**语义分割调研报告**

报告人：蒋承知 学校：电子科技大学 联系方式：13098775238

**0 综述**

语义分割是对图像中每个像素进行划分，本质上仍旧是一种逐像素的分类问题。语义分割的方法可以抽象为：全卷积CNN对图像进行特征提取，然后融合浅层特征进行上采样将score map恢复到输入大小。其中涉及到的几个重要的部分为：CNN backbone、特征融合策略、上采样策略。另外，与图像分类不同的是，单看每个像素都是一个孤立点，要想分割效果好就要充分利用当前像素点周围的特征信息。语义分割的论文始终都是在围绕这四个点进行不断迭代改进，只是所选择的思路有所不同。比如对于“pool层在扩大感受野的同时会损失空间信息”这个问题，FCN采用与浅层特征相融合的方式解决，而DeepLab系列的策略则是用空洞卷积代替pool层，在不损失空间信息的前提下增大卷积的感受野。

**1 FCN(CVPR2015)**

**1.1 原理**

对于普通的CNN网络，将最后的全连接层变成卷积层得到相应的feature map而不是特征向量（此时的feature map空间大小为输入的1/32）。然后通过1\*1卷积将通道数规范到与类别数相等，通过反卷积将feature map的空间尺寸**直接一步**扩大到与输入相等（FCN-32s）得到最终的输出。

仅仅只利用特征提取部分最后一层的feature map的话，会存在两个问题影响最终的分类性能：1、由于经过了5次最大池化此时feature map的大小仅为原始的1/32，损失了太多的空间信息，对于逐像素划分的任务来说此时的feature map是很粗糙的；2、从1/32的尺度一步恢复到原输入大小，空间尺寸的陡增其实也是不利于信息的恢复的。

高层网络损失的信息在低网络层中还存在，所以在反卷积的过程中**逐步**与低层（Pool3、Pool4）的feature map进行结合，一方面融合所需的空间信息，另一方面也以一个适宜的速率恢复空间大小（×2、×2、×8），得到最终的输出（FCN-16s、FCN-8s）。

**1.2 关于改进的思考**

(1) **从网络backbone的角度**：论文中的网络backbone基于AlexNet，在FCN的算法框架下对于更复杂的场景将backbone替换成更深的网络，特征提取的效果会更好，并且用于融合的低层feature map选择会更加灵活多样。

(2) **从损失函数的角度**：FCN的原始损失是逐像素的分类损失，将原始的softmax loss换成引入间隔之后的大间隔损失应该也会有所提升，参考sphereface、cosface、arcface等。

**1.3 为什么全卷积可以？**

CVPR2016的Class Activation Mapping说明了：CNN中其实不仅隐含了关于输入的特征信息，同时也包含了物体的位置信息，只是全连接层把这种位置信息破坏掉了。全卷积结构保留了图像中的位置信息，结合提取出的特征从而得已对不同位置像素的类别做出正确的判断。

**2 SegNet(TPAMI 2017)**

**2.1 原理**

SegNet与FCN的思路类似，都是运用全卷积网络得到特征图然后将特征图恢复到原图尺寸得到相应的稠密预测输出。SegNet基于FCN但区别（改进）在于：

(1) **Backbone结构不同**：SegNet的特征提取backbone更深，采用VGG16的前13的卷积层；

(2) **上采样策略不同**：FCN在上采样的过程中采用反卷积技术，并不断与低层网络的特征图进行信息融合。而SegNet中采用带indices记忆的池化技术，当上采样时用存储的indices可以将当前特征图的值恢复到原始的位置得到稀疏的扩大的特征图，紧接着用卷积将稀疏的特征图稠密化（这一步操作可以理解为高维特征信息的解压）。特征图大小逐步扩大，直到与输入相等。SegNet中每一步上采样都用到了对应尺寸下的池化indices，这也解释了为什么SegNet是一个Encoder-Decoder的对称结构。

**2.2 关于改进的思考**

SegNet在上采样过程中并没有与低层的特征进行融合，仅仅只用了pool indices的信息，池化操作保存的indices即是特征图中的精准的位置信息，这解决了特征在pool kernel内部的位置混淆问题，这是相比直接特征融合更好的地方，但缺点在于没有低层特征的信息，下采样过程中缺失的空间信息很难通过卷积恢复。

