

总结-空洞卷积(Dilated/Atrous Convolution)

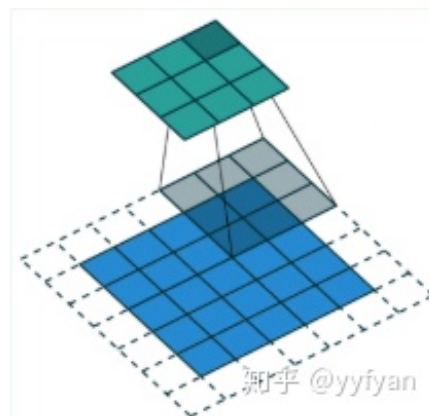
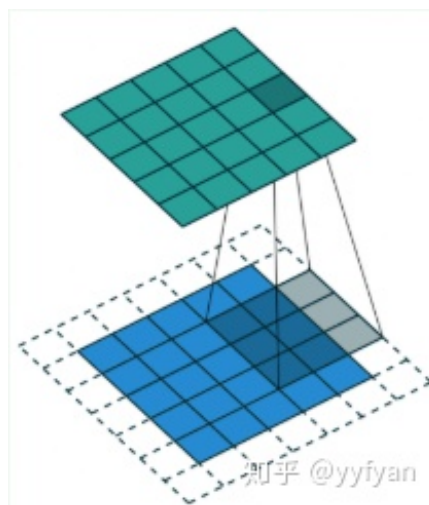
知 zhuanlan.zhihu.com/p/50369448

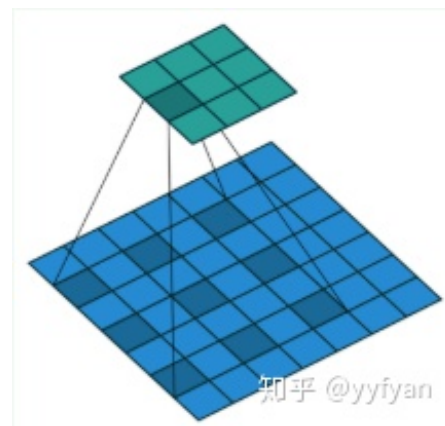
引言

空洞卷积(Dilated/Atrous Convolution)，广泛应用于语义分割与目标检测等任务中，语义分割中经典的deeplab系列与DUC对空洞卷积进行了深入的思考。目标检测中SSD与RFBNet，同样使用了空洞卷积。

标准卷积：以 3×3 为例，以下分辨率不变与分辨率降低的两个实例；

空洞卷积：在 3×3 卷积核中间填充0，有两种实现方式，第一，卷积核填充0，第二，输入等间隔采样。





空洞卷积的作用

空洞卷积有什么作用呢？

- **扩大感受野**：在deep net中为了增加感受野且降低计算量，总要进行降采样（pooling或s2/conv），这样虽然可以增加感受野，但空间分辨率降低了。为了能不丢失分辨率，且仍然扩大感受野，可以使用空洞卷积。这在检测，分割任务中十分有用。一方面感受野大了可以检测分割大目标，另一方面分辨率高了可以精确定位目标。
- **捕获多尺度上下文信息**：空洞卷积有一个参数可以设置dilation rate，具体含义就是在卷积核中填充dilation rate-1个0，因此，当设置不同dilation rate时，感受野就会不一样，也即获取了多尺度信息。**多尺度信息在视觉任务中相当重要啊。**

从这里可以看出，空洞卷积可以任意扩大感受野，且不需要引入额外参数，但如果把分辨率增加了，算法整体计算量肯定会增加。

ps: 空洞卷积虽然有这么多优点，但在实际中不好优化，速度会大大折扣。

空洞卷积感受野如何计算

说了这么多有关感受野的话，感受野究竟怎么计算呢？其实和标准卷积是一致的。

空洞卷积实际卷积核大小：

$K = k + (k-1)(r-1)$ ，k为原始卷积核大小，r为空洞卷积参数空洞率；

以三个 $r=2$ 的 3×3 空洞卷积为例计算感受野

$$\begin{aligned} K &= k + (k-1)(r-1) = 3 + 2 \times 1 = 5 \\ R &= 1 + 4 + 4 + 4 = 13 \end{aligned}$$

感受野 (Receptive Field) 是个相对概念，指神经网络中某层神经看到的“输入区域，feature map上某个元素的计算受输入图像上某个区域的影响，这个区域即是该元素的感受野。

