i) = ck\left( \left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\|^2 \right)$$

Size of window

We get :

$$\nabla P(\mathbf{x}) = \frac{c}{n}\sum_{i=1}^{n} \nabla k_i = \frac{c}{n}\left[ \sum_{i=1}^{n} g_i \right] \bullet \left[ \frac{\sum_{i=1}^{n} \mathbf{x}_i g_i}{\sum_{i=1}^{n} g_i} - \mathbf{x} \right]$$

$$\mathrm{g}(\mathbf{x}) = -k'(\mathbf{x})$$

# Computing The Mean Shift

# Kernel Density Estimation

*Gradient*

$$\nabla P(\mathbf{x}) = \frac{c}{n}\sum_{i=1}^{n}\nabla k_i = \frac{c}{n}\left[\sum_{i=1}^{n} g_i\right]\bullet\left[\frac{\sum_{i=1}^{n}\mathbf{x}_i g_i}{\sum_{i=1}^{n} g_i} - \mathbf{x}\right]$$

$$\mathrm{g}(\mathbf{x}) = -k'(\mathbf{x})$$

# Computing The Mean Shift

$$\nabla P(\mathbf{x}) = \frac{c}{n}\sum_{i=1}^{n}\nabla k_i = \frac{c}{n}\left[\sum_{i=1}^{n} g_i\right]\bullet\left[\frac{\sum_{i=1}^{n}\mathbf{x}_i g_i}{\sum_{i=1}^{n} g_i} - \mathbf{x}\right]$$

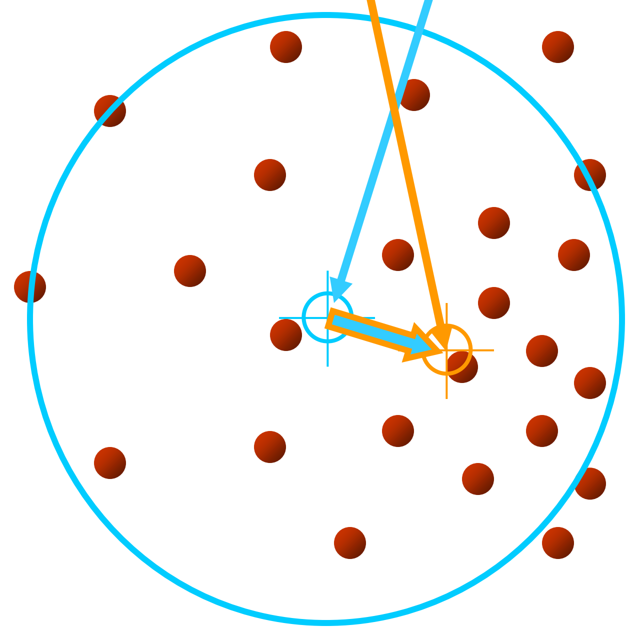
Yet another Kernel density estimation !

<u>Simple Mean Shift procedure</u>:

- Compute mean shift vector

$$\mathbf{m}(\mathbf{x}) = \left[\frac{\sum_{i=1}^{n}\mathbf{x}_i g\left(\frac{\left\|\mathbf{x}-\mathbf{x}_i\right\|^2}{h}\right)}{\sum_{i=1}^{n} g\left(\frac{\left\|\mathbf{x}-\mathbf{x}_i\right\|^2}{h}\right)} - \mathbf{x}\right]$$

•Translate the Kernel window by **m(x)**



$$g(\mathbf{x}) = -k'(\mathbf{x})$$

# 算法流程

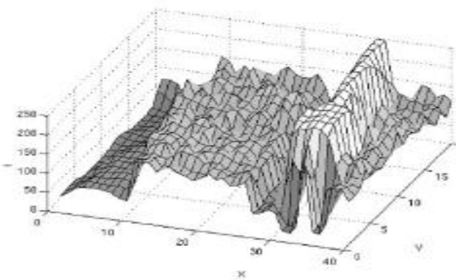
（1）计算m(x)；

（2）如果|| m(x) - x||小于一个给定的阈值,结束循环;否则,将my(x)赋给x，继续执行(1)。

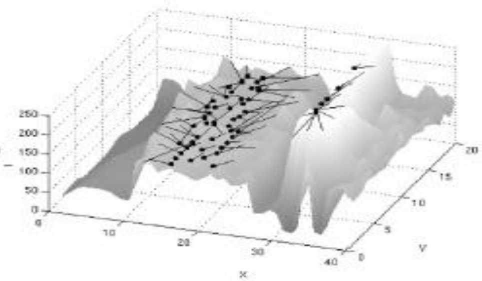
# 算法流程（聚类）

1. For each $j = 1 \ldots n$ run the mean shift procedure for $\mathbf{x}_j$ and store the convergence point in $\mathbf{z}_j$.
2. Identify clusters $\{\mathbf{C}_p\}_{p=1\ldots m}$ of convergence points by linking together all $\mathbf{z}_j$ which are closer than 0.5 from each other in the joint domain.
3. For each $j = 1 \ldots n$ assign $L_j = \{p \mid \mathbf{z}_j \in \mathbf{C}_p\}$.
4. Optional: Eliminate spatial regions smaller than $M$ pixels.

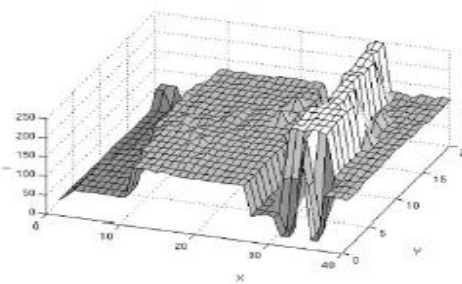
# 实验结果



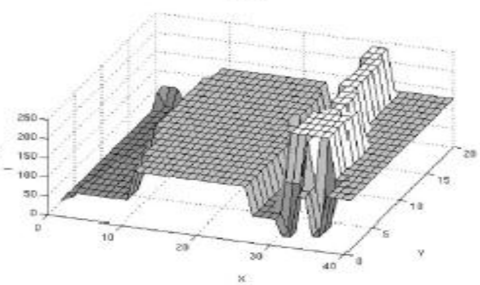
(a)



(b)



(c)



(d)

# 实验结果

(a)

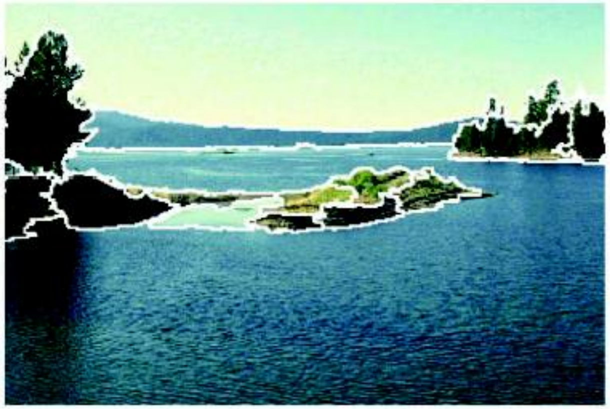
(b)

(c)

# 实验结果

(a) (b)

