

# 图像分割技术介绍

原创： AI人工智能平台

**SIGAI推荐**

SIGAI知识库论文解读小视频升级上线

支持向量机1-从线性分类器到最大化分类间隔

支持向量机2-拉格朗日对偶

支持向量机3-线性不可分的问题

支持向量机4-核映射与核函数

支持向量机5-KKT条件的使用

Generative Adversarial Nets论文解读

【含讲解视频、讲解文档、PPT以及源代码】

Playing Atari with Deep Reinforcement Learning论文解读

【含讲解视频、讲解文档、PPT】

【含讲解视频、讲解文档、PPT以及源代码】

详见: [www.sigai.cn](http://www.sigai.cn) 知识库

图像分割 (image segmentation) 技术是计算机视觉领域的一个重要的研究方向, 是图像语义理解的重要一环。图像分割是指将图像分成若干具有相似性质的区域的过程, 从数学角度来看, 图像分割是将图像划分成互不相交的区域的过程。近些年来随着深度学习技术的逐步深入, 图像分割技术有了突飞猛进的发展, 该技术相关的场景物体分割、人体前背景分割、人脸人体Parsing、三维重建等技术已经在无人驾驶、增强现实、安防监控等行业都得到广泛的应用。



图像分割技术从算法演进历程上, 大体可划分为基于图论的方法、基于像素聚类的方法和基于深度语义的方法这三大类, 在不同的时期涌现出了一批经典的分割算法。

### 基于图论的分割方法

此类方法基于图论的方法利用图论领域的理论和方法, 将图像映射为带权无向图, 把像素视作节点, 将图像分割问题看作是图的顶点划分问题, 利用最小剪切准则得到图

像的最佳分割。此类方法把图像分割问题与图的最小割（MIN-CUT）[1]问题相关联，通常做法是将待分割的图像映射为带权无向图 $G=(V, E)$ ，其中， $V=\{v_1, \dots, v_n\}$ 是顶点的集合， $E$ 为边的集合。图中每个节点 $N \in V$ 对应于图像中的每个像素，每条边 $\in E$ 连接着一对相邻的像素，边的权值 $w(v_i, v_j)$ ，其中 $(v_i, v_j) \in E$ ，表示了相邻像素之间在灰度、颜色或纹理方面的非负相似度。而对图像的一个分割 $S$ 就是对图的一个剪切，被分割的每个区域 $C \in S$ 对应着图中的一个子图。

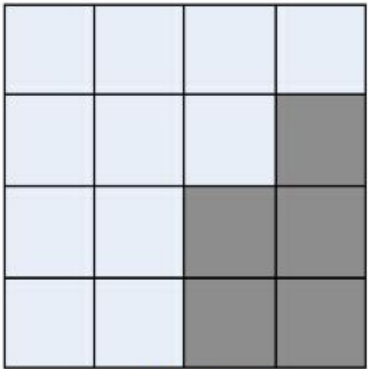
分割的原则就是使划分后的子图在内部保持相似度最大，而子图之间的相似度保持最小。我们以一个两类的分割为例，把 $G = (V, E)$ 分成两个子集 $A, B$ ，另： $A \cup B = V, A \cap B = \emptyset$ ,