Receptive Field = center + surround (都需要递推计算)

而语义分割由于需要获得较大的分辨率图，因此经常在网络的最后两个stage，取消降采样操作，之后采用空洞卷积弥补丢失的感受野。

以语义分割中常使用的VGG和ResNet50为例，计算其空洞前与空洞后的感受野。

VGG16：将FC6层使用7*7卷积替换，其他不变，此版本我们称为VGG_Conv，根据deeplabv1的设置，我们使用DeepLab-CRF-7x7，即取消pool4和pool5的降采样操作，改为了3*3/s1，同时将 conv5_1-conv5_3使用r=2的空洞卷积。FC的空洞率r=4。

VGG_Conv : $R=1+6) * 2+2+2+2) * 2+2+2+2) * 2+2+2+2) * 2+2+2) * 2+2+2=404$
 DeepLab-CRF-7x7 :
 $R=1+24) + 2+4+4+4) + 2+2+2+2) * 2+2+2+2) * 2+2+2) * 2+2+2=412$

两者和感受野近似相等，从deeplab实验结果可以看出，感受野不一定和之前完全一样，但感受野大了效果会好一些。表中结果和自己计算的不一样，不知道表中感受野怎么计算的。。

Method	kernel size	input stride	receptive field	# parameters	mean IOU (%)	Training speed (img/sec)
DeepLab-CRF-7x7	7×7	4	224	134.3M	67.64	1.44
DeepLab-CRF	4×4	4	128	65.1M	63.74	2.90
DeepLab-CRF-4x4	4×4	8	224	65.1M	67.14	2.90
DeepLab-CRF-LargeFOV	3×3	12	224	20.5M	67.64	4.84