# 实验结果




# Graph Cut（图割）

基本思想：

1. 将图像用图的方式表示，顶点表示像素，边表示像素之间的关系。图像分割对应图的割集。

2. 确定图中边的权值，使图像分割目标（能量最小化）对应图的最小割。

3. 用最大流算法求解最小割问题。

# 图论的相关知识

- 图（Graph）：

  由点集和边集构成的集合 *G*=（*V*，*E*）。

*V*------点集

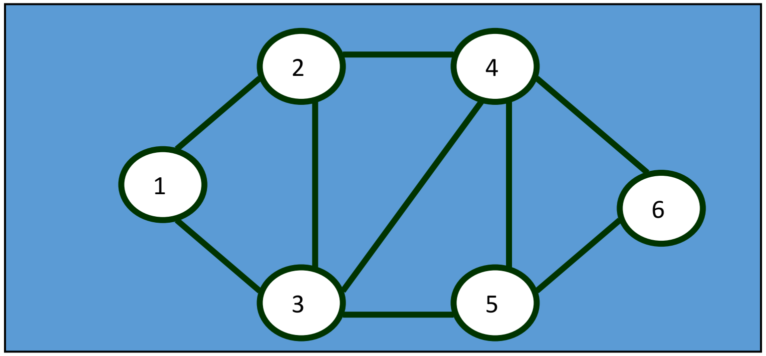
*E*------边集

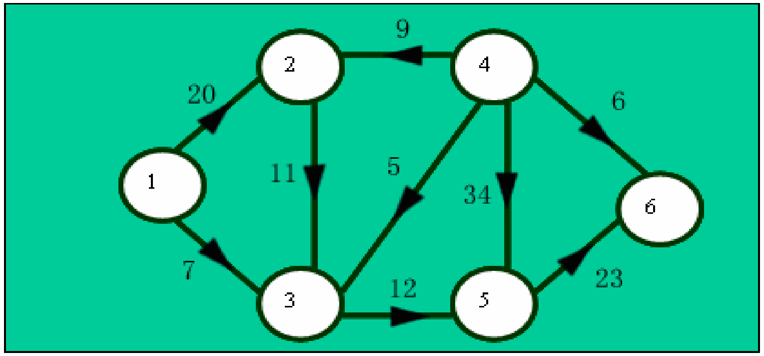
赋权图：每条边赋有一个权值*W*（*p*， *q*）

有向图：每条边有方向

有向赋权图： 每条边既有方向又有一个权值

无向赋权图……

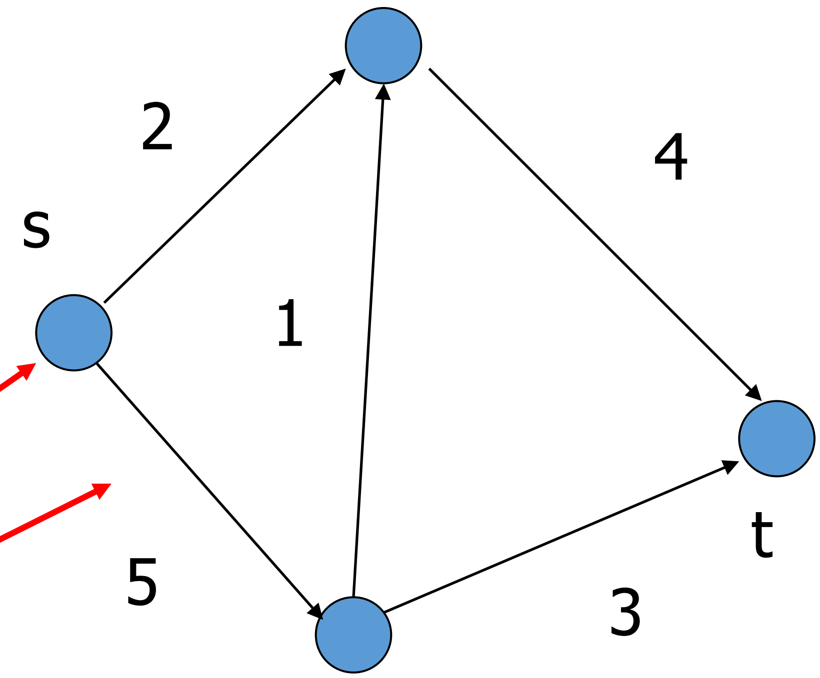


# S-T图

- 有源节点（s）和终节点（t）
- 每条边有一个非负的容量Cap(i,j)
- 对于不存在的边，其容量为0

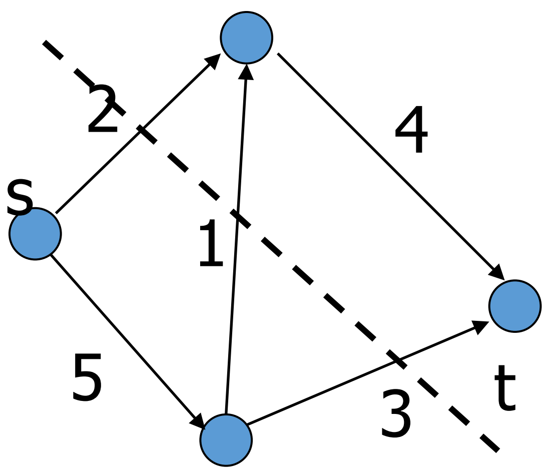
$$G = \{V, E\}$$



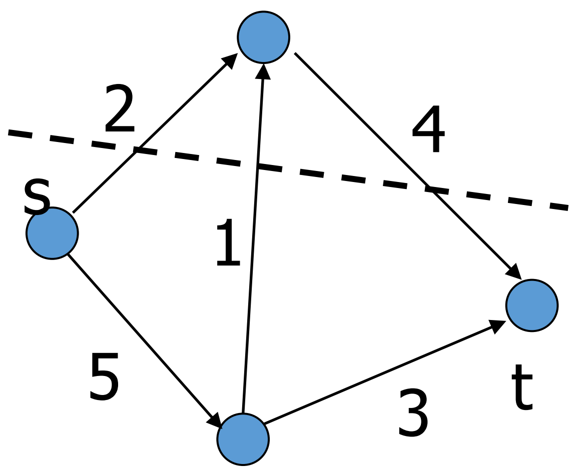
# S-T割

割：将s-t图分成两个子集$S$和$T$

s-t割：当且仅当s属于$S$，t属于$T$



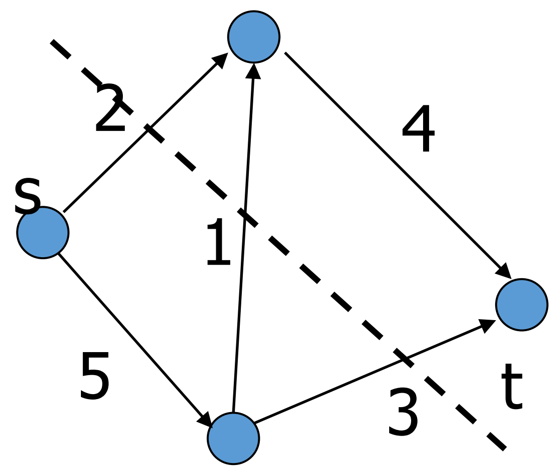

s-t割




非s-t割

# 最小割

$$Cap([S,T]) = \sum_{\{i,j\}\in E, i\in S, j\in T} Cap(i,j)$$




s-t图中容量最小的s-t割

# 图割在图像分割中的应用

1. Normalized cut及其在图像分割中的应用
2. 用graph cut算法求解计算机视觉中的能量极小化问题

# 1. Normalized cut及其在图像分割中的应用

（1）图论在分类问题中的应用

（2）Normalized cut

（3）求解最小Normalized cut的近似算法

（4）Normalized cut在图像分割中应用

# (1) 图论在分类问题中的应用

一般的分类问题：

- ■给定一个点集V，按照一定的相似度量（距离）寻求一种划分，将点集V划分成不相交的若干子集合$V_1, V_2 \cdots V_m$。使得每一子集内部的相似度尽量高，而子集之间的相似度尽量低。

两个问题：

- 什么是最优划分准则？
- 有没有有效算法？

# (1) 图论在分类问题中的应用

用图论的方法来解决分类问题：

- ■在给定点集V中的每一点对i、j之间，建立一条边（i，j），给这条边赋权$w_{ij}$相似度量。这样就建立了一个无向赋权图。

对于这样的图我们可以用邻接矩阵来表示：

$$\begin{bmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{bmatrix}$$

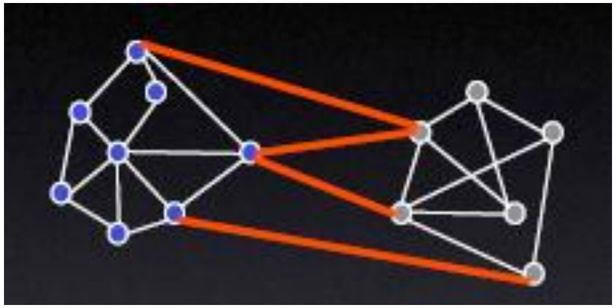
# (1) 图论在分类问题中的应用

考虑二分类问题

将点集V分成不相交的两部分A、B。则两类别之间的相似性我们可以用图割来度量。

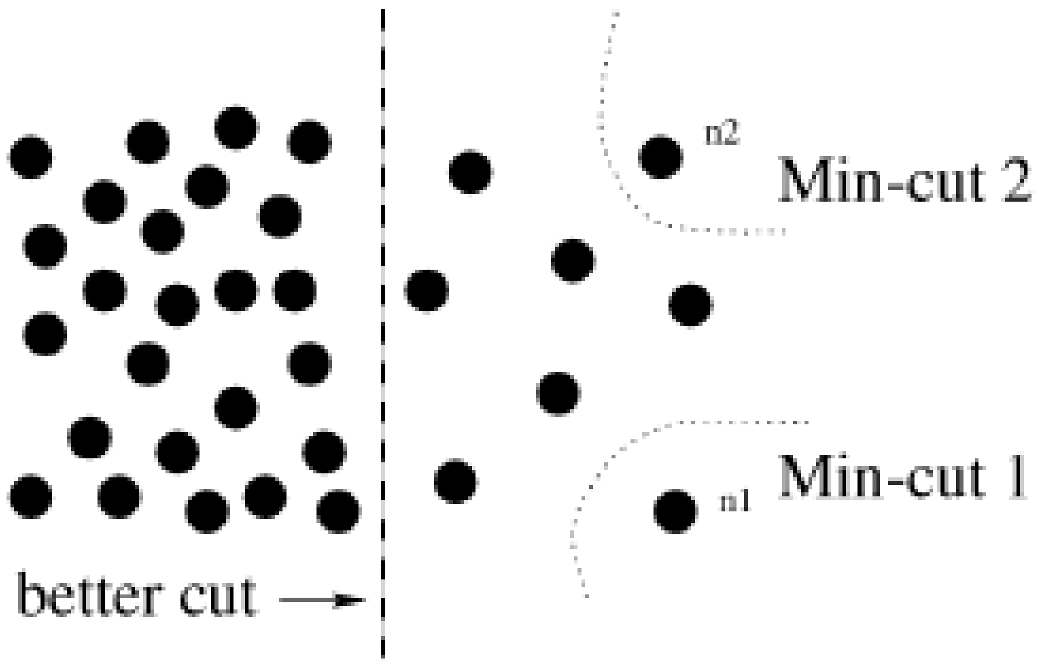
$$cut(A,B) = \sum_{u \in A, v \in B} w(u,v)$$

显然最优划分应使$cut(A,B)$取最小值。



# (1) 图论在分类问题中的应用

需要注意的是：仅考虑用割集的权值之和来度量两个集合之间的相关性，会容易出现孤立分割的问题。如下图所示：



# (2) Normalized cut

一个解决上述问题的办法是通过定义新的类间相似性度量。Normalized cut(*Ncut*):

$$Ncut(A,B)=\frac{cut(A,B)}{assoc(A,V)}+\frac{cut(A,B)}{assoc(B,V)}$$

这里 $assoc(A,V)=\sum_{u\in A,t\in V} w(u,t)$

这样包含孤立点的*Ncut*值不会小。

再定义总的类内相似性度量

$$Nassoc(A,B)=\frac{assoc(A,A)}{assoc(A,V)}+\frac{assoc(B,B)}{assoc(B,V)}$$

# (2) Normalized cut

通过简单的推导可以证明

$$Ncut(A,B)=\frac{cut(A,B)}{assoc(A,V)}+\frac{cut(A,B)}{assoc(B,V)}=2-Nassoc(A,B)$$