$$CUT(A, B) = \sum_{\mu \in A, v \in B} w(\mu, v)$$

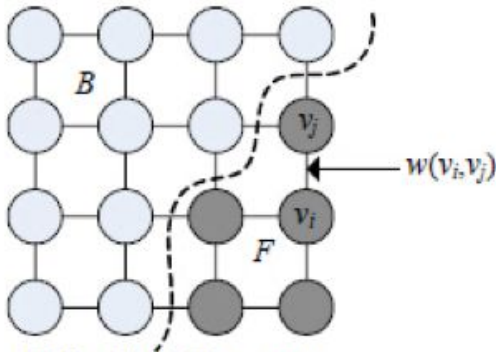
, 其中,

$$w(\mu, v)$$

是权重(weight), 最小割就是让上式的值最小的分割。



原始图像

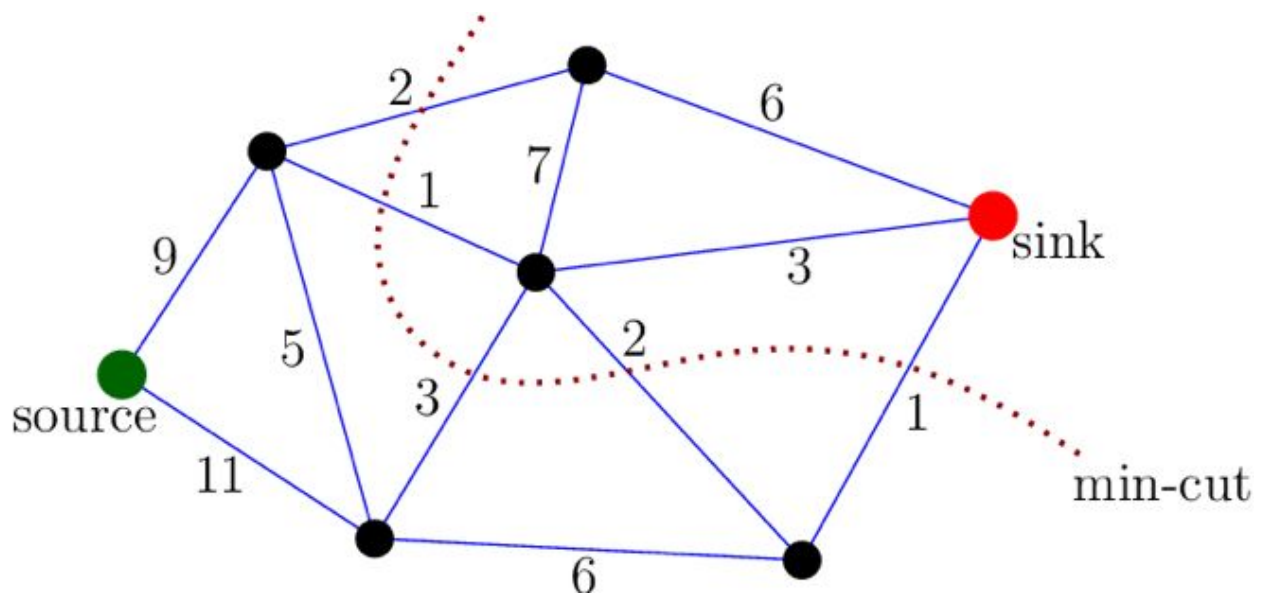


图像对应的图 $G=(V, E)$

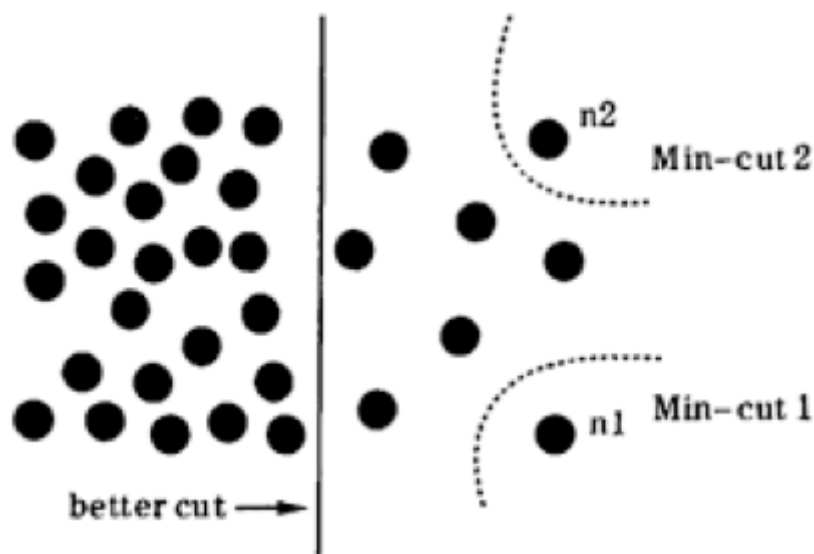
基于图论的代表有NormalizedCut，GraphCut和GrabCut等方法

想要理解Normalized Cut 需要先理解什么是分割(CUT)与最小化分割（MIN-CUT），我们拿下图做例子，我们把图一看成一个整体 $G$ ，现在需要把它分成两个部分。显然

中间红色虚线切割的边就是最小化分割。



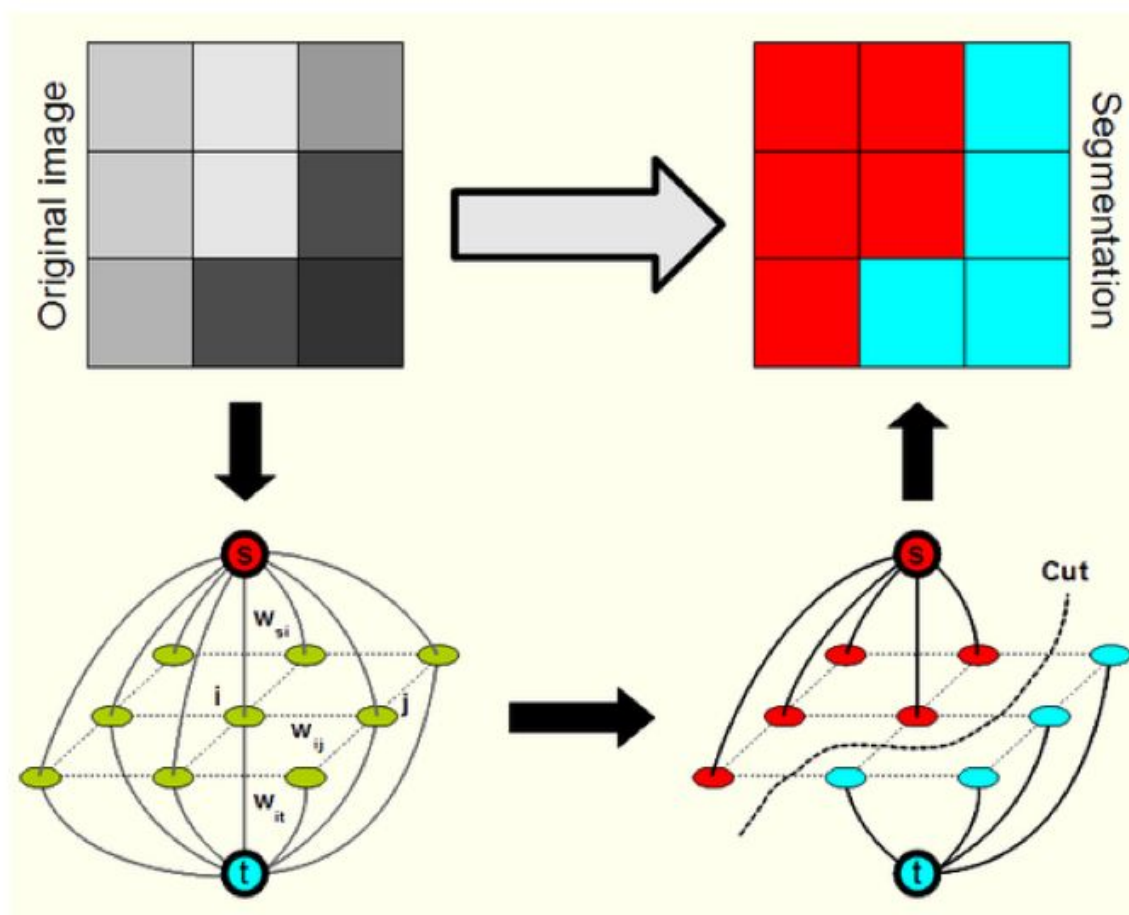
最小化分割解决了把权重图G分成两部分的任务，但是问题来了，如下图所示，想要的结果是中间实线表示的分割，但是最小化切割却切掉了最边缘的角。这种情况很容易理解，因为最小化切割就是让CUT(A,B)的值最小的情况，而边缘处CUT值确实是最小，因此我们输最小化切割时会有偏差的(bias)。如何去除这种偏差就要引入Normalized Cut算法了。