**3 U-Net**

**3.1 原理**

U-Net仍旧是在FCN算法框架下做出的改进，虽然U-Net的图画的很新颖，但把网络拉直之后仍旧是一个FCN网络，区别（改进）在于：

(1) U-Net对低层的特征运用的更加充分，每一个尺寸的特征图在上采样阶段都会得到融合，这也导致了U-Net在网络结构上的对称性；

(2) 上采样阶段的特征融合策略不同，FCN中采用的是逐点相加的方法，而U-Net则采用的是concate + 2×2 conv的策略。

**3.2 关于改进的思考**

U-Net在FCN算法框架下把能完善的点都进行了完善，如果要继续改进可以考虑将backbone与ResNet或者DenseNet进行结合，加强**Encoder阶段特征的复用程度**。

**3.3 为什么U-Net可以适应小的数据集？**

个人觉得这是个伪命题。在医学图像领域中确实存在数据量小的问题，U-Net也能在小数据量下得到较好的泛化性，但主要原因不在于网络结构，而在于医学图像本身不会像自然图像那样具有较大的方差，较小的数据量也能cover住所涉及到的问题的分布。再加上U-Net本身充分利用了低层的网络特征，能够得到一个较好的学习效果。

**阶段总结：**

上述的FCN、SegNet、U-Net都是在FCN算法框架下的演化和改进：SegNet的改进着重点在网络的上采样策略上，而U-Net是在特征融合粗略上做了改良和完善。FCN框架下每个点的感受野是随着网络的加深而不断扩大的，这种框架下的通病在于，**网络丢失了太多的空间信息并且没有充分地去运用像素点周围的上下文信息**。

为了解决这个问题，*（从当下的眼光来看）*，可以考虑采用**空洞卷积**或者**可变卷积**在参数量不变的情况下扩大感受野，或者引入**注意力机制**，如Non-Local，更好的获取像素点与周围的特征关系。同时对于损失函数的改进，都可以尝试大间隔分类损失的效果。

**4 DeepLab系列**

**4.1 原理**

**4.1.1** **DeepLab V1 (ICLR 2015)**

V1采用backbone为VGG16的全卷积网络，通过CNN得到feature map并通过双线性插值恢复到原图大小得到coarse output，最后用CRF得到精细输出。V1的改进：

(1) **空洞卷积**。FCN存在“多次下采样导致空间信息损失过多不利于稠密预测”的问题。但去掉maxpool层又会减小网络的感受野，V1选择了一种和FCN不同的解决思路，通过将网络最后两层maxpool层去掉，在相应的网络层引入空洞卷积在更多保留空间信息的同时对网络的感受野不产生影响，通过空洞卷积的引入最终的特征图大小从原输入的1/32变为1/8，特征密度也更大了。

(2) **条件随机场CRF。**V1中运用CRF对CNN输出的coarse output进行精细化调整。CRF将coarse output建模为一个全连接的无向图模型，每个像素代表一个节点，边代表节点间的关系。CRF就相当于用每个节点的全局关系进行调整（个人理解类似于non-local的作用）。

(3) **上采样策略不同。**不同于FCN中采用的反卷积以及SegNet中采用的pool indices策略。V1中直接采用**双线性插值**(×8)将feature map恢复到输入大小。

(4) **多尺度特征融合。**V1中同样也融合了低网络层的特征，将网络前四个max pool的feature map提取出来(每个featuremap后接3\*3卷积、1\*1卷积，输出128通道的feature map)与主干网络输出concate起来之后一起送入最终的分类层中。

**4.1.2 DeepLab V2 (arXiv 2016、TPAMI 2017)**

相比于V1，V2的算法Pipeline没有太多的变化，在保留CRF后处理的基础上，区别（改进）在于：

(1) **backbone升级**。V2的backbone从VGG16升级为ResNet-101，网络的表达能力更强；

(2) **ASPP(Atrous Spatial Pyramid Pooling)模块**。V1中引入了空洞卷积但没有对其扩张率做太多的调整，空洞卷积的扩张率对应了不同的尺度和感受野。针对“图像中同一类物体存在不同的尺度”这一特点，V2中提出ASPP模块，对同一特征图并行地用多个不同扩张率（6，12，18，24）的空洞卷积进行处理然后进行特征融合。