可见最小化类间相似性和最大化总类内相似性是等价的。
这样就解决了划分准则的问题，即

最优划分对应于最小$Ncut$

通过求最小$Ncut$，就可以得到最优划分。

下一个问题是有没有求最小$Ncut$的有效算法？

# （3）求解最小*Ncut*的近似算法

不幸的是求一个图的*Ncut*是一个NP问题。

但是通过精巧的构造，可以通过求解如下广义特征值问题来得到最小Normalized cut近似解。

令$W=\{w_{ij}\}$

$$D=diag(d(1),d(2),\dots,d(N)),\quad d(i)=\sum_j w_{ij}$$

则广义特征值问题 $(D-W)x=\lambda Dx$的次小特征值是最小*Ncut*对应的实数解。该特征值所对应的特征向量对应于最优划分。

# (3) 求解最小*Ncut*的近似算法

- 特征向量离散化

　　由于我们需要特征向量是仅含有不同符号的两个值，故还需要对所得特征向量做离散化处理。即需要选择一个分界点。有两种方法：

（1）取中点

（2）取0

- 多分类问题

　　首先用次小特征值所对应特征向量进行二分类。然后用再次小特征值所对应的特征向量对已分好的两类再次细分。或者在每个分好的类别中，重复用上述算法进行分类。

# 具体算法

① 给定一个点集，构建图G(V,E)，边的权为对应两端点的相似度。

② 求解$(D-W)x=\lambda Dx$的特征值及其所对应的特征向量。

③ 用次小特征值所对应的特征向量进行二分类。

④ 若需再分，则在每个分好的类别中重复上述过程。否则终止。

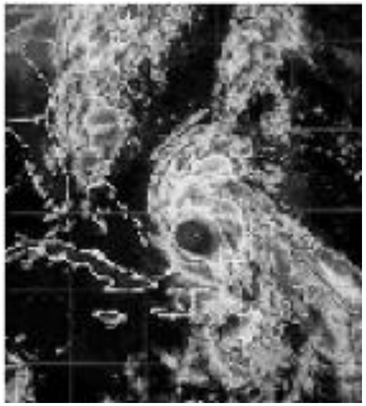
# (4) Normalized cut 在图像分割中应用

将一幅图像上所有像素点看作点集V，每两个点之间都建立一条边，得到边集E。为每条边按下面方法赋权

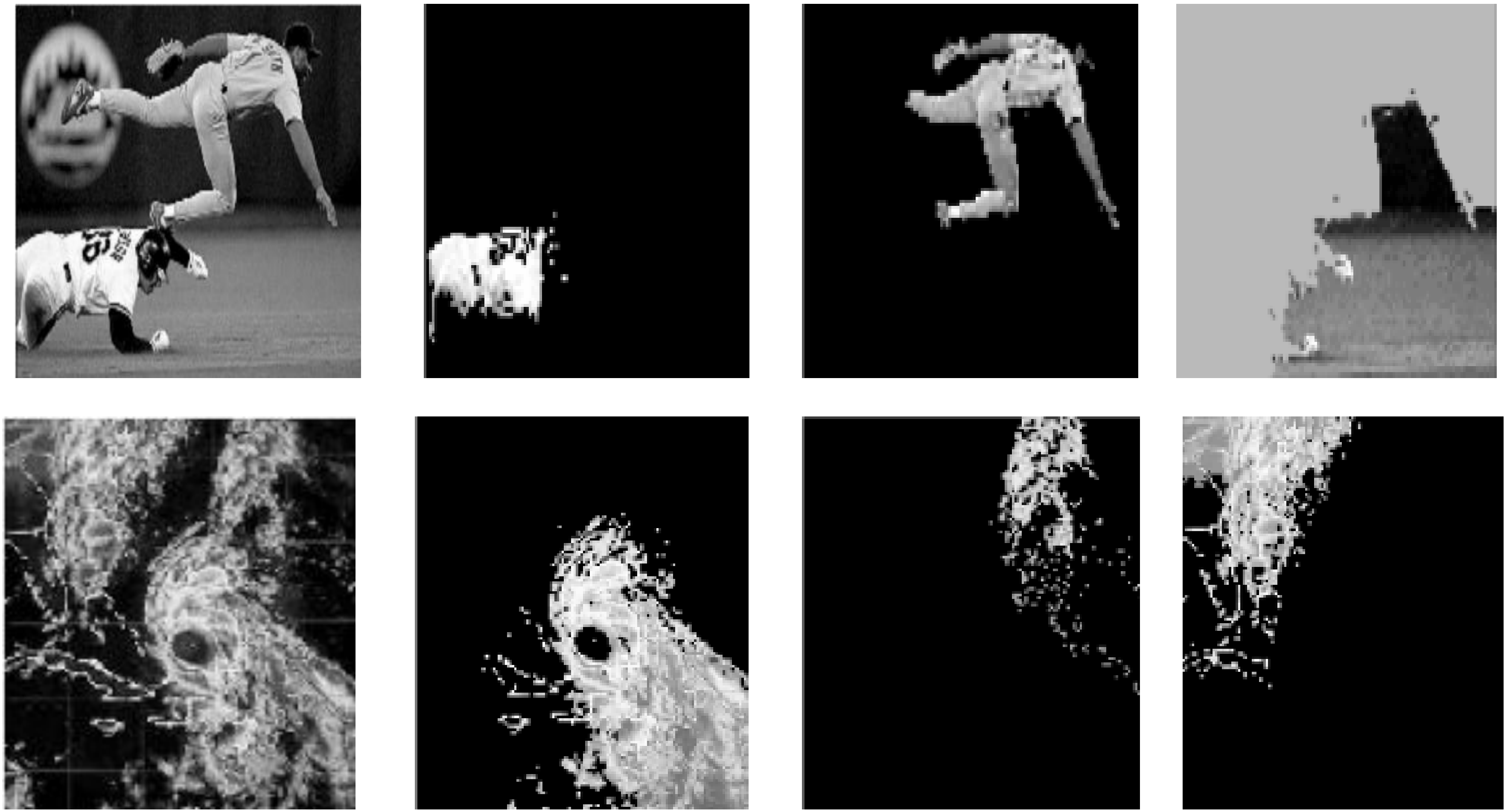
$$w_{ij} = e^{-\frac{||F_i - F_j||^2}{\sigma_I^2}} \times \begin{cases} e^{-\frac{||X_i - X_j||^2}{\sigma_X^2}}, & if \quad ||X_i - X_j|| < r \\ 0, & else \end{cases}$$

这样就建立一个赋权无向图$G=(V, E)$

按照前述算法，我们就可以完成对该幅图像分割操作。



# 实验结果



# 2. 用graph cut求解能量极小化问题

（1）计算机视觉中的多标记问题

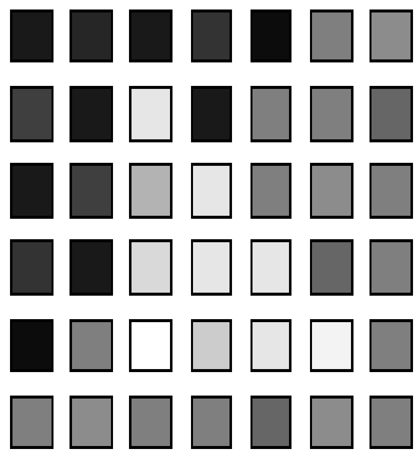
（2）多标记问题的能量极小化模型

（3）运用Graph cut算法求解这类问题

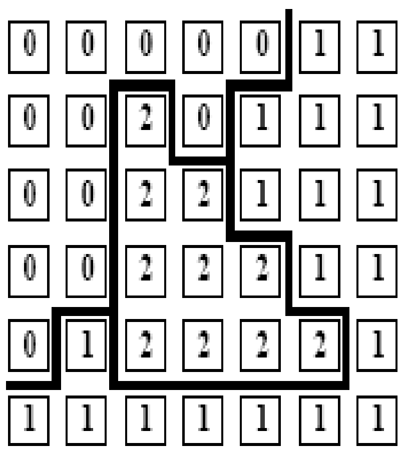
# （1）计算机视觉中的多标记问题

计算机视觉的很多问题可以看作一个最优标记（labeling）问题，如：

- 图像分割
- 图像恢复
- 立体视觉
- 三维重建



(a) An image




(b) A labeling

# （1） 计算机视觉中的多标记问题

同样面临的两个问题

- 什么是最优标记准则？
- 有没有有效求解算法？

# （2）多标记问题的能量极小化模型

我们可以通过构造一个如下的能量函数来得到最优标记准则。

$$E(f)=E_{data}(f)+E_{smooth}(f)=\sum_{p\in P}D_p(f_p)+\sum_{\{p,q\}\in N}V_{pq}(f_p,f_q)$$

f: P→L 的映射。P是像素点集，L是标记集。

Data项表示给每个像素点赋予标记（label）的代价

Smooth项表示每两个相邻的像素分别赋予标记$f_p$和$f_q$的代价

# （2）多标记问题的能量极小化模型

$$E(f) = E_{data}(f) + E_{smooth}(f) = \sum_{p \in P} D_p(f_p) + \sum_{\{p,q\} \in N} V_{pq}(f_p, f_q)$$

我们希望最优标记应使得能量函数取最小值，即可建立下面最优化模型

$$\min E(f)$$
$$\text{s.t. } f_p \in L$$

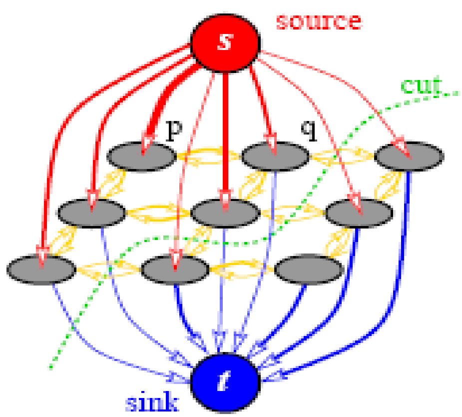
# (3)运用Graph cut算法求解这类问题

- Graph cut与能量函数的对应关系
- 运用Graph cut算法求解能量极小化问题
  - 两标记问题
  - 多标记问题

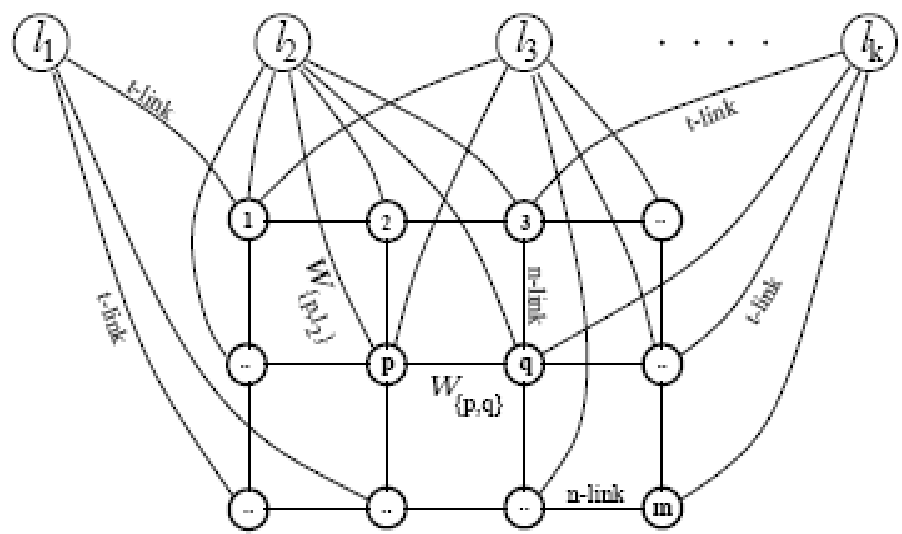
# Graph cut与能量函数的对应关系

能量函数：

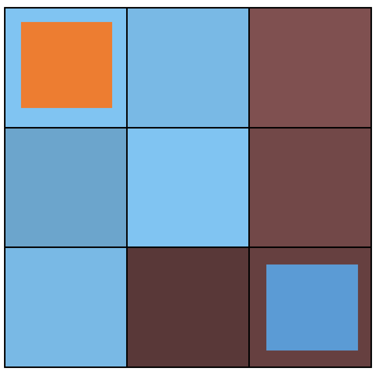
$$E(f) = E_{data}(f) + E_{smooth}(f) = \sum_{p \in P} D_p(f_p) + \sum_{\{p,q\} \in N} V_{pq}(f_p, f_q)$$