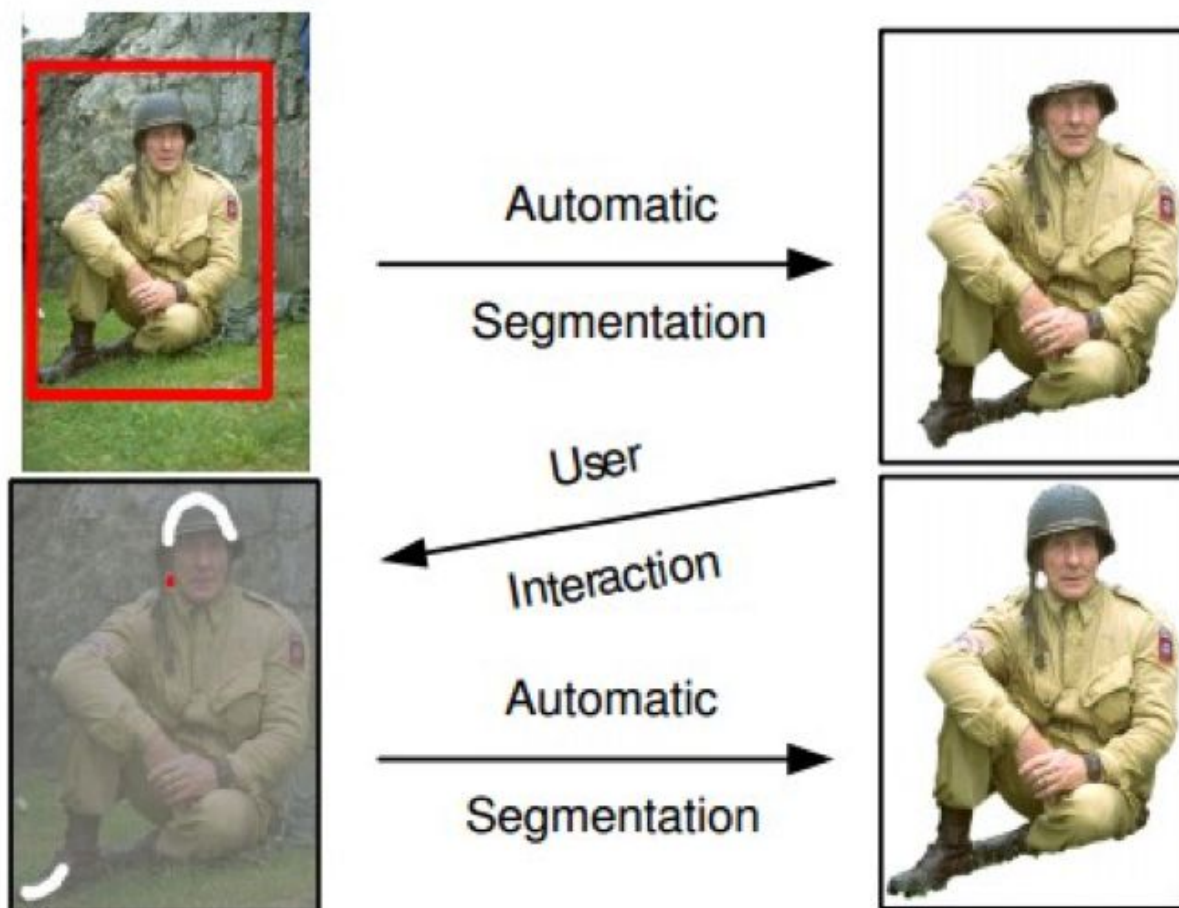
思路很简单，将Cut normalize一下，除以表现顶点集大小的某种量度(如  $\text{vol } A = \text{所有 } A \text{ 中顶点集的度之和}$ ，含义是A中所有点到图中所有点的权重的和)，也就是  $\text{NormalizeCut}(A, B) = \text{Cut}(A, B) / \text{vol } A + \text{Cut}(A, B) / \text{vol } B$ ，通过公式可以很清晰的看到NormalizeCut在追求不同子集间点的权重最小值的同时也追求同一子集内点的权重和最大值。

Graph Cuts图是在普通图的基础上多了2个顶点，这2个顶点分别用符号“S”和“T”表示，称为终端顶点。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，所以Graph Cuts中有两种顶点，也有两种边，第一种普通顶点对应于图像中的每个像素。每两个邻域顶点的连接就是一条边。这种边也叫n-links。除图像像素外，还有另外两个终端顶点，叫S源点和T汇点。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，如下图所示。

Graph Cuts中的Cuts是指这样一个边的集合，这些边集合包括了上面定义的2种边，该集合中所有边的断开会导致残留“S”和“T”图的分开，所以就称为“割”。如果一个割，它的边的所有权值之和最小，那么这个就称为最小割，也就是图割的结果。根据网络中最大流和最小割等价的原理，将图像的最优分割问题转化为求解对应图的最小割问题。由Boykov和Kolmogorov发明的max-flow/min-cut算法[1, 4]就可以用来获得S-T图的最小割，这个最小割把图的顶点划分为两个不相交的子集S和T，其中 $s \in S$ ， $t \in T$ 和 $S \cup T = V$ 。这两个子集就对应于图像的前景像素集和背景像素集，那就相当于完成了图像分割。



Graph Cuts 算法利用了图像的像素灰度信息和区域边界信息，代价函数构建在全局最优的框架下，保证了分割效果。但Graph Cuts 是NP 难问题，且分割结果更倾向于具有相同的类内相似度。Rother 等人提出了基于迭代的图割方法，称为Grab Cut 算法。该算法使用高斯混合模型对目标和背景建模，利用了图像的RGB 色彩信息和边界信息，通过少量的用户交互操作得到非常好的分割效果。



### 基于聚类的分割方法

机器学习中的聚类方法也可以用于解决图像分割问题，其一般步骤是：

- 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和SLIC等。

K-means算法是输入聚类个数 $k$ ，以及包含  $n$  个数据对象的数据库，输出满足方差最小标准 $k$ 个聚类的一种算法。K-means 算法接受输入量  $k$ ，然后将 $N$ 个数据对象划分为  $k$ 个聚类以便使得所获得的聚类满足：同一聚类中的对象相似度较高；而不同聚类中的对象相似度较小。

算法过程如下：

(1) 从 $N$ 个数据文档（样本）随机选取 $K$ 个数据文档作为质心（聚类中心）。

本文在聚类中心初始化实现过程中采取在样本空间范围内随机生成 $K$ 个聚类中心。

(2) 对每个数据文档测量其到每个质心的距离，并把它归到最近的质心的类。

(3) 重新计算已经得到的各个类的质心。

(4) 迭代（2）~（3）步直至新的质心与原质心相等或小于指定阈值，算法结束。



谱聚类(Spectral Clustering, SC)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，而子图间距离尽量距离较远，以达到常见的聚类的目的。与 K-means 算法相比不容易陷入局部最优解，能够对高维度、非常规分布的数据进行聚类。与传统的聚类算法相比具有明显的优势，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。为了进行聚类，需要利用高斯核计算任意两点间的相似度以此构成相似度矩阵。

谱聚类方法缺点：

- a)谱聚类对参数非常敏感；
- b)时间复杂度和空间复杂度大。

对于 k-way 谱聚类算法，一般分为以下步骤：

- (1) 构建相似度矩阵  $W$ ；
- (2) 根据相似度矩阵  $W$  构建拉普拉斯矩阵  $L$ （不同的算法有不同的  $L$  矩阵）；
- (3) 对  $L$  进行特征分解，选取特征向量组成特征空间；
- (4) 在特征空间中利用 K 均值算法，输出聚类结果；

