**前言:**

1.本次汇报尚未给出DeepLabv2的实现与filter支持特征向量对齐图像边缘的具体过程(filter那块还没看明白)

2.本次汇报使用word(PPT太难做了而且视觉效果差)

3.本次汇报未使用英文(视觉效果差,行距稀奇古怪,且难度较高)

4.Semantic Soft Segmentation实现原理:提取一个精心构建的图的拉普拉斯矩阵的特征值,取最小的100个特征值对应的特征向量通过优化方程进行线性组合,生成多个层(layers),再通过K-mean cluster将生成的有意义的层合并为5个,后在能量最小预置共轭梯度优化下获取更精确的最终层.其中对拉普拉斯矩阵特征向量在matting中发挥的作用,参考语义软分割（Semantic Soft Segmentation）的学习笔记（上篇 ） - 黄二二的文章 - 知乎

<https://zhuanlan.zhihu.com/p/58606246>

**需通过读代码才能理解的问题:**

**全文是否均是基于超像素来实现的?后续提到的pixels是否直接代表的是superpixels?**

**理由:**

<1>Laplacian matrices是基于超像素构建的,其能直接解决的也理应是SLIC处理后的图像.

<2>原文中在 部分有提及

”,然而原文并未具体讲明如何对特征向量表示的像素使用k-means clustering.而SLIC生成超像素的过程可等价为对像素使用k-means clustering.

**一、具体实现过程**

**1. 构建拉普拉斯矩阵(是基于超像素的!!!!!!!).**

使用公式:



公式中, D为degree matrix(度矩阵),为光谱亲和度矩阵,为高级语义亲和度矩阵,为非局部颜色亲和度矩阵,与控制对应矩阵对拉普拉斯矩阵的影响权重,本方法中均设置为0.01.对公式中各项的详细介绍与计算方式如下.

1.1超像素的生成.该拉普拉斯矩阵是建立在超像素的基础上的.使用SLIC为图像生成2500个超像素点,(SLIC生成超像素点的过程另见文档)

**1.2**

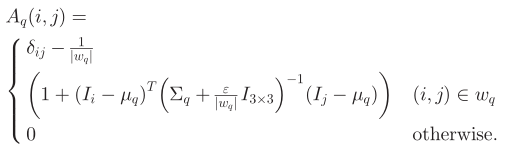
是图的度矩阵.若图的邻接矩阵为W,则

i为行,j为列

**1.3**

WL是代表matting affinity的矩阵,其计算方式如下:

设置每个像素点为I,则有

其中为图像中第i个超像素点的平均前景色,为第i个超像素点在第k层上对应的α值(代表前景不透明度),为第i个超像素点在第k层上对应的前景色.由此引出的计算方程:

其中, 表示Kronecker delta(克罗内克函数):

,表示一个window(窗口)中超像素的总数, 表示超像素q周围的平均颜色向量(mean colar vector),表示该window的一个3\*3协方差矩阵,表示一个3\*3大小的单位矩阵,

**1.4 WC**

WC是代表Nonlocal color affinity的矩阵,其计算方式如下:

此后有



表示生成的超像素表示两超像素的平均颜色向量,(使用 (α\*R,β\*G,γ\*B)表示,且α+β+γ=1),|| X ||表示X向量的绝对值,, ,用于控制affinity下降速度.该项目中,设为50,设为0.05.erf是高斯误差函数,其定义如下:



PS:每个超像素仅计算自身与以质心为中心,图像大小的20％为半径的弧形区域(应该就是⚪)内的其它超像素的colar affinity

**1.5**

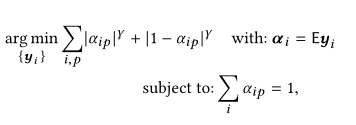
WS是代表 semantic affinity的矩阵.WS的计算过程如下:

其中,与表示超像素s与t的平均特征向量,像素的特征向量由神经网络生成(接下来会提到).该过程中,,分别为20,0.2.至此拉普拉斯矩阵表示完成.