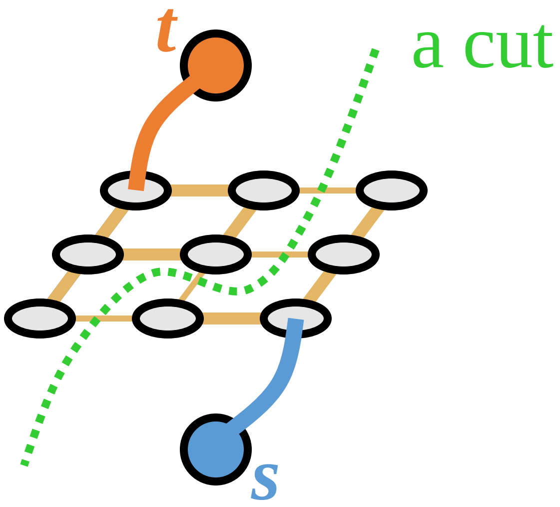



(b) A cut on $\mathcal{G}$



# Graph cut与能量函数的对应关系



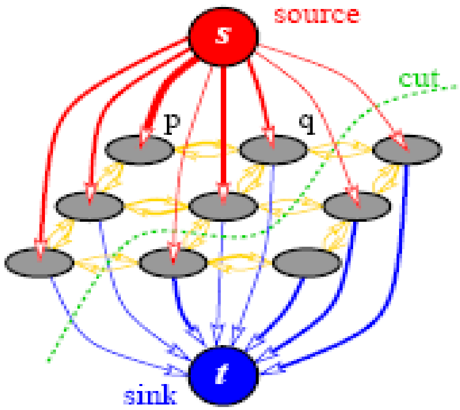



$$E(f) = E_{data}(f) + E_{smooth}(f) = \sum_{p \in P} D_p(f_p) + \sum_{\{p,q\} \in N} V_{pq}(f_p, f_q)$$

# 运用Graph cut算法求解能量极小化问题

两标记问题：

对于两标记问题，最小能量对应于图的最小割。图论中已有经典的算法，可以求得一个图的最小割，从而得到极小能量。



(b) A cut on $\mathcal{G}$

# 运用Graph cut算法求解能量极小化问题

多标记问题：

当标记数量大于2时，已经证明该问题是NP-hard问题。故很难求得该问题的全局极小值。

Boykov等构造了两个运用最小割求解该类能量函数的近似极小值的算法：

α－β swap

α expansion

这两个算法运算速度快，且能得到比较好的结果，从而得到了广泛应用，并使得用能量极小化模型和图割来处理计算机视觉中的一些问题成为目前的一个研究热点。

# 提纲

- 概述
- 早期的图像分割方法
- 基于特定理论的方法
- 基于深度神经网络的图像分割

# Fully convolutional networks（ FCN ）

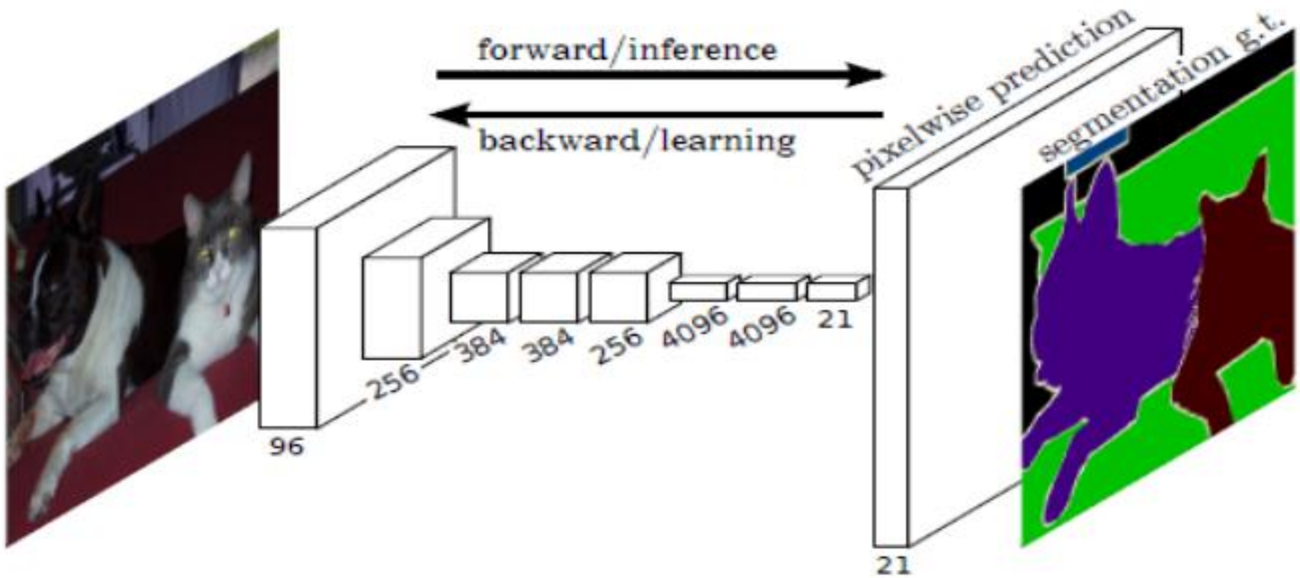
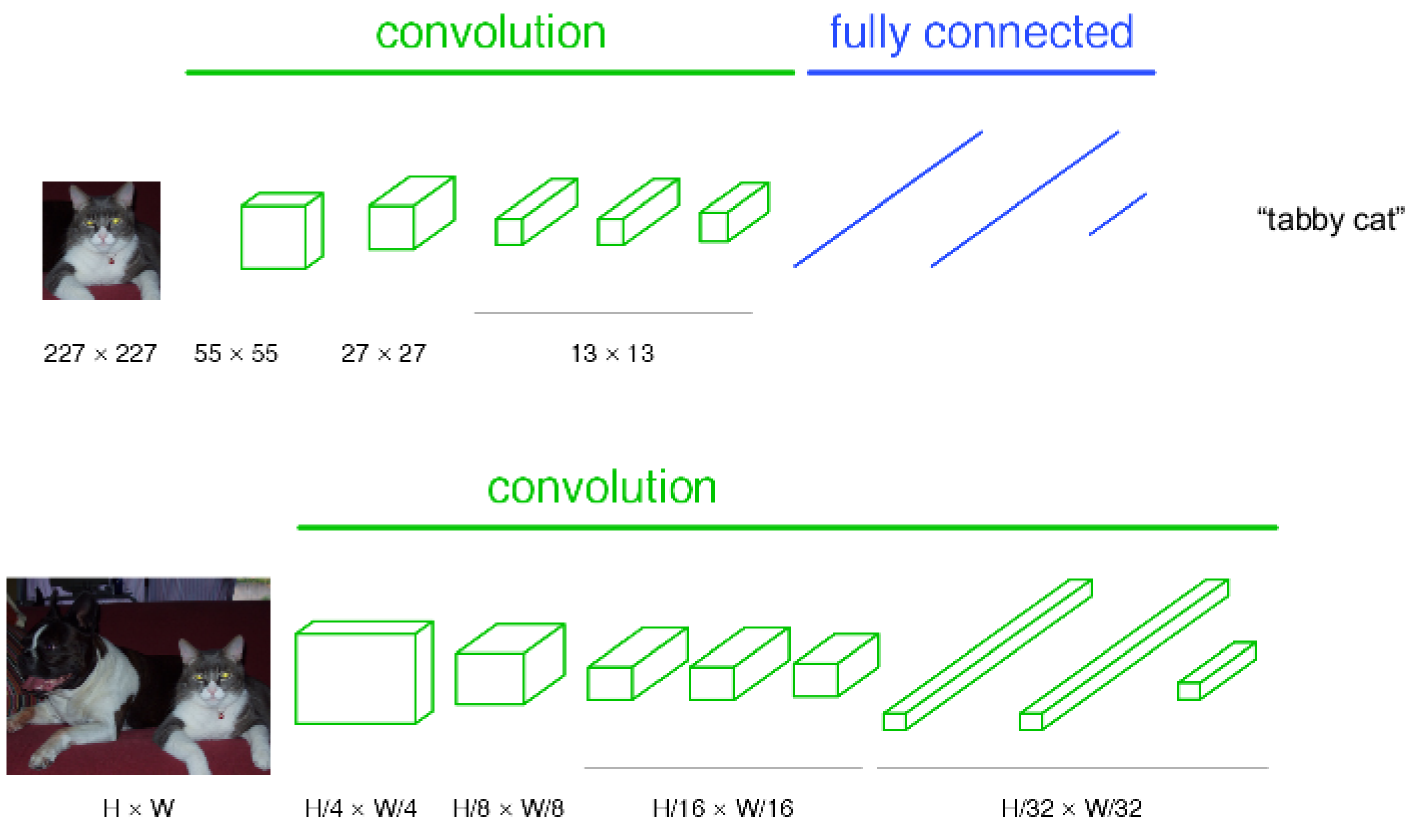



图1 全卷积网络能有效学习对每个像素做出dense prediction，比如语义分割

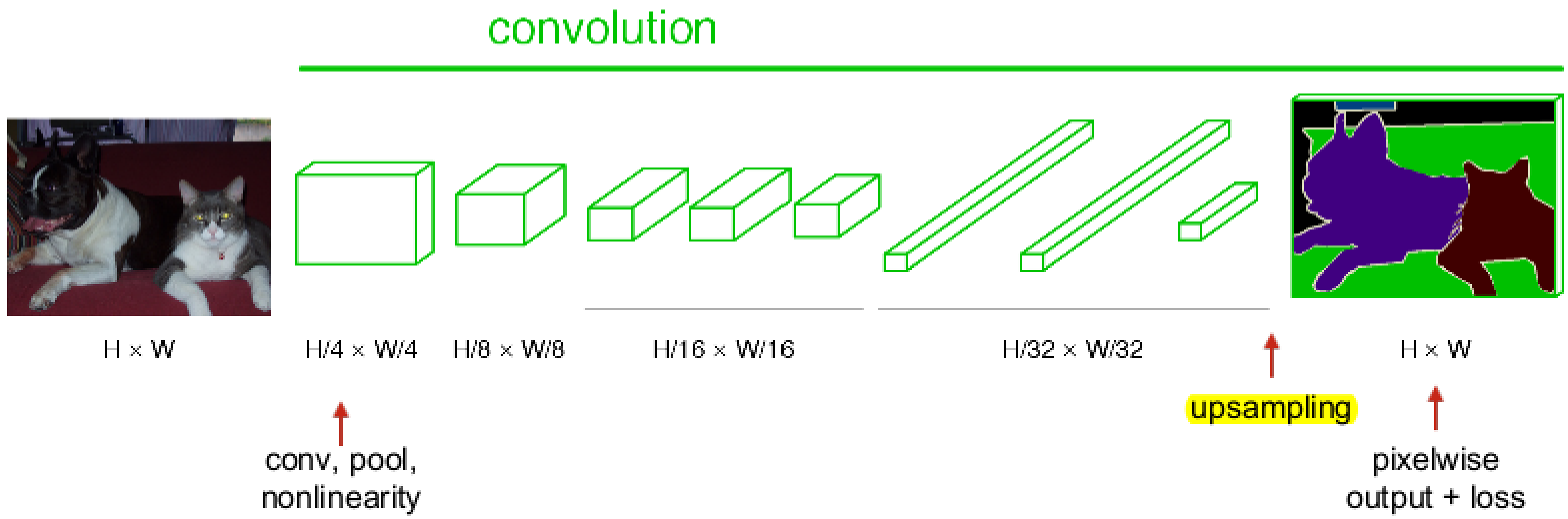
Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation, CVPR, 2015