Meanshift 算法的原理是在  $d$  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，得到的结果就是

Meanshift 向量。继续以Meanshift 向量的终点为圆心做圆，得到下一个Meanshift 向量.通过有限次迭代计算， Meanshift 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。利用Meanshift做图像分割，就是把具有相同模点的像素聚类到同一区域的过程,其形式化定义为：

$$y_{k+1}^{mean} = \arg \min_z \sum_i \|x_i - z\|^2 \varphi\left(\left\|\frac{x_i - y_k}{h}\right\|^2\right)$$

其中， $x_i$ 表示待聚类的样本点， $y_k$ 代表点的当前位置， $y_{k+1}$ 代表点的下一个位置， $h$ 表示带宽。Meanshift 算法的稳定性、鲁棒性较好，有着广泛的应用。但是分割时所包含的语义信息较少，分割效果不够理想，无法有效地控制超像素的数量，且运行速度较慢，不适用于实时处理任务。

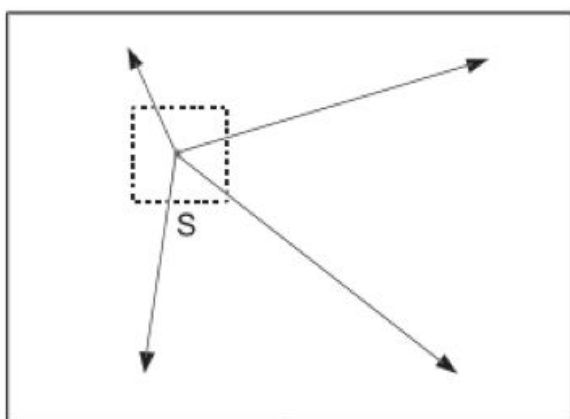
SLIC (simple linear iterative clustering) , 是Achanta 等人2010年提出的一种思想简单、实现方便的算法，将彩色图像转化为CIELAB颜色空间和XY坐标下的5维特征向量，然后对5维特征向量构造距离度量标准，对图像像素进行局部聚类的过程。SLIC算法能生成紧凑、近似均匀的超像素，在运算速度，物体轮廓保持、超像素形状方面具有较高的综合评价，比较符合人们期望的分割效果。



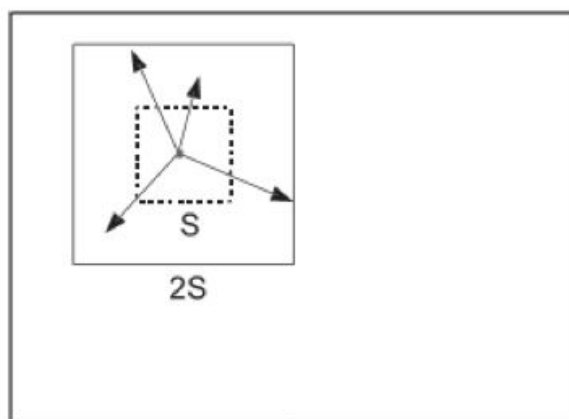
SLIC 算法的实质是将K-means 算法用于超像素聚类,众所周知, K-means 算法的时间复杂度为 $O(NKI)$ ,其中,  $N$  是图像的像素数, $K$  是聚类数,  $I$  是迭代次数.

SLIC具体实现的步骤:

- (1) 将图像转换为CIE Lab颜色空间
- (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。
- (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, 选择值最小(最平滑)的点作为新的种子点, 这一步主要是为了防止种子点落在了轮廓边界上。
- (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(计算方法在后文), 对于K-means 算法是计算整张图的所有像素点, 而SLIC得计算范围是 $2S*2S$ , 所以SLIC算法收敛速度很快。其中 $S = \sqrt{N/k}$ ,  $N$ 是图像像素个数。
- (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。