**4.1.3 DeepLab V3 (arXiv 2017)**

V3主要做了三件事：

(1) **ASPP模块进行了改进**。V3中提到虽然不同扩张率的空洞卷积可以得到不同尺度的特征信息，但是随着扩张率的增大，kernel所能覆盖到的有效特征值越来越少，极端情况下会退化成1×1卷积。所以V3对ASPP的支路做了变化：用1×1卷积将rate=24的支路替换掉，同时为了结合图像全局信息还新增了全局平均池化支路，并运用双线性插值恢复到所需的尺寸。另外，ASPP还加入了BN层帮助训练。

(2) 对于空洞卷积模块，文章给出了一种串行的网络设计使得网络可以更深；

(3) **去掉了CRF后处理模块**，DCNN的输出已经够好，后处理意义不大了。

**4.1.4 DeepLab V3+ (ECCV 2018)**

V3+基于V3的改进和结构，主要的区别（改进）在于：

(1) **backbone升级**。从原本的ResNet101改为了Xception，并且将深度可分离卷积应用到了ASPP模块中，所有的pool层都用；

(2) **替换Pool层**。V3+将Xception的结构也做了一定的改进，其中最重要的就是将所有的Pool层都用stride=2的深度可分离卷积进行了替换；

(3) **上采样策略精细化**。前三板的最终输出都是直接通过双线性插值直接恢复到输入大小。V3+中在双线性插值的基础上借鉴了Encoder-Decoder的结构：首先将特征图双线性插值上采样4倍，与浅层特征空间大小一致的特征图（ResNet中为Conv2，Xception中为第二个stride=2的输出）进行结合，这种策略能够得到更加准确的物体轮廓信息。特征融合的过程为：浅层特征先经过1×1卷积缩减通道数，然后与4倍上采样的特征图concate之后经过3×3卷积融合，最后再进行一次4倍的双线性插值得到最终的输出。

**4.2 各版本的优劣比较**

**4.2.1 比较**

(1) **V1与FCN**：相比之前FCN框架下的语义分割算法，DeepLab系列最重要的改进就在于引入空洞卷积机制来平衡网络感受野不断加大过程中丢失的空间信息，同时对于每一个点都增大了上下文信息。DeepLab中还引入了score map的后处理阶段，用CRF对网络上采样得到的输出运用概率图模型进一步的精炼。

(2) **V1与V2**：用空洞卷积解决了空间信息损失的问题之后，语义分割还面临的一个问题是在同一个场景下同类物体可能会表现出不同的尺度。V2用ASPP模块提取多尺度的特征解决这个问题。

(3) **V2与V3**：V3主要是对ASPP模块的完善。V2中用不同扩张率的空洞卷积来提取不同尺度的特征。但是V3中提出，当扩张率过大时卷积核所能覆盖到的有效特征会减少，极端情况下空洞卷积会退化成1×1卷积。针对这个问题V3删除掉了扩张率最大的支路用1×1卷积代替，同时增加了全局平均池化的支路引入全局信息。

(4) **V3与V3+**：当主要的模块（空洞卷积、ASPP）都完善之后，V3+引入了更强大的网络backbone，并且对最后的网络上采样策略进行了完善，引入了Encoder-Decoder的结构，提高输入对轮廓信息的恢复能力。

**4.2.2 个人想法**

1、**DeepLab系列总结**：DeepLab从V1到V3+始终是围绕着**综述中的四点**在改进，V2和V3过渡阶段提出并改进了ASPP，V3+将前面工作的精华都结合起来之后得到了最终的DeepLab形态。

2、**V3/V3+为什么是stride从8变为16？** 空洞卷积需要在高分辨率的特征图上进行操作V1和V2的stride都为8，高分辨率的特征图会带来超大的计算量。但随着网络backbone的不断升级，ASPP的提出并完善，DeepLab的性能不断增强也就不需要这么高的特征图分辨率（同时在这个过程中CRF也变得没有意义），在保证一个比较好的性能的同时，将stride变为16能够加快计算。