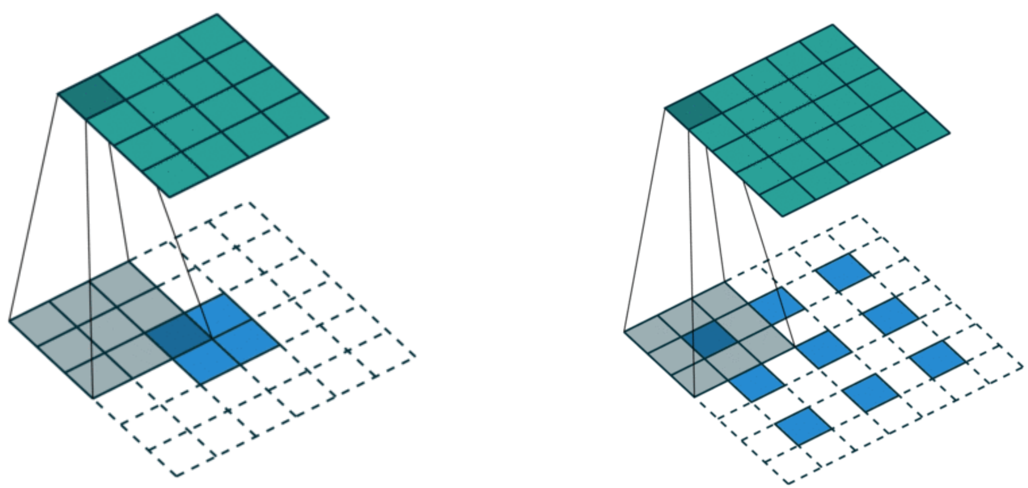
# 卷积化


convolution
fully connected
"tabby cat"
227 × 227
55 × 55
27 × 27
13 × 13
convolution
H × W
H/4 × W/4
H/8 × W/8
H/16 × W/16
H/32 × W/32

# 上采样

上采样（Upsampling）：增大图像尺寸



# Skip结构

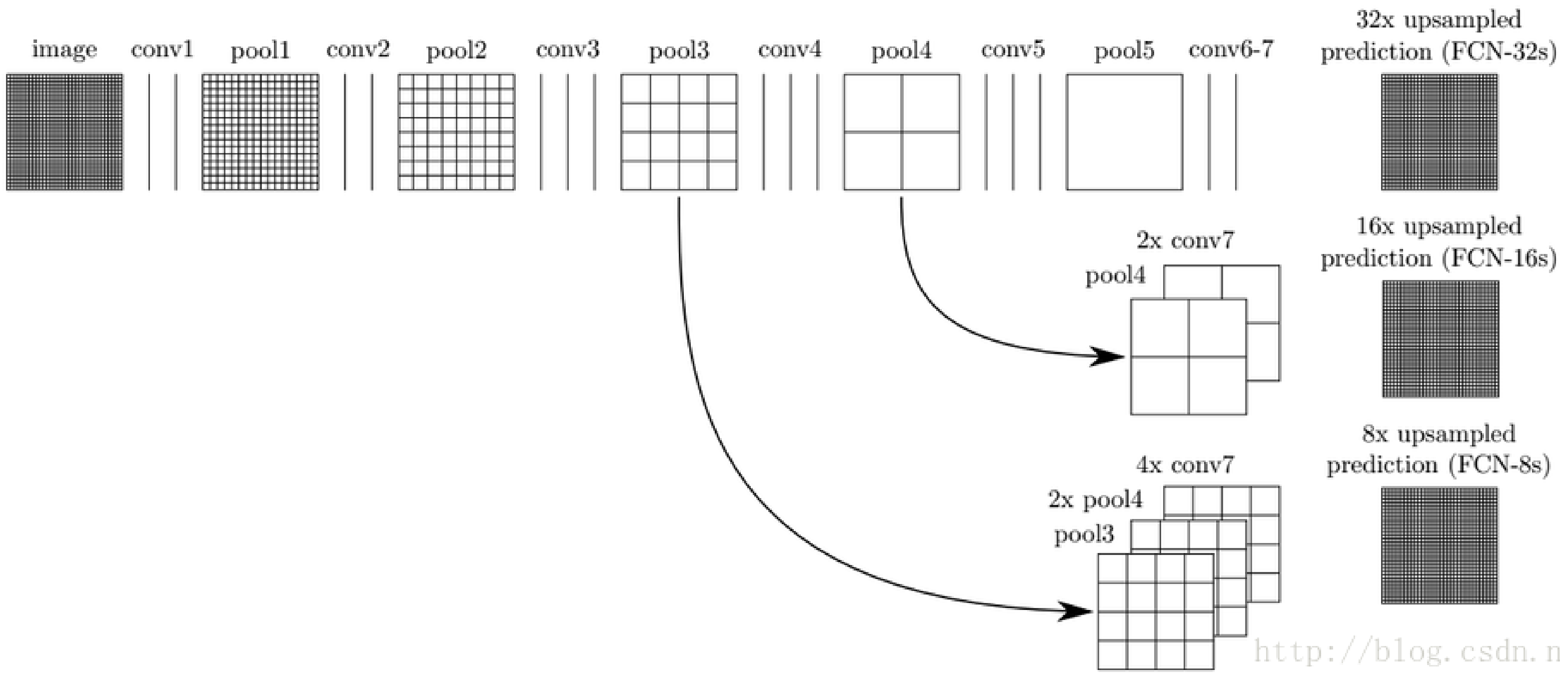
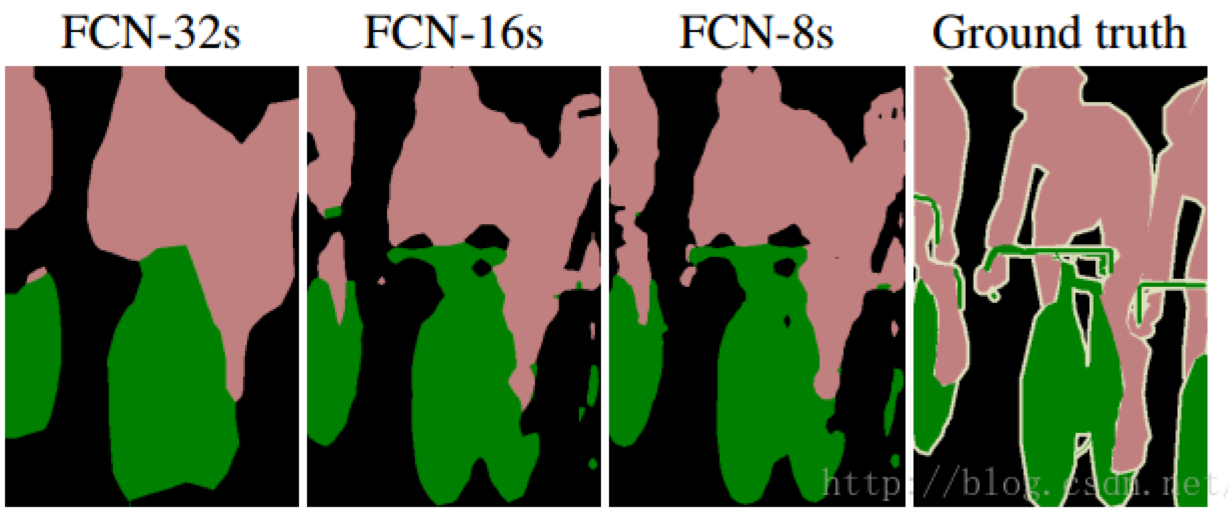
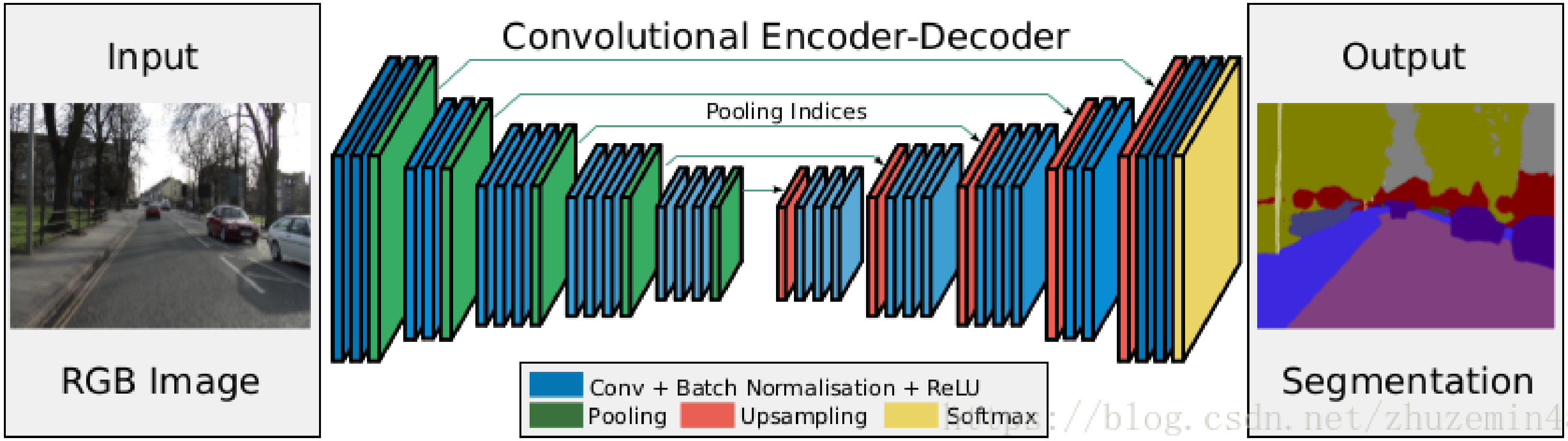


image
conv1
pool1
conv2
pool2
conv3
pool3
conv4
pool4
conv5
pool5
conv6-7
32x upsampled prediction (FCN-32s)
2x conv7
pool4
16x upsampled prediction (FCN-16s)
4x conv7
2x pool4
pool3
8x upsampled prediction (FCN-8s)