K-means对整个图片空间进行搜索



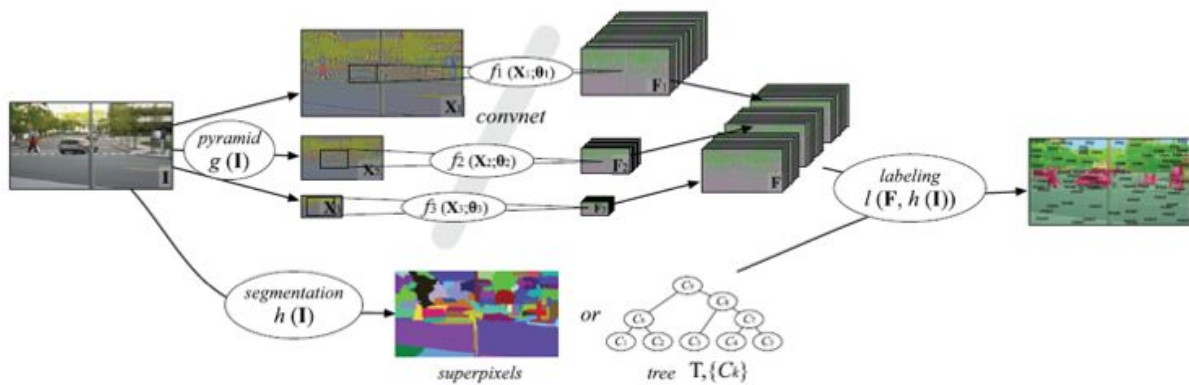
SLIC对某个局部空间进行搜索聚类

### 基于语义的分割方法

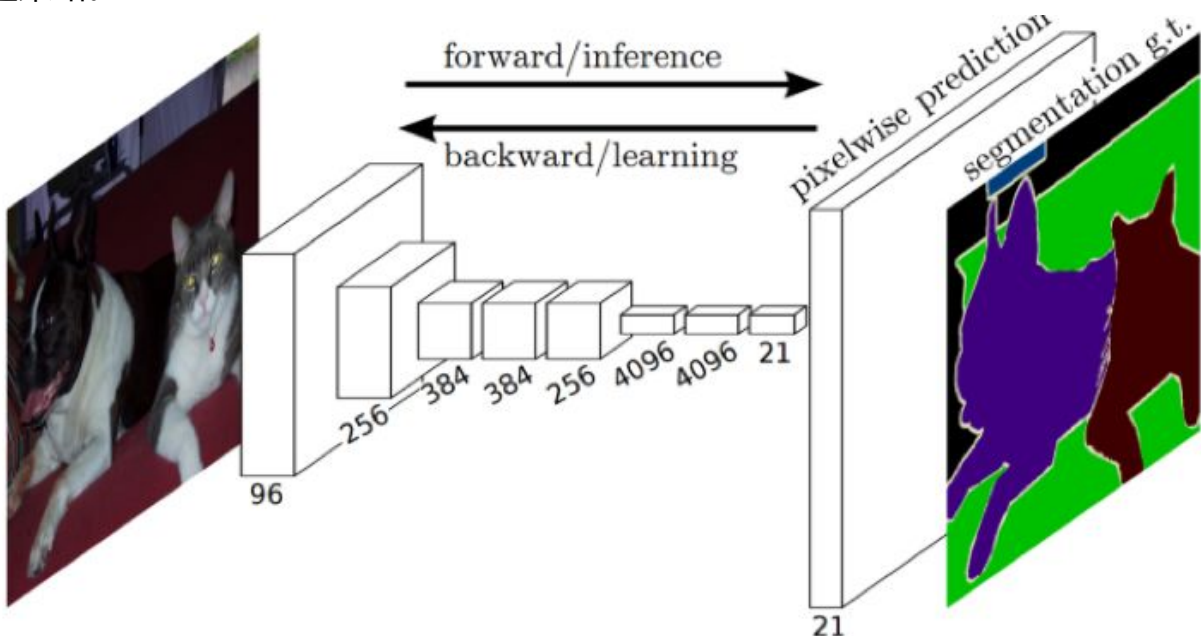
聚类方法可以将图像分割成大小均匀、紧凑度合适的超像素块,为后续的处理任务提供基础,但在实际场景的图片中,一些物体的结构比较复杂,内部差异性较大,仅利用像素点的颜色、亮度、纹理等较低层次的内容信息不足以生成好的分割效果,容易产生错误的分割。因此需要更多地结合图像提供的中高层内容信息辅助图像分割,称为图像语义分割。

深度学习技术出现以后,在图像分类任务取得了很大的成功,尤其是其对高级语义信息的model能力很大程度上解决了传统图像分割方法中语义信息缺失的问题。

2013年,LeCun的学生Farabet等人使用有监督的方法训练了一个多尺度的深度卷积分类网络[9]。该网络以某个分类的像素为中心进行多尺度采样,将多尺度的局部图像patch送到CNN分类器中逐一进行分类,最终得到每个像素所属的语义类别。在实际操作中,作者首先对图片进行了超像素聚类,进而对每个超像素进行分类得到最后的分割结果,一定程度上提高了分割的速度。这种做法虽然取得了不错的效果,但是由于逐像素的进行窗口采样得到的始终是局部信息,整体的语义还是不够丰富,于是就有了后面一系列的改进方案,本文中选择了几种代表性的网络进行逐一分析。



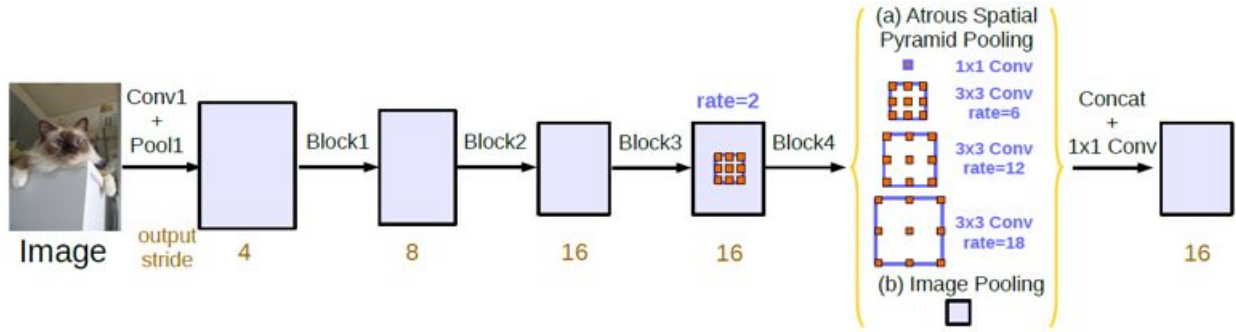
Long 等人于2014 年提出了FCN方法，这是深度学习在图像分割领域的开山之作，作者针对图像分割问题设计了一种针对任意大小的输入图像，训练端到端的全卷积网络的框架，实现逐像素分类，奠定了使用深度网络解决图像语义分割问题的基础框架。为了克服卷积网络最后输出层缺少空间位置信息这一不足，通过双线性插值上采样和组合中间层输出的特征图，将粗糙(coarse)分割结果转换为密集(dense)分割结果。FCN由于采用的下采样技术会丢失很多细节信息，后续的一系列方法也都做了相应的改进策略。





DeepLab-v1 [11]在FCN 框架的末端增加可fully connected CRFs，使得分割更精确。DeepLab 模型，首先使用双线性插值法对FCN的输出结果上采样得到粗糙分割结果，以该结果图中每个像素为一个节点构造CRF 模型 (DenseCRF)提高模型捕获细节的能力。该系列的网络中采用了Dilated/Atrous Convolution的方式扩展感受野，获取更多的上下文信息，避免了DCNN中重复最大池化和下采样带来的分辨率下降问题，分辨率的下降会丢失细节。

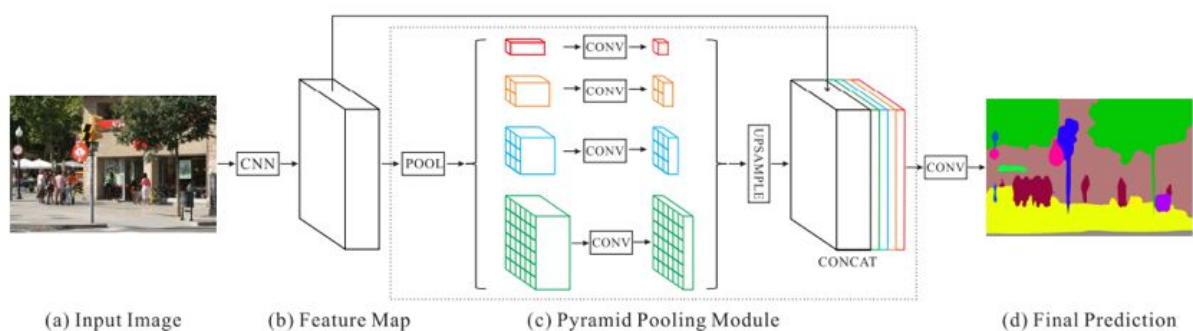
DeepLab-v2 [12]提出了一个类似的结构，在给定的输入上以不同采样率的空洞卷积并行采样，相当于以多个比例捕捉图像的上下文，称为ASPP(atrous spatial pyramid pooling)模块，同时采用了深度残差网络替换掉了VGG16增加了模型的拟合能力。DeepLab-v3 [13]重点探讨了空洞卷积的使用，同时改进了ASPP模块，便于更好的捕捉多尺度上下文，在实际应用中获得了非常好的效果。