结论：空洞前与空洞后感受野可以不一致，但空洞后的感受野不要小于之前的，这样其实是不影响网络精调的，因为参数没有变。恍然大悟啊！

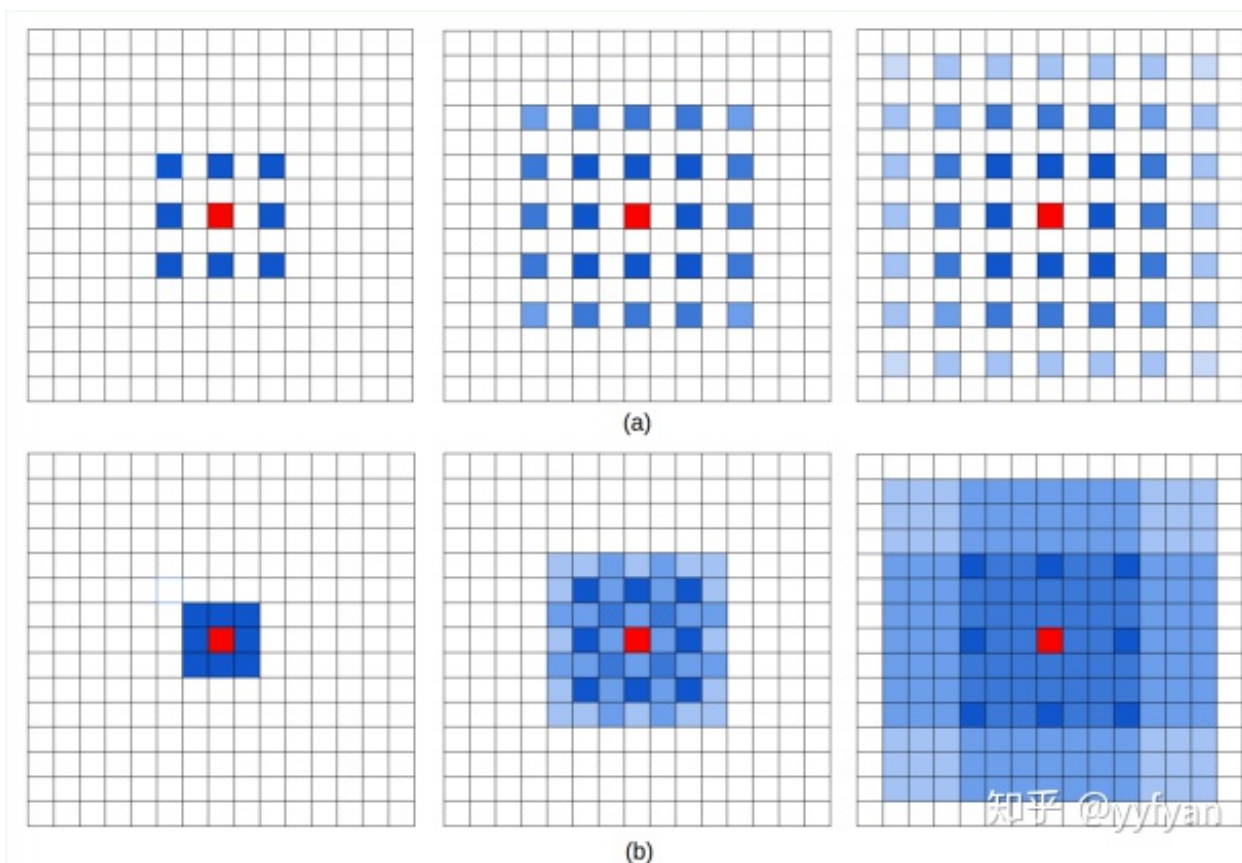
空洞卷积gridding问题

是的，空洞卷积是存在理论问题的，论文中称为gridding，其实就是网格效应/棋盘问题。因为空洞卷积得到的某一层的结果中，邻近的像素是从相互独立的子集中卷积得到的，相互之间缺少依赖。

- **局部信息丢失**：由于空洞卷积的计算方式类似于棋盘格式，某一层得到的卷积结果，来自上一层的独立的集合，没有相互依赖，因此该层的卷积结果之间没有相关性，即局部信息丢失。
- **远距离获取的信息没有相关性**：由于空洞卷积稀疏的采样输入信号，使得远距离卷积得到的信息之间没有相关性，影响分类结果。

解决方案

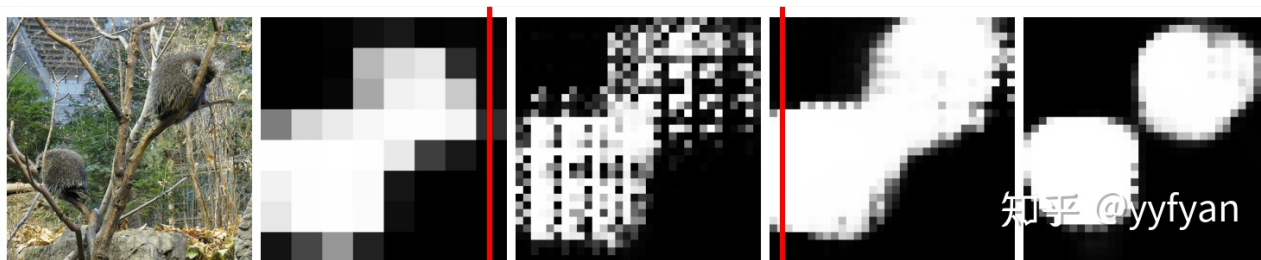
Panqu Wang, Pengfei Chen, *et al.* **Understanding Convolution for Semantic Segmentation.** // WACV 2018



通过图a解释了空洞卷积存在的问题，从左到右属于top-bottom关系，三层卷积均为 $r=2$ 的dilatedConv,可以看出最上层的红色像素的感受野为13且参与实际计算的只有75%，很容易看出其存在的问题。

使用HDC的方案解决这个问题，不同于采用相同的空洞率的deeplab方案，**该方案将一定数量的layer形成一个组，然后每个组使用连续增加的空洞率，其他组重复。**如deeplab使用 $rate=2$,而HDC采用 $r=1, r=2, r=3$ 三个空洞率组合，这两种方案感受野都是13。但HDC方案可以从更广阔的像素范围获取信息，避免了grid问题。同时该方案也可以通过修改rate任意调整感受野。