# 实验结果



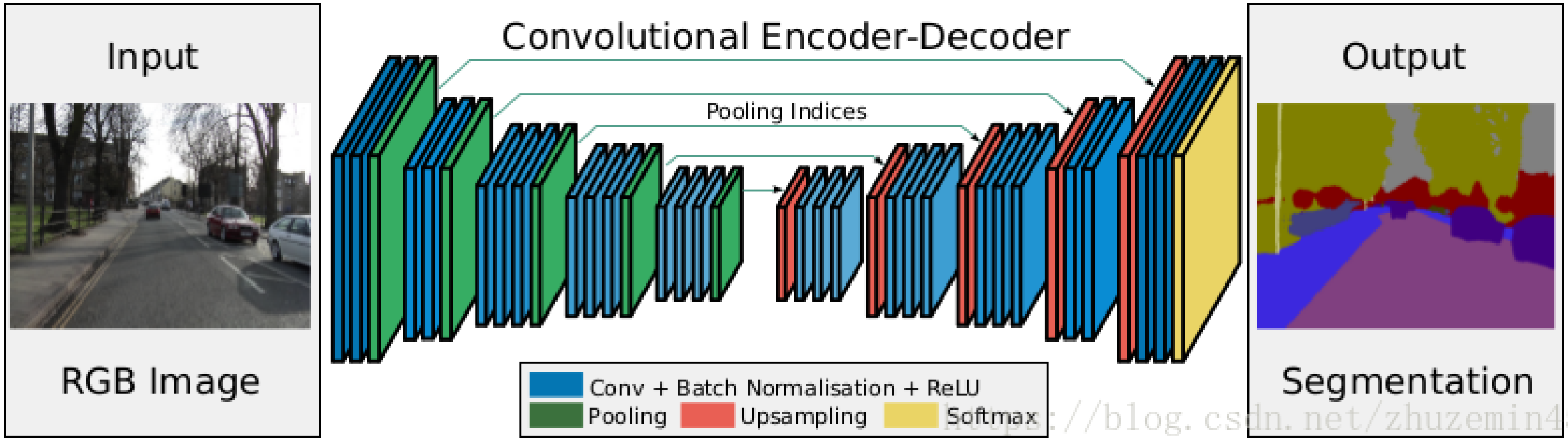
# SegNet



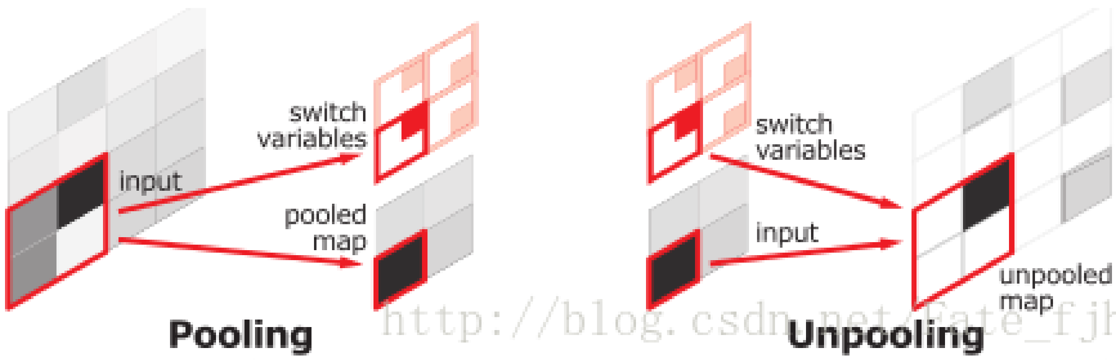


Badrinarayanan V , Kendall A , Cipolla R . SegNet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017