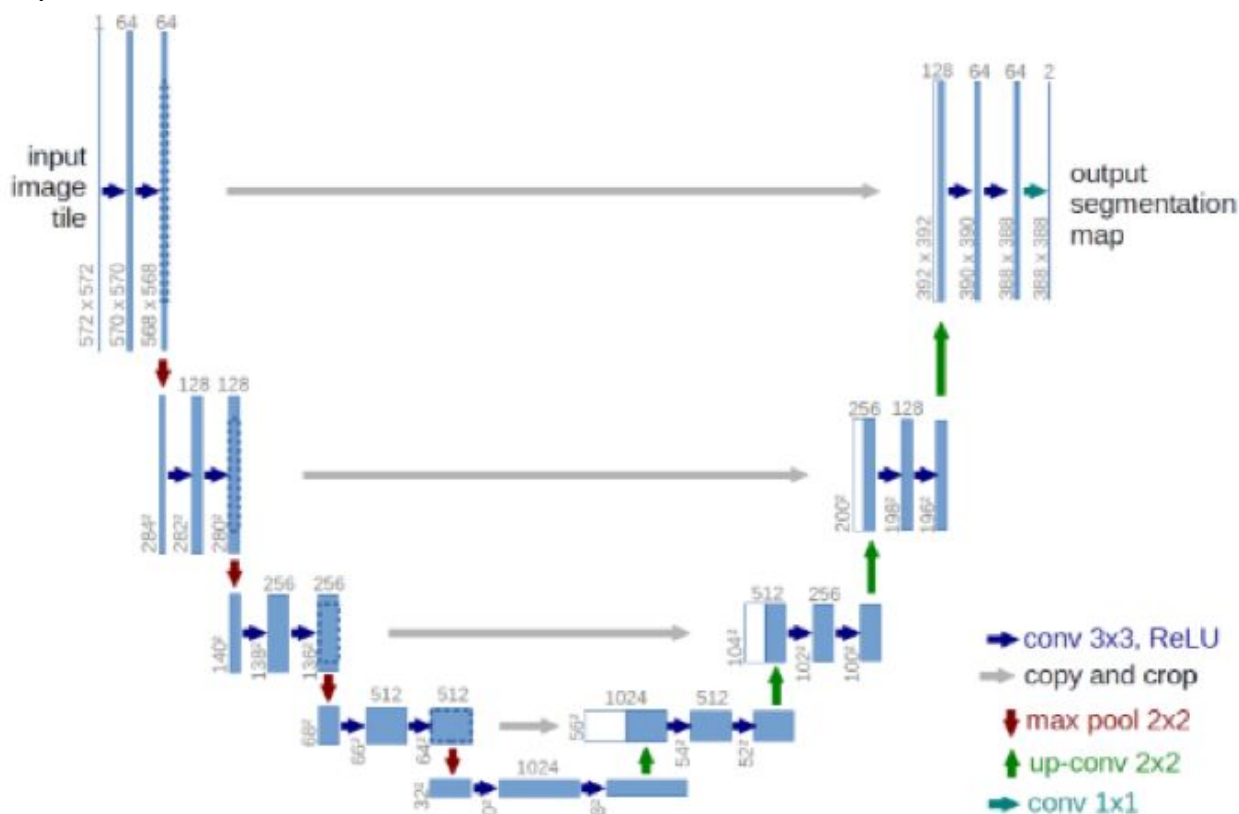
使用不同采样率的空洞卷积并行采样

作者提出的金字塔池化模块( pyramid pooling module)能够聚合不同区域的上下文信息,从而提高获取全局信息的能力，通过金字塔结构将多尺度的信息，将全局特征和局部特征嵌入基于FCN预测框架中，针对上下文复杂的场景和小目标的做了提升。为了提高收敛的速度，作者在主干网络中增加了额外的监督损失函数。



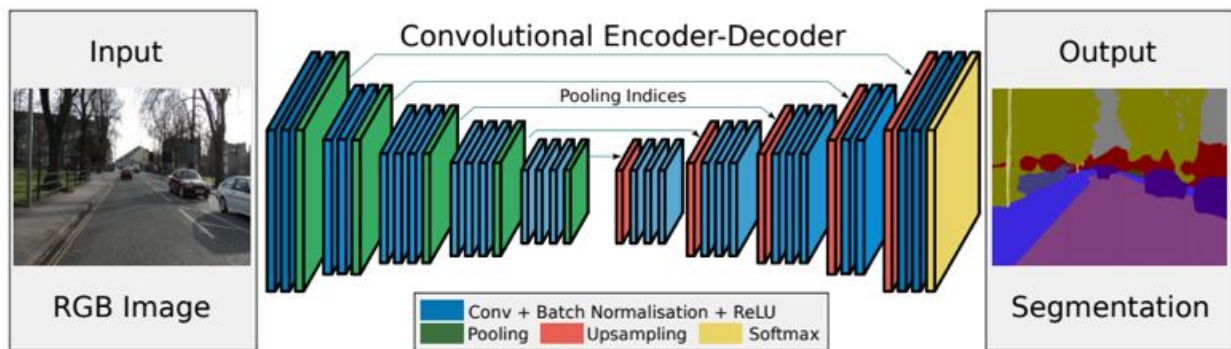


U-Net由菲兹保大学的Olaf Ronneberger等人在2015年提出，相较于FCN多尺度信息更加丰富，最初是用在医疗图像分割上。在生物图像分割中，最为突出了两个挑战是：可获得的训练数据很少；对于同一类的连接的目标分割。作者解决第一个问题的方法是通过数据扩大（data augmentation）。他们通过使用在粗糙的3\*3点阵上的随机取代向量来生成平缓的变形。解决第二个问题是通过使用加权损失（weighted loss），这是基于相邻细胞的分界的背景标签在损耗函数中有很高的权值。