Fisher Yu, *et al.* **Dilated Residual Networks.** //CVPR 2017

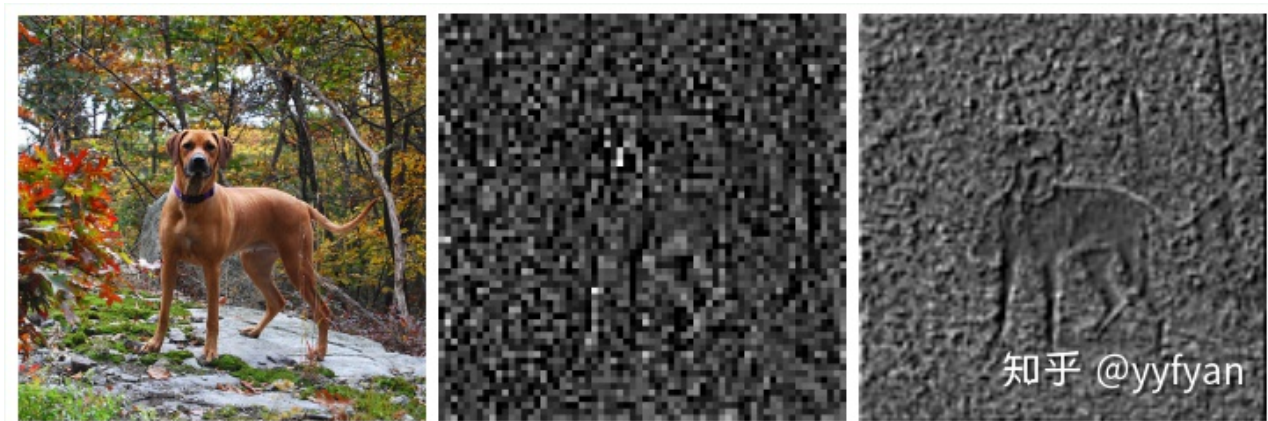


如果特征map有比空洞率更高频的内容，则grid问题更明显。

解决空洞率、网格效应等问题的三个方法：

提出了三种方法：

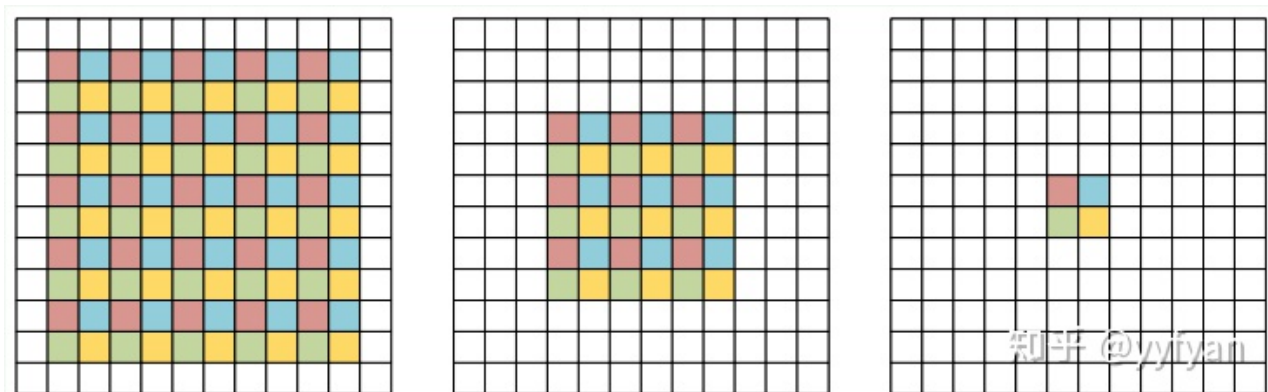
Removing max pooling : 由于maxpool会引入更高频的激活，这样的激活会随着卷积层往后传播，使得grid问题更明显。