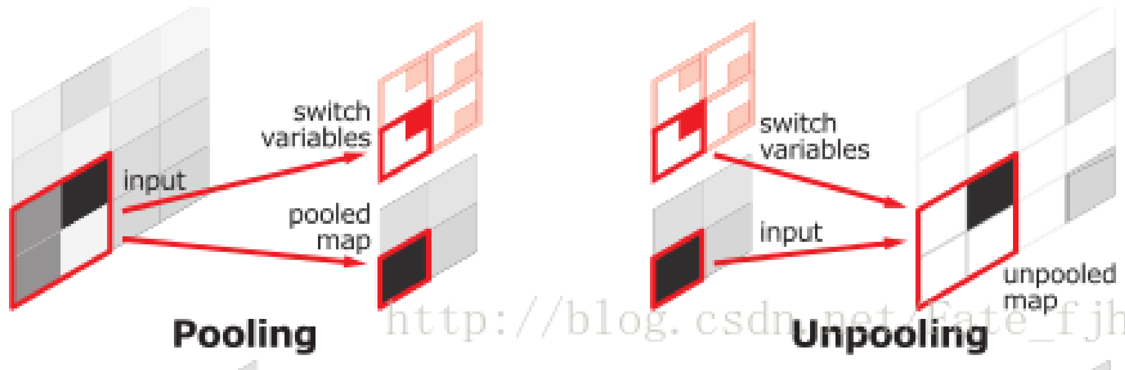
# 上采样（Pooling&Upsampling）

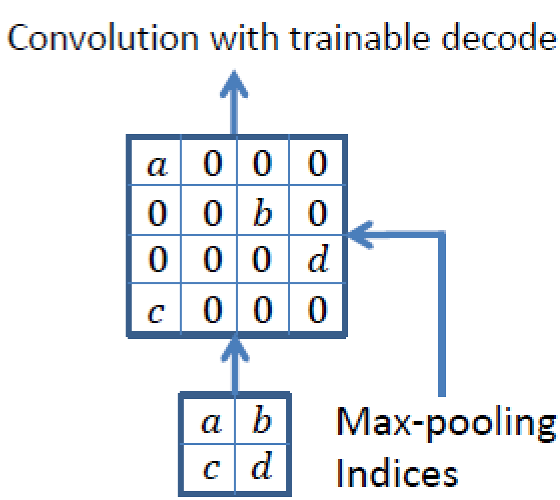




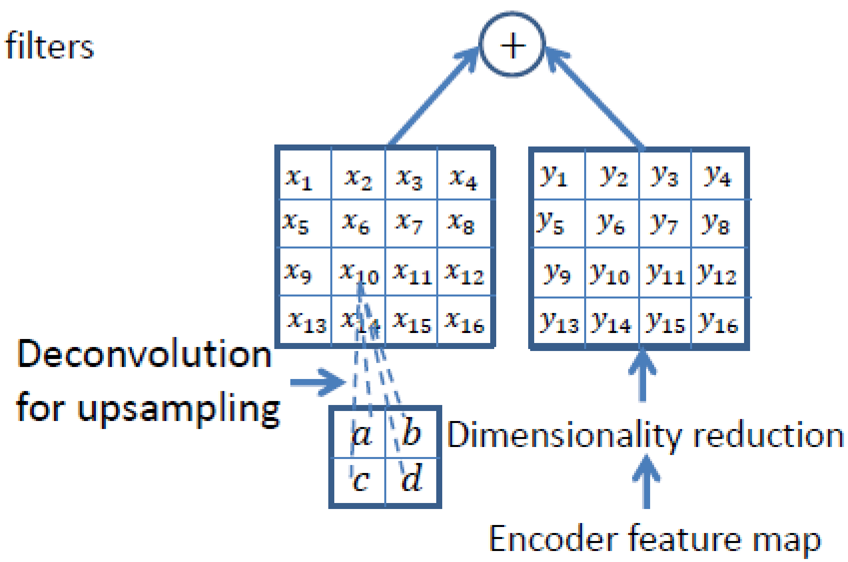


# 上采样（Pooling&Upsampling）



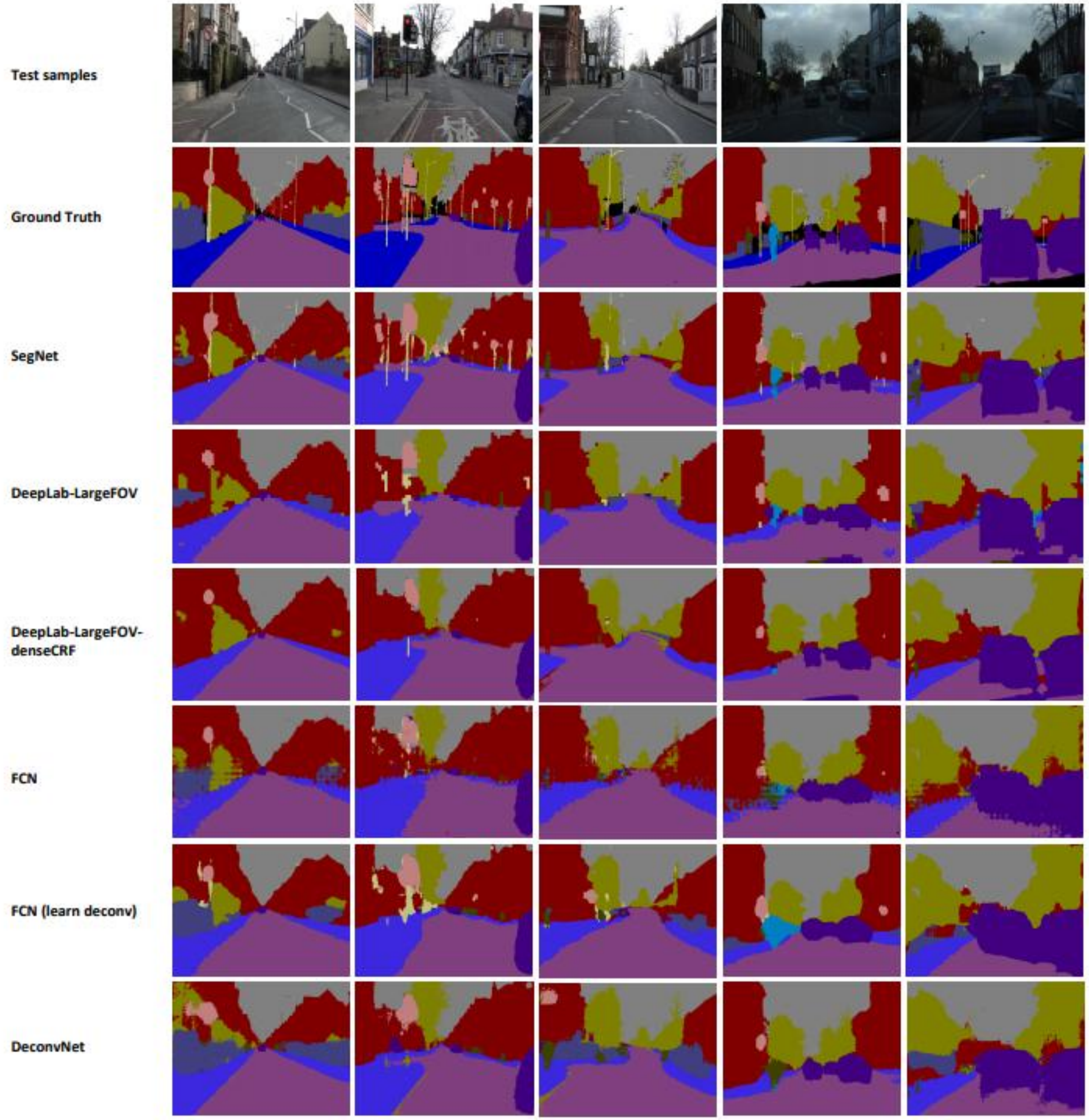




SegNet

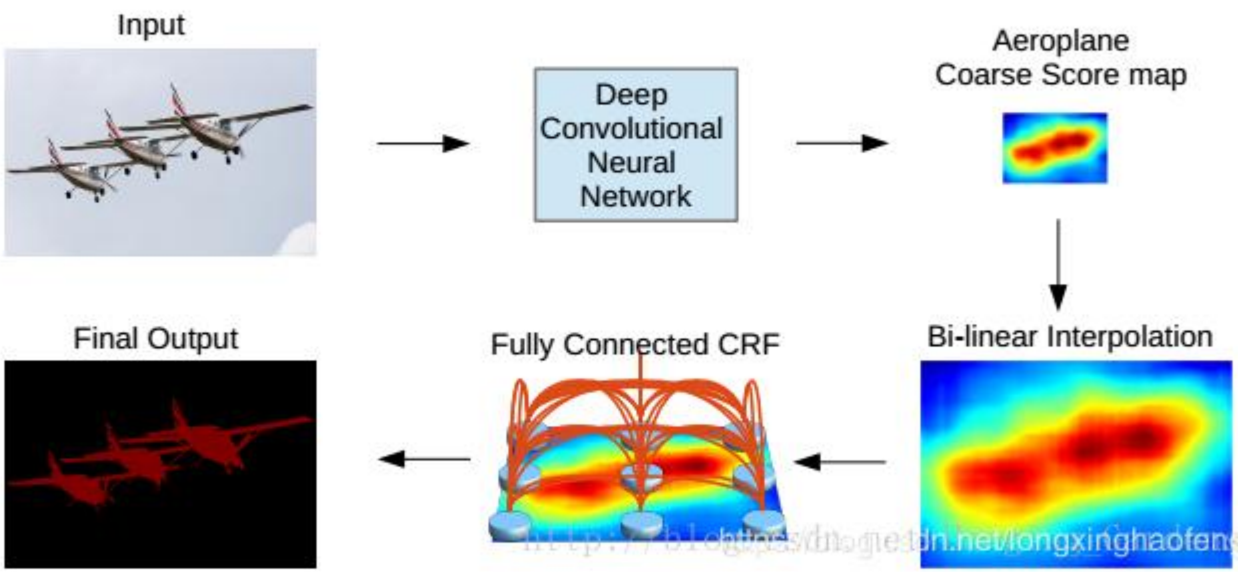



FCN

# 对比实验


Test samples
Ground Truth
SegNet
DeepLab-LargeFOV
DeepLab-LargeFOV-denseCRF
FCN
FCN (learn deconv)
DeconvNet

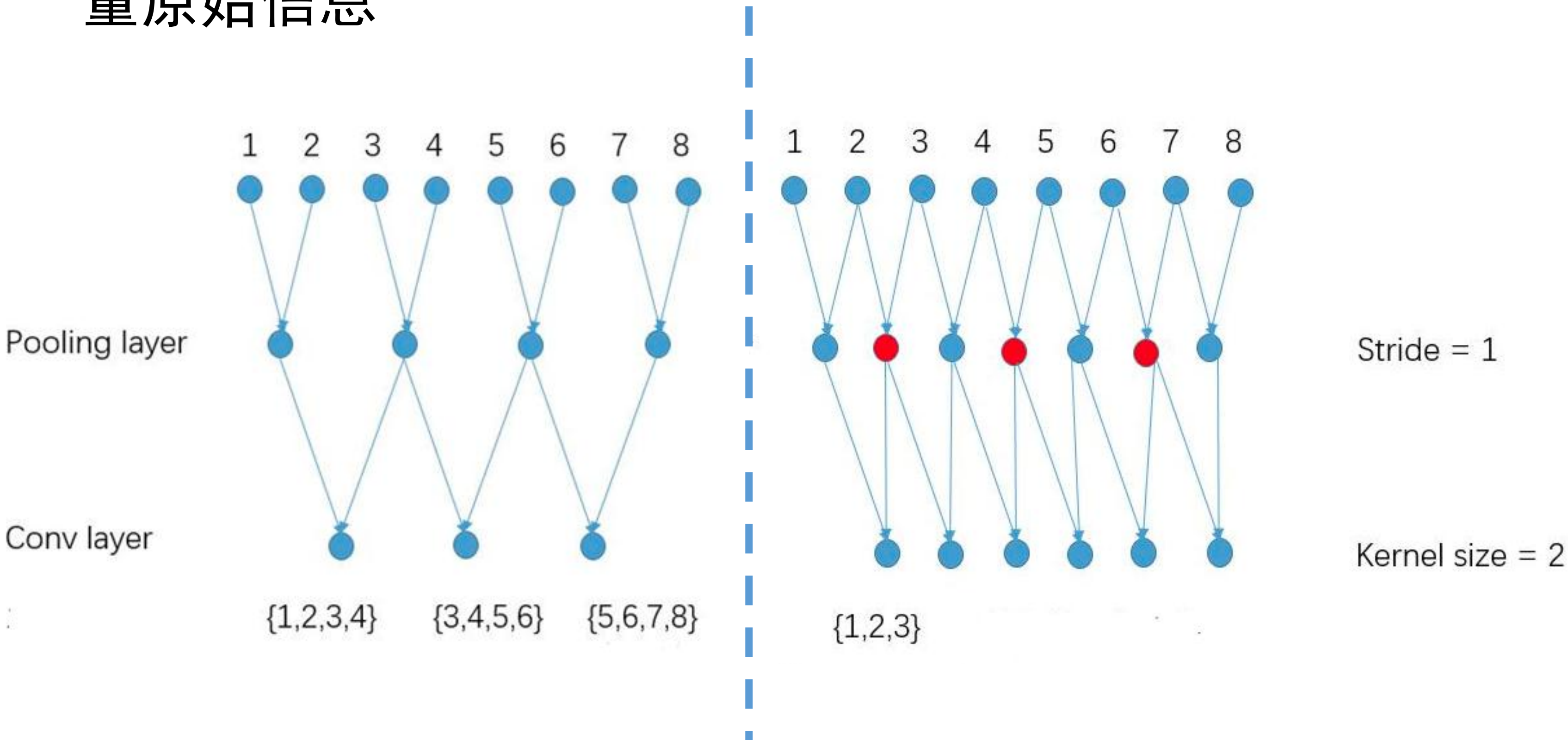
# DeepLab (v1,v2,v3)




Chen L C , Papandreou G , Kokkinos I , et al. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2016, 40(4):834-848.

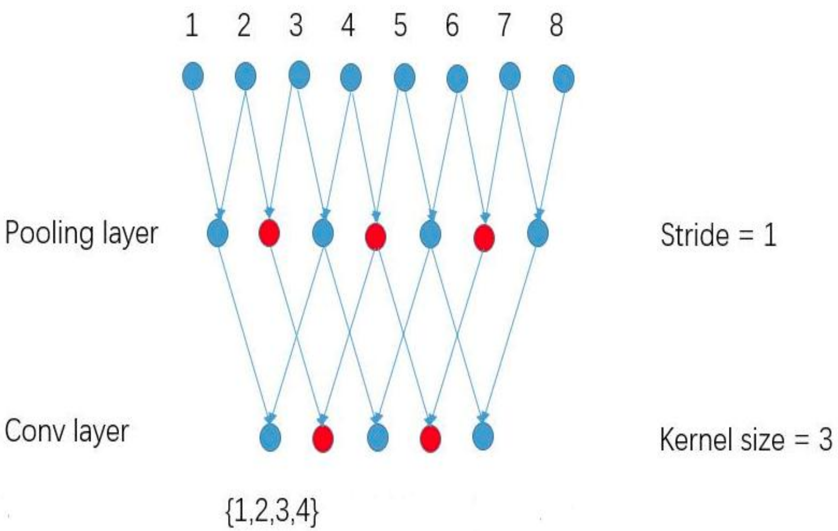
# DeepLab (v1)