**2.对拉普拉斯矩阵进行进一步处理.**

**2.1 约束稀疏化()**

获取拉普拉斯矩阵后,计算并提取其100个最小特征值对应的特征向量,在特征向量所表示的像素上使用K-mean聚类(仍然基于超像素?所以下方的所有”像素”都实际代表的是超像素?),并将其结果映射到矩阵中,具体实现过程如下:

在以下优化方程中令=0.8.(在spectral matting中γ=0.9)

其中代表第个matting component(即层layer),代表第个线性组合(一个matting component用一组特征向量的线性组合表示),p代表第i层的第p个像素.

约束条件:同一像素的各层的α值和为1,α为不透明度.

生成多个层(获取多个)后,舍弃全0层,对有意义层再次使用K-mean cluster ,将层数聚类至5.新生成的层用对应的多个线性组合的平均值表示.

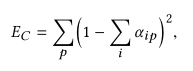
**2.2 宽松稀疏化()**

设计energy function:



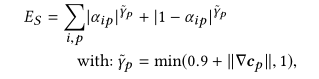
其中,的作用是放宽子空间约束,使得生成的层靠近通过建立的层即可.

的作用是放宽求和为1的要求.



是拉普拉斯矩阵的传播能量.

用于适应图像内容.在很多情况下,其对应于场景元素之间的过渡.

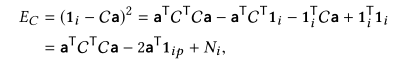


其中各项具体计算过程如下:

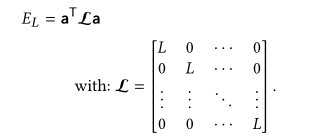
设=5(生成的图像层数),等于图片所包含的像素数.

设为使用宽松稀疏化生成的layers,为使用约束稀疏化生成的layers.

设为生成的5层软分割组成的矩阵,为生成的5层软分割构成的矩阵,则与的规模均为.设为规模为的单位矩阵,则



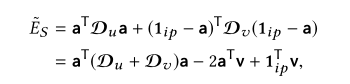
设为规模为的单位矩阵水平连接而成的规模为的矩阵, 向量由个1组成，向量由个1组成.则有



:

定义是的二阶导数,是的二阶导数. 即,

定义,,为值构成的列向量.

此处使用来近似.

将各项计算结果代入,得

其中λ置100.

使用**预置共轭梯度优化**()求解方程,使得取得最小值,对应即为最终软分割矩阵.

**3.语义特征向量():**

使用DeepLab-ResNet-101 作为特征提取器，但网络训练是采用的是度量学习方法，最大化了不同物体特征间的 L2 距离(稍微修改了 N-Pair loss).

在 COCO-Stuff 数据集上进行语义分割网络的训练.

采用 guided filter 将网络生成的 feature map 与图像进行边缘对齐.  
然后采用 PCA(主成分分析法) 降低 feature map 维度到 3.  
最后，归一化特征向量值到 [0, 1],生成语义特征向量,供拉普拉斯矩阵使用.

**二.遇到的问题**

1.文献部分内容理解不够透彻.如:对能量函数E优化后如何得到稀疏特征向量的线性组合一知半解(已解决)

2.对经典算法了解不够.

3.对IDE的认知尚显浅薄,没能成功将代码跑起来(python实现的语义特征向量生成了,matlab他给的readme啥也没有,仍在尝试).

4.基础知识仍需巩固!尤其是矩阵部分!

**三.下一步计划**

1.阅读代码,基于代码修正对于文献的不当理解.

2.读透文献,进一步掌握文献中提到的方法.如: DeepLabResNet-101如何进行特征提取(注:semantic soft segmentation中生成语义特征向量的CNN为DeepLabv2,目前该CNN已更新至DeepLabv3+)?

Guided filter如何将 feature map与图像边缘对齐(涉及大量图像滤波器相关知识)?等等.

3学习(这个忘了好多,太顶了)矩阵计算相关知识.

附: <1>SLIC生成超像素过程

SLIC具体实现(和 K-mean cluster 类似!)

设图像共有N个像素点,预将这些像素点划分为K个超像素,图像在CIELAB空间中处理.

CIELAB中颜色性质: 一种颜色由 l,a\*,b\* 3个参数表示.

l表示亮度值,0为黑色,100为白色.