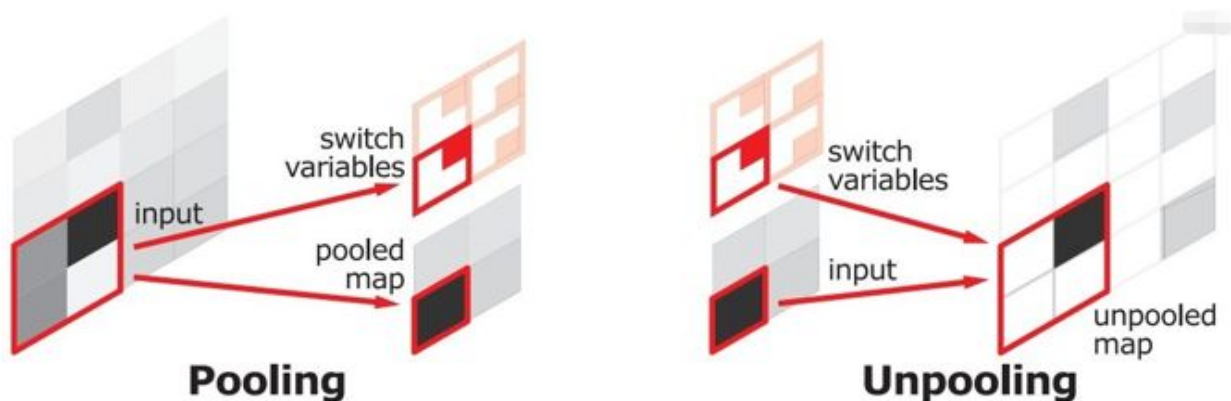


编码部分，每经过一个池化层就构造一个新的尺度，包括原图尺度一共有5个尺度。解码部分，每上采样一次，就和特征提取部分对应的通道数相同尺度融合。这样就获得了更丰富的上下文信息，在Decode的过程中通过多尺度的融合丰富了细节信息，提高分割的精度。

SegNet和FCN思路十分相似，只是Encoder中Pooling和Decoder的Upsampling使用的技术不一致。此外SegNet的编码器部分使用的是VGG16的前13层卷积网络，每个编码器层都对应一个解码器层。



在SegNet中的pooling与FCN的Pooling相比多了一个记录位置index的功能，也就是每次pooling，都会保存通过max op选出的权值在filter中的相对位置，Unpooling就是Pooling的逆过程，Unpooling使得图片变大2倍。max-pooling之后，每个filter会丢失的权重是无法复原的，但是在Unpooling层中可以得到在pooling中相对pooling filter的位置，所以Unpooling中先对输入的特征图放大两倍，然后把输入特征图的数据根据pooling indices放入，总体计算效率也比FCN略高。



随着深度学习在计算机视觉领域的巨大成功,以卷积神经网络为基础的图像语义分割方法取得了突破性的进展，将图像分割带到了一个新的高度。笔者认为如何继续提高分割算法的精度，以及降低分割算法的复杂度是值得继续研究的问题。

- [1] Greig DM, Porteous BT, Seheult AH. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society*, 1989,51(2):271–279.
- [2] J. Shi and J. Malik, Normalized Cuts and Image Segmentation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, Aug. 2000.
- [3] Y. B. Yuri, G. F. Lea, "Graph Cuts and Efficient N-D Image Segmentation," *International Journal of Computer Vision*, vol.70, no.2, pp.109-131, 2006.
- [4] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *International Conference on Computer Vision*, volume I, pages 26–33, 2003.
- [5] Rother C, Kolmogorov V, Blake A. GrabCut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. on Graphics*, 2004, 23(3):309–314. [doi:10.1145/1186562.1015720]
- [6] Comaniciu D, Meer P. Mean Shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2002,24(5):603–619. [doi: 10.1109/34.1000236]
- [7] Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Susstrunk S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2012,34(11):2274–2282. [doi: 10.1109/tpami.2012.120]
- [8] 姜枫,顾庆,郝慧珍,李娜,郭延文,陈道蓄.基于内容的图像分割方法综述.软件学报,2017,28(1):160–183.  
<http://www.jos.org.cn/1000-9825/5136.htm>
- [9] Farabet, C., Couprie, C., Najman, L., and LeCun, Y. (2013). Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929.
- [10] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: *Proc. of the 28th IEEE Conf. on CVPR*. Washington: IEEE Computer Society, 2015. 1337–1342. [doi: 10.1109/cvpr.2015.7298965]
- [11] Krähenbühl P, Koltun V. Efficient inference in fully connected CRFS with gaussian edge potentials. In: *Proc. of the Advances in Neural Information Processing Systems*. Cambridge: MIT

Press, 2012. 109–117. <http://papers.nips.cc/paper/4296-efficientinference-in-fully-connected-crfs-with-gaussian-edge-potentials>

[12] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv preprint arXiv:1606.00915, 2016.

[13] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587, 2017. 5, 8, 11

[14] Zhao, Hengshuang, Shi, Jianping, Qi, Xiaojuan, Wang, Xiaogang, and Jia, Jiaya. Pyramid scene parsing network. arXiv preprint arXiv:1612.01105, 2016.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in MICCAI, 2015.

[16] Kendall, Alex, Vijay Badrinarayanan, and Roberto Cipolla. "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding." arXiv preprint arXiv:1511.02680 (2015).

## ▲向上滑动查看更多推荐文章

十一月后文章，本公众号不提供PDF版本，请移步[www.sigai.cn](http://www.sigai.cn)

☐知识库☐技术文章，获取PDF版本。



科普类

【获取码】SIGAI0413

[机器学习——波澜壮阔四十年](#)

【获取码】SIGAI0620

[理解计算：从 \$\sqrt{2}\$ 到AlphaGo ——第1季从 \$\sqrt{2}\$ 谈起](#)

【获取码】SIGAI0702

[理解计算：从 \$\sqrt{2}\$ 到AlphaGo ——第2季 神经计算的历史背景](#)

【获取码】SIGAI0702

[理解计算：从 \$\sqrt{2}\$ 到AlphaGo ——第3季 神经网络的数学模型](#)

【获取码】SIGAI0702

[理解计算：从 \$\sqrt{2}\$ 到AlphaGo ——第4季 凛冬将至](#)

【获取码】SIGAI1019

[理解计算：从 \$\sqrt{2}\$ 到AlphaGo ——第5季 导数的前生今生](#)

【获取码】SIGAI0829

[AI时代大点兵——国内外知名AI公司2018年最新盘点【完整版】](#)

【获取码】SIGAI1008

[非算法类人工智能从业者须知的十件事](#)

【获取码】SIGAI1102

[永远的金大侠——人工智能的江湖](#)



数学类

【获取码】SIGAI0417

[学好机器学习需要哪些数学知识](#)

【获取码】SIGAI0511

[理解梯度下降法](#)

【获取码】SIGAI0518

[理解凸优化](#)

【获取码】SIGAI0531

[理解牛顿法](#)

【获取码】SIGAI1031

[理解概率密度函数](#)



## 机器学习类

【获取码】SIGAI0428

[用一张图理解SVM的脉络](#)

【获取码】SIGAI0505

[理解神经网络的激活函数](#)