不断下采样往往导致输入特征尺寸不断减小，进而损失大量原始信息

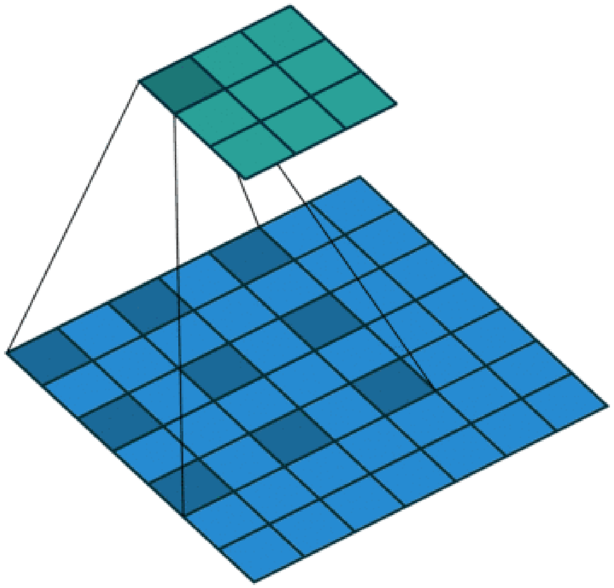



感受野变小了

# 卷积与空洞卷积

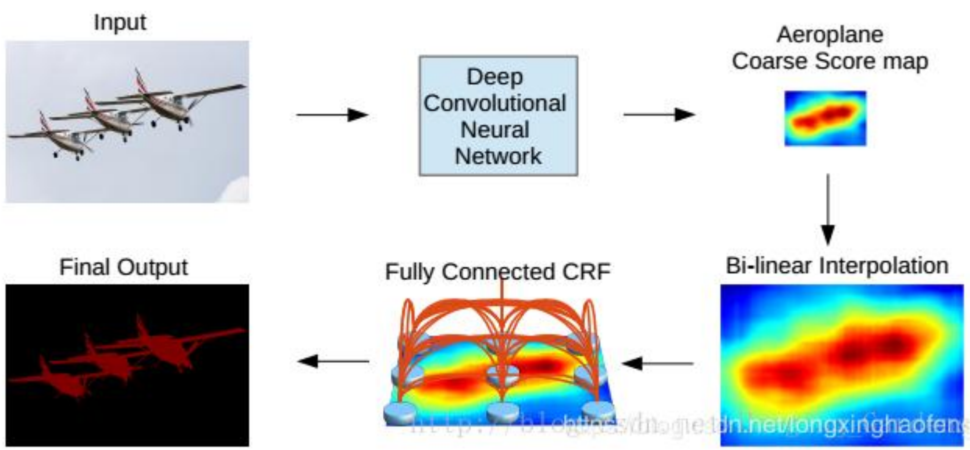



一维空洞卷积

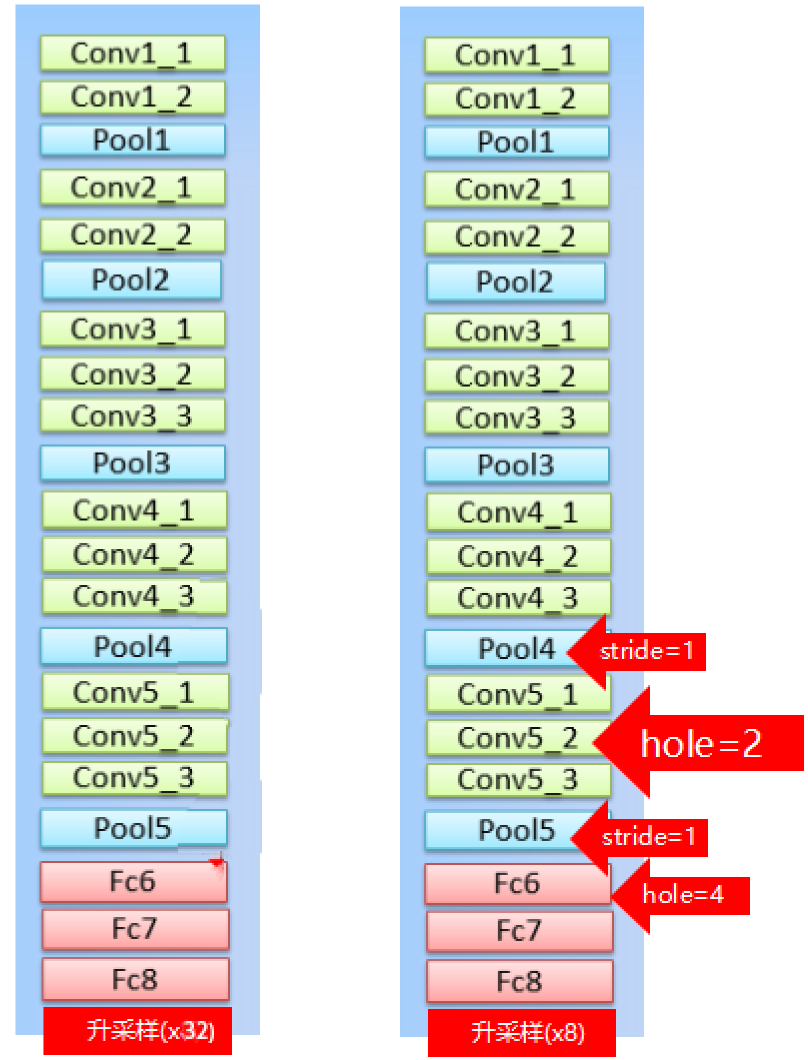


二维空洞卷积

# DeepLab (v1)



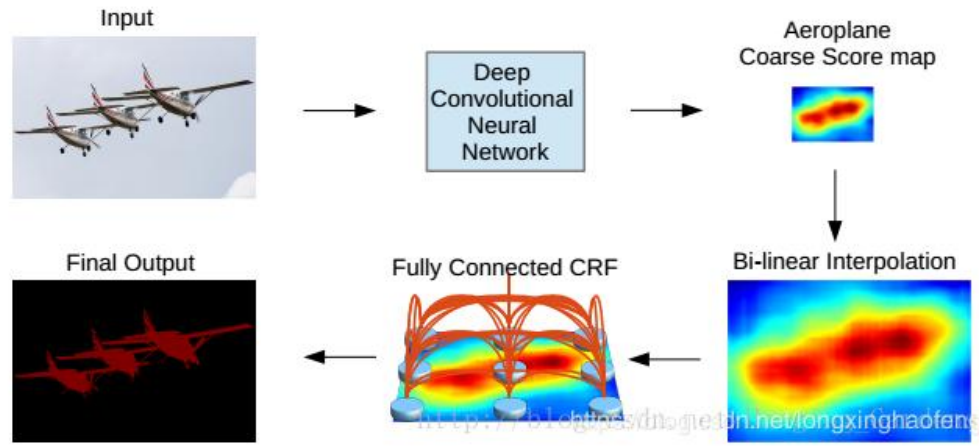
Input
Deep Convolutional Neural Network
Aeroplane Coarse Score map
Bi-linear Interpolation
Fully Connected CRF
Final Output

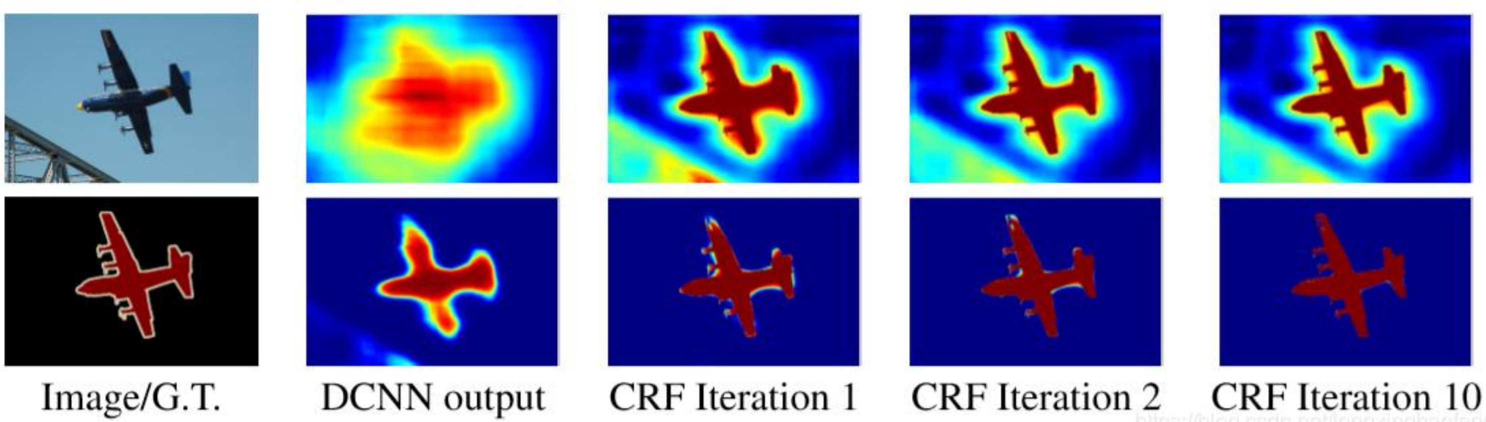



Conv1_1
Conv1_2
Pool1
Conv2_1
Conv2_2
Pool2
Conv3_1
Conv3_2
Conv3_3
Pool3
Conv4_1
Conv4_2
Conv4_3
Pool4
Conv5_1
Conv5_2
Conv5_3
Pool5
Fc6
Fc7
Fc8
升采样(x32)
Conv1_1
Conv1_2
Pool1
Conv2_1
Conv2_2
Pool2
Conv3_1
Conv3_2
Conv3_3
Pool3
Conv4_1
Conv4_2
Conv4_3
Pool4
stride=1
Conv5_1
Conv5_2
hole=2
Conv5_3
Pool5
stride=1
Fc6
hole=4
Fc7
Fc8
升采样(x8)

# Fully connected CRF

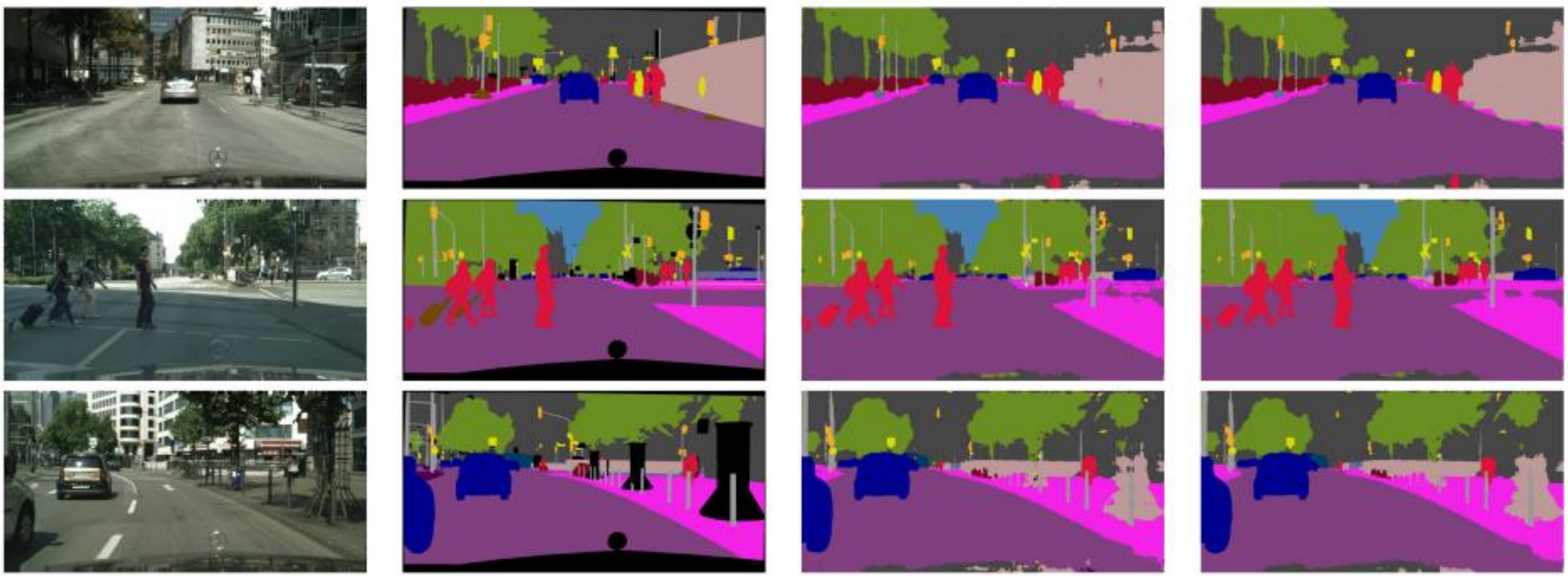
Conditional Random Field (CRF，条件随机场)



# 实验结果

课后练习：了解DeepLab v2 v3



(a) Image (b) G.T. (c) Before CRF (d) After CRF

# 课后练习与参考文献

课后练习：

1. 试使用C或C++实现基于Mean Shift算法的图像平滑。
2. 复现SegNet网络并进行图像分割验证实验。

参考文献：


1. Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(11):1222–1239, November 2001.
2. Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation, CVPR, 2015.
3. Badrinarayanan V , Kendall A , Cipolla R . SegNet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.
4. Chen L C , Papandreou G , Kokkinos I , et al. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2016, 40(4):834-848.

# 小节

- 早期的图像分割方法
- 基于特定理论的方法
  - Mean Shift
  - Normalized Cut、Graph Cut
- 基于深度神经网络的图像分割
  - FCN
  - SegNet
  - DeepLab

谢谢