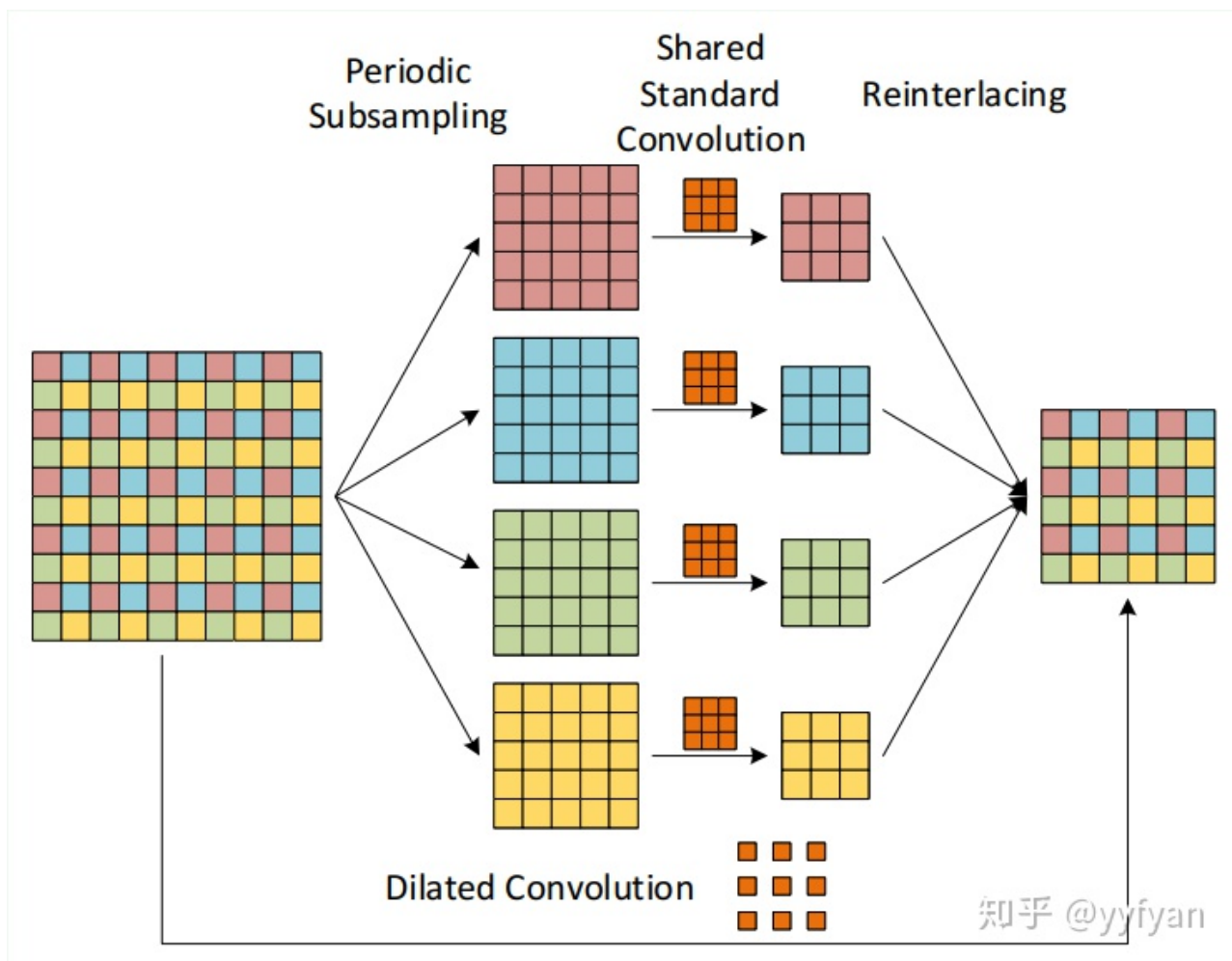
Adding layers : 在网络最后增加更小空洞率的残参block, 有点类似于HDC。

Removing residual connections : 去掉残参连接，防止之前层的高频信号往后传播。

Zhengyang Wang, *et al.* **Smoothed Dilated Convolutions for Improved Dense Prediction.** // KDD 2018.