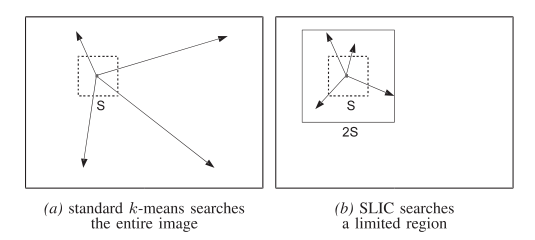
a\*表示红绿转变,符号为负数时显示为绿,正值显示为红

b\*表示蓝黄转变,符号为负时显示为蓝,符号为正时显示为黄.

1.挑选K个初始聚类中心,每个聚类中心之间的距离为S,S=sqrt(N/K) .聚类中心可用聚类向量C1,C2,……CK表示,Ci=[ l ,a\*, b\*, x, y ].其中前3个参数表示为聚类中心颜色特性,x,y表示聚类中心在image中所处位置.

2.移动聚类中心位置.在单个聚类中心周围n\*n( n常取值为3)范围内计算各像素点梯度,将聚类中心移动到该范围内梯度值最低的像素点上.(图像边缘像素点梯度高,避免聚类中心位于图像边缘.)

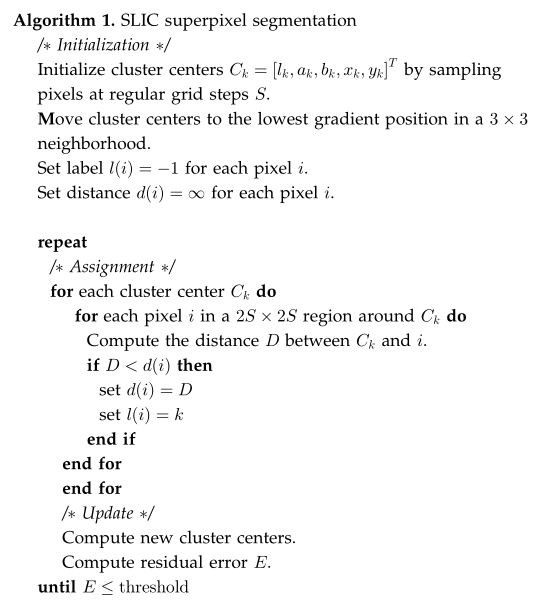
3.将每个像素与距离其最近的聚类中心相关联.使用距离函数 D(稍后提到).一个聚类中心关联范围为2S \* 2S .该方法极大减少了关联过程中的距离计算量,减少了计算时间.具体情况如下:



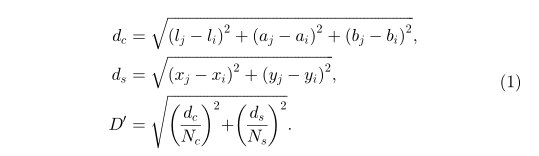
4.当所有像素都聚类到对应的聚类中心后,依次计算每个簇的平均聚类向量 ,此处记为L1, 并将聚类中心移动到L1对应的位置. 使用L2记录初始聚类中心和新的聚类中心的残差E.重复该步骤,直至残差E收敛(大致需要迭代重复十次)

(聚类中心是在不断调整变化的!)

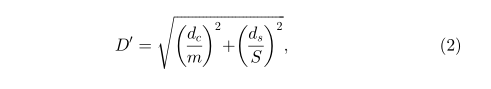
对应的伪代码如下:

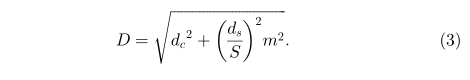


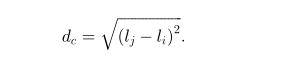
5.距离计算:



Ns=sqrt(N/K),Nc直接用常数m代替,则原方程式可以优化为:

进一步简化得:



若处理灰度图,仅需将dc替换:

若要处理超体素,则需将ds替换:

<2>K-mean cluster 实现原理:

5 分钟带你弄懂 K-means 聚类 - 打工人小飞的文章 - 知乎

<https://zhuanlan.zhihu.com/p/357072839>

<3>主成分分析(用于降维)实现原理: [CodingLabs - PCA的数学原理](http://blog.codinglabs.org/articles/pca-tutorial.html)

<4>预置共轭梯度法: <http://t.csdn.cn/aRnfk>