【获取码】SIGAI0522

[【实验】理解SVM核函数和参数的作用](#)

【获取码】SIGAI0601

[【群话题精华】五月集锦—机器学习和深度学习中一些值得思考的问题](#)



【获取码】SIGAI0602

[大话AdaBoost算法](#)

【获取码】SIGAI0606

[理解主成分分析（PCA）](#)

【获取码】SIGAI0611

[理解决策树](#)

【获取码】SIGAI0613

[用一句话总结常用的机器学习算法](#)

【获取码】SIGAI0618

[理解过拟合](#)

【获取码】SIGAI0627

[k近邻算法](#)

【获取码】SIGAI0704

[机器学习算法地图](#)

【获取码】SIGAI0706

[反向传播算法推导—全连接神经网络](#)

【获取码】SIGAI0720

## [流形学习概论](#)

【获取码】SIGAI0725

## [随机森林概述](#)

【获取码】SIGAI0711

## [怎样成为一名优秀的算法工程师](#)

【获取码】SIGAI0802

## [机器学习和深度学习中值得弄清楚的一些问题](#)

【获取码】SIGAI0808

## [机器学习和深度学习核心知识点总结--写在校园招聘即将开始前](#)

【获取码】SIGAI0822

## [机器学习中的最优化算法总结](#)

【获取码】SIGAI0824

## [浓缩就是精华--SIGAI机器学习蓝宝书](#)

【获取码】SIGAI0831

## [理解AdaBoost算法](#)

【获取码】SIGAI0903

## [深入浅出聚类算法](#)

【获取码】SIGAI0905

## 机器学习发展历史回顾

【获取码】SIGAI0914

## 理解Logistic回归

【获取码】SIGAI0917

## 机器学习中的目标函数总结

【获取码】SIGAI0921

## 机器学习与深度学习常见面试题

【获取码】SIGAI1008

## 理解生成模型与判别模型

【获取码】SIGAI1024

## 机器学习和深度学习常见面试题（下）

### 深度学习类

【获取码】SIGAI0907

## 网络表征学习综述

【获取码】SIGAI0917

## 机器学习中的目标函数总结

【获取码】SIGAI0822

## 机器学习中的最优化算法总结

【获取码】SIGAI0426

[卷积神经网络为什么能够称霸计算机视觉领域？](#)

【获取码】SIGAI0827

[DenseNet详解](#)

【获取码】SIGAI0508

[深度卷积神经网络演化历史及结构改进脉络-40页长文全面解读](#)

【获取码】SIGAI0515

[循环神经网络综述—语音识别与自然语言处理的利器](#)

【获取码】SIGAI0625

[卷积神经网络的压缩与加速](#)

【获取码】SIGAI0709

[生成式对抗网络模型综述](#)

【获取码】SIGAI0711

[怎样成为一名优秀的算法工程师](#)

【获取码】SIGAI0718

[基于深度负相关学习的人群计数方法](#)

【获取码】SIGAI0723

## [关于感受野的总结](#)

【获取码】SIGAI0802

## [机器学习和深度学习中值得弄清楚的一些问题](#)

【获取码】SIGAI0806

## [反向传播算法推导--卷积神经网络](#)

【获取码】SIGAI0808

## [机器学习和深度学习核心知识点总结--写在校园招聘即将开始前](#)

【获取码】SIGAI0810

## [理解SpatialTransformer Networks](#)

【获取码】SIGAI0824

## [浓缩就是精华--SIGAI机器学习蓝宝书](#)

【获取码】SIGAI0905

## [机器学习发展历史回顾](#)

【获取码】SIGAI0921

## [机器学习与深度学习常见面试题](#)

【获取码】SIGAI0928

## [轻量化神经网络综述](#)

【获取码】SIGAI1015

### 三维深度学习中的目标分类与语义分割

【获取码】SIGAI1017

### 化秋毫为波澜：运动放大算法(深度学习版)

【获取码】SIGAI1022

### 深度多目标跟踪算法检测

【获取码】SIGAI1024

### 机器学习和深度学习常见面试题（下）

## 机器视觉类

【获取码】SIGAI0420

### 人脸识别算法演化史

【获取码】SIGAI0424

### 基于深度学习的目标检测算法综述

【获取码】SIGAI0426

### 卷积神经网络为什么能够称霸计算机视觉领域？

【获取码】SIGAI0503

### 人脸检测算法综述

【获取码】SIGAI0525

### 【SIGAI综述】行人检测算法



【获取码】SIGAI0604

[FlowNet到FlowNet2.0: 基于卷积神经网络的光流预测算法](#)

【获取码】SIGAI0608

[人体骨骼关键点检测综述](#)

【获取码】SIGAI0615

[目标检测算法之YOLO](#)

【获取码】SIGAI0622

[场景文本检测——CTPN算法介绍](#)

【获取码】SIGAI0629

[自然场景文本检测识别技术综述](#)

【获取码】SIGAI0716

[人脸检测算法之S3FD](#)

【获取码】SIGAI0727

[基于内容的图像检索技术综述--传统经典方法](#)

【获取码】SIGAI0817

[基于内容的图像检索技术综述--CNN方法](#)

【获取码】SIGAI0910

## [视觉多目标跟踪算法综述（上）-附开源代码下载链接整理](#)

【获取码】SIGAI0914

## [计算机视觉技术self-attention最新进展](#)

【获取码】SIGAI0919

## [人脸识别中的活体检测算法综述](#)

【获取码】SIGAI0924

## [浅谈动作识别TSN, TRN, ECO](#)

【获取码】SIGAI0926

## [OCR技术简介](#)

【获取码】SIGAI1012

## [目标检测最新进展总结与展望](#)

【获取码】SIGAI1029

## [目标检测算法中检测框合并策略技术综述](#)

## 自然语言处理

【获取码】SIGAI0803

## [基于深度神经网络的自动问答概述](#)

【获取码】SIGAI0820

## [文本表示简介](#)

工业应用类

【获取码】SIGAI0529

[机器学习在自动驾驶中的应用-以百度阿波罗平台为例【上】](#)

[本文为SIGAI原创](#)

[如需转载，欢迎发消息到本订阅号](#)

**[免费挑战4P计划【2.0】：用Python实现一个深度学习实践项目](#)**

**小程序Python课程试听（详情戳蓝字）**

报名方式：

方式1：关注SIGAI公众号，回复“4P计划”即可获得申请链接



长按识别上方二维码关注

方式2： 登陆[www.sigai.cn](http://www.sigai.cn)，选择《深度学习Python开发-基础篇》，进入详情页，点击立即报名

[阅读原文](#)