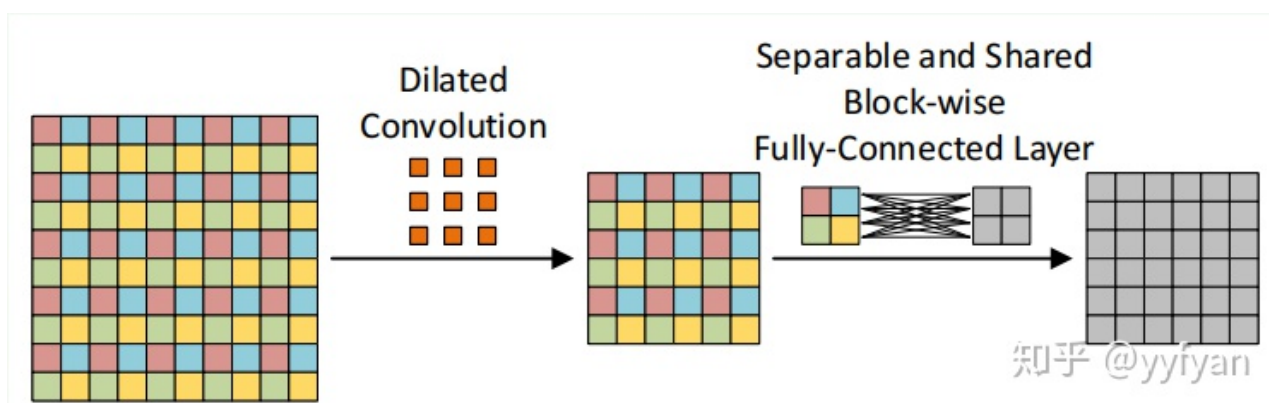
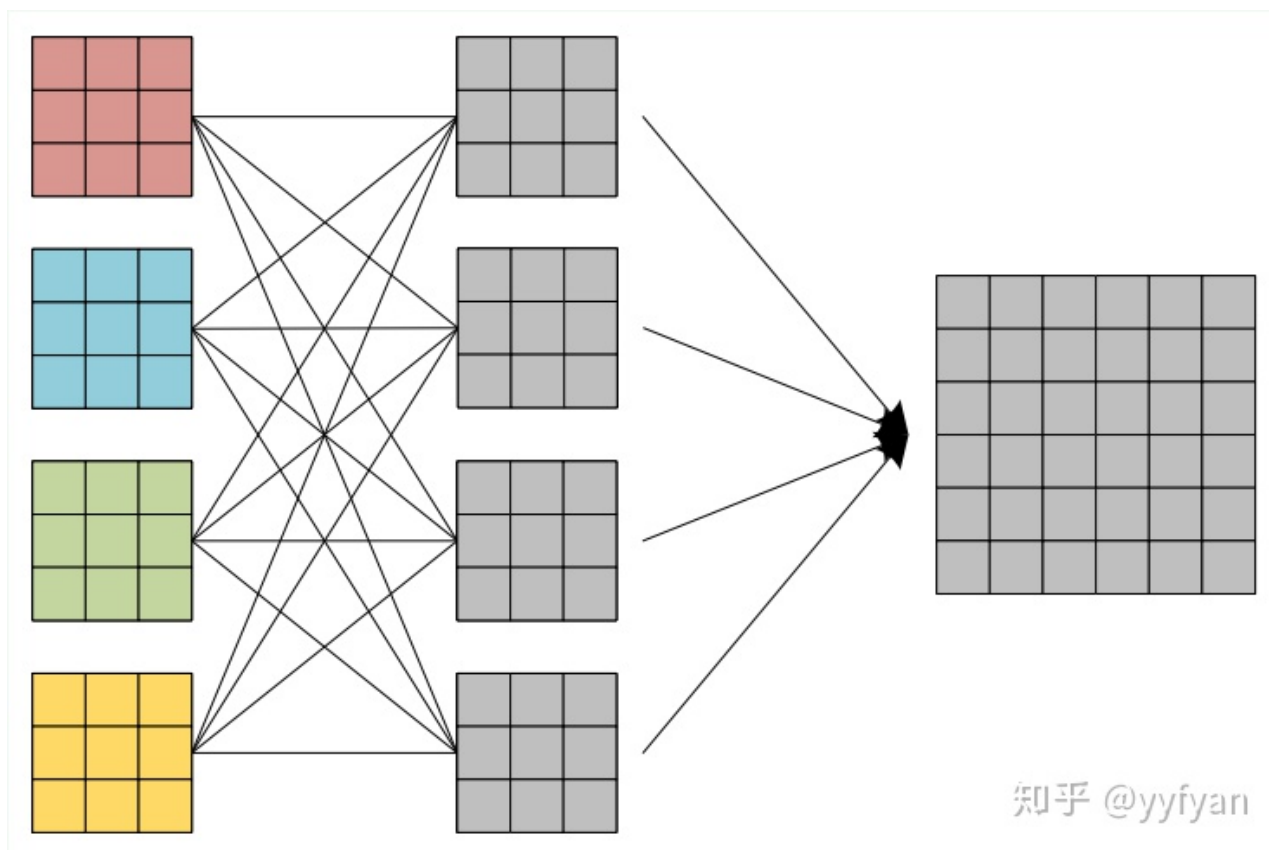


空洞卷积的分解观点，在原始特征图上周期性采样形成4组分辨率降低的特征图，然后使用原始的空洞卷积参数(去掉了空洞o)分别进行卷积，之后将卷积的结果进行上采样组合。从该分解观点可以看出，卷积前后的4个组之间没有相互依赖，使得收集到不一致的局部信息。