3、**为什么是Xception？** 类似2中的理由，空洞卷积需要的高分辨率特征会加大网络的计算量，Xception网络在性能提升的时候引入了深度可分离卷积可以进一步减小DeepLab的计算量。

**4.3 关于改进的思考**

(1) **用可变卷积替代空洞卷积。**空洞卷积的初衷是为了在不过多损失空间信息的前提下保证感受野不变，但空洞卷积只能按照固定形状对感受野进行扩张。可变卷积在扩张卷积kernel感受野的时候还可以根据图像的信息进行形状的自适应，对于轮廓不规整的物体应该会有性能提升。

(2) DeepLab的作者在之后的研究中还尝试了运用**NAS**来得到比更有效的ASPP架构。

**5 PSPNet(CVPR2017)**

**5.1 原理**

PSPNet的结构与DeepLabV2类似，都是想要扩大每个位置的感受野，PSPNet采用的也是ResNet101+空洞卷积的结构。论文中提到了CNN中实际的经验感受野要比理论感受野要小，PSPNet中提出了一个多路并行的多尺度特征提取模块（Pyramid Pooling Module），通过对输入特征图进行不同程度的下采样，然后用双线性插值上采样到原feature大小，并与输入concate起来经过卷积层融合最终上采样到输入大小。

**5.2 与ASPP对比**

PPM与ASPP在同一年发布，在结构上非常像，但有三点区别：

(1) **扩大感受野的途径不一样。**ASPP通过在kernel中插0来扩大kernel覆盖的面积从而扩大感受野，PPM是直接对feature map进行下采样。这两种方式的差别在于空洞卷积会照顾到更多的原feature map的值，而对feature map直接进行下采样的过程中会直接损失掉部分空间信息，而这部分空间信息在空洞卷积中是能计算到的。

(2) **特征融合的方式不一样**。ASPP只把模块各支路的输出concate起来融合，而PPM还会把模块输入的feature map一起concate。

(3) **出发点不一样**。从PPM的特征融合的方式就能看出来，ASPP的出发点是对原特征图进行多尺度的特征提取，而PPM的出发点在原特征图的基础上加入不同程度的全局信息，所以需要将原特征图一起concate。

**5.3 关于改进的思考**

PPM这个模块也是在stride=8的高分辨率特征图上运行计算的，同样也面临着计算量大的问题，对于这一点同样可以考虑引入深度可分离卷积来减小一部分的计算量。

**6 EncNet(CVPR2018)**

**6.1 原理**

DeepLab以及PSPNet中都是通过扩大感受野来融入更多的全局信息，但是EncNet中提出“扩大感受野不等于引入全局信息”。EncNet做了两件事：

(1) 提出**CEM( Context Encoding Module )模块**来提取**全局信息**：这个全局信息表示的是输入中有哪些类别出现。CEM模块输入为（H\*W\*C）的feature map，输出为（1\*1\*C）的权重向量，用这个输出对模块输入进行加权。在这个过程中该图片的全局信息得到引入。并且Encode模块会学习一个固有的codebook，该模块是对输入与codebook之间的残差进行处理，模块的计算是一个类似于softmax的过程。

(2) **SE-Loss来辅助训练**。SE-Loss通过CEM模块Encode输出外接FC支路来对图片中的物体进行预测，作为一个正则项来辅助主网络的训练。并且由于这是一个分类过程对图片中出现的各类物体大小不敏感，能够规避多尺度的问题。

**6.2 关于EncNet的思考和疑惑**

EncNet对全局信息的引入跳出了早期工作的思维框架，通过对输入图像的整体分类来得到物体的信息，从而来对指导网络训练，并且对得到的score map在上采样之前进行逐类的加权。

**疑惑：**

在初读论文时，先入为主的认为CEM的输入维度中的C等于类别数，这个模块受启发于SE-Net但是变得更具解释性和针对性，但是细品之后发现论文中并没有严格的指明C=num\_class，而是用class-dependent来概述，通过GitHub上的代码确认CEM的模块输入中的C确实不是等于类别数。在这样的参数设置下，个人觉得论文的motivation就被削弱了，虽然最后的效果好，但是效果好的原因完全有可能是CEM包装下的**通道注意力机制**在发挥作用。