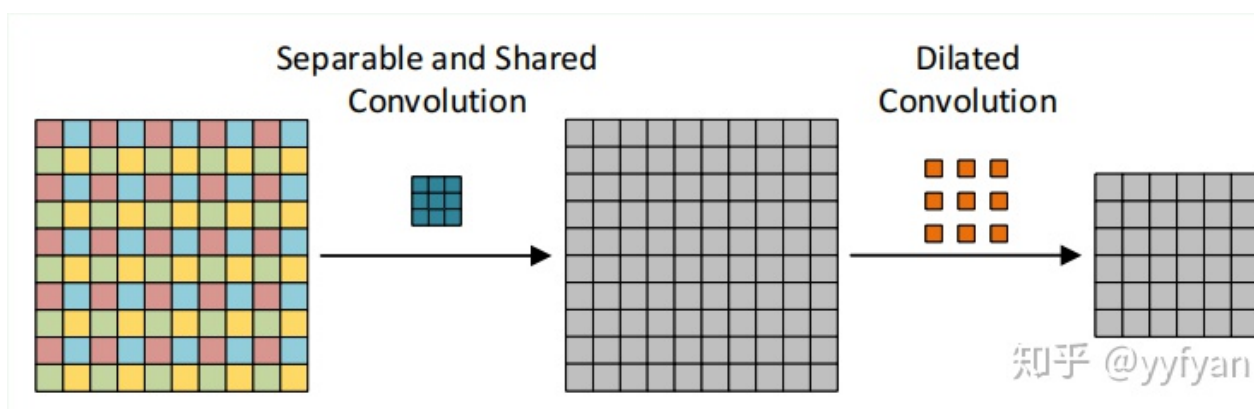
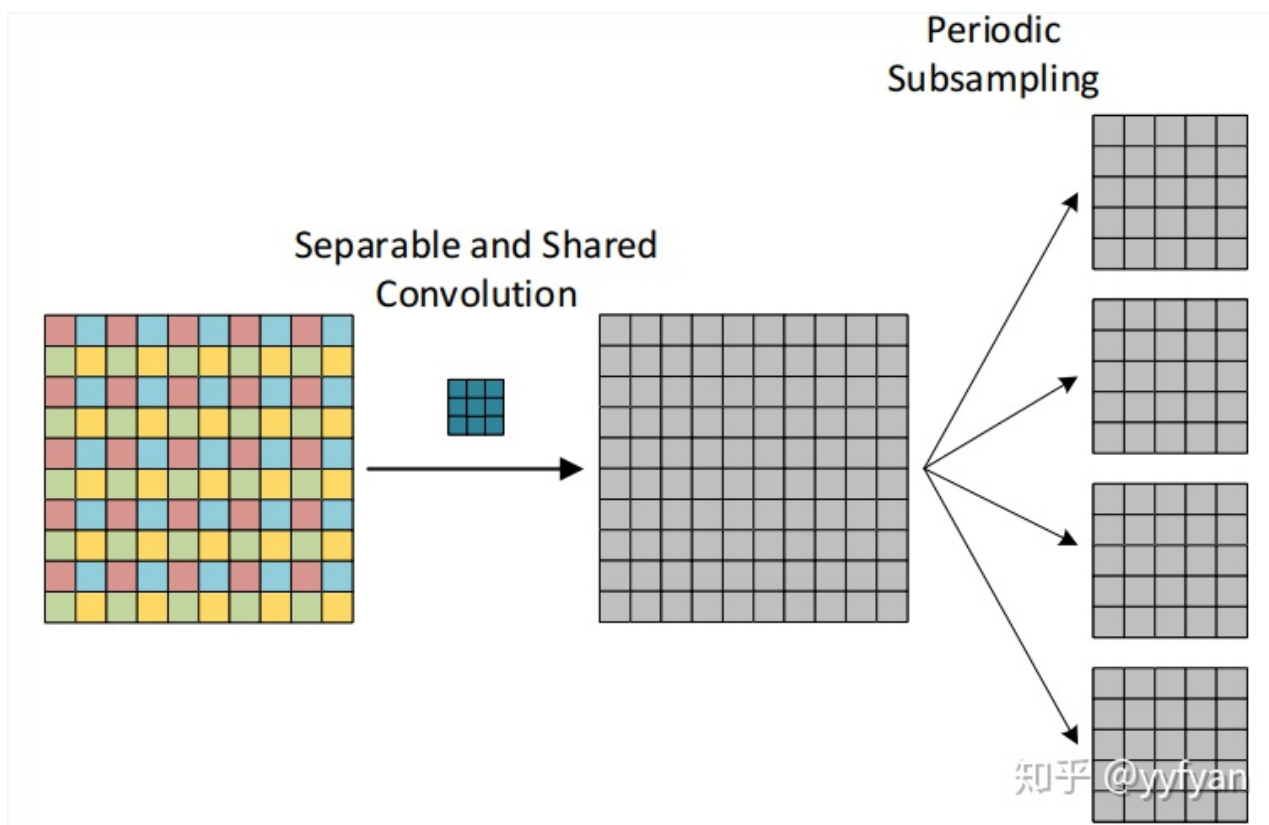


从上面分解的观点出发：

(1) 在最后生成的4组卷积结果之后，*经过一层组交错层*，类似于ShuffleNet，使得每组结果能进行相互交错，相互依赖，以此解决局部信息不一致的问题。

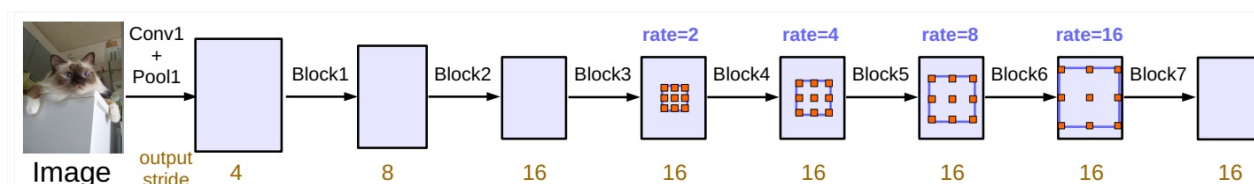


(2) 第二种方法为在空洞卷积之前进行局部信息依赖，即增加一层卷积操作，卷积利用了分离卷积，并且所有通道共享参数。



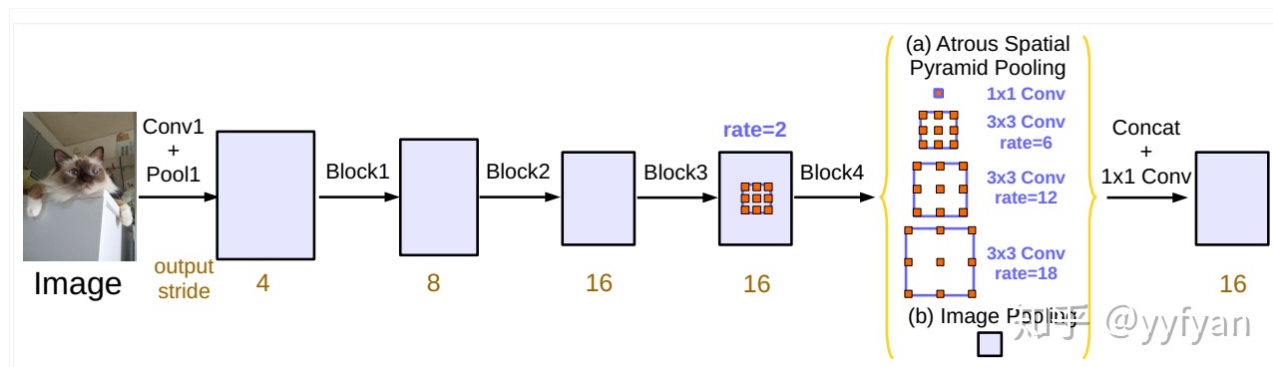
Liang-Chieh Chen, *et al.* **Rethinking Atrous Convolution for Semantic Image Segmentation**//2017

deeplabv3在v2基础上进一步探索空洞卷积，分别研究了级联ASPP与并联ASPP两种结构。



deeplabv3不同于deeplabv2，在resnet101基础上级联了更深的网络，随着深度的增加，使用了不同的空洞率的卷积，这些卷积保证分辨率不降低的情况下，感受野可以任意控制，一般让空洞率成倍增加。同时使用了Multigrid策略，在同一个blocks的不同层使用

分层的空洞率，如2,4,8，而不是都使用2，这样使得感受野相比原来的有所增加。**但这样同样会存在grid问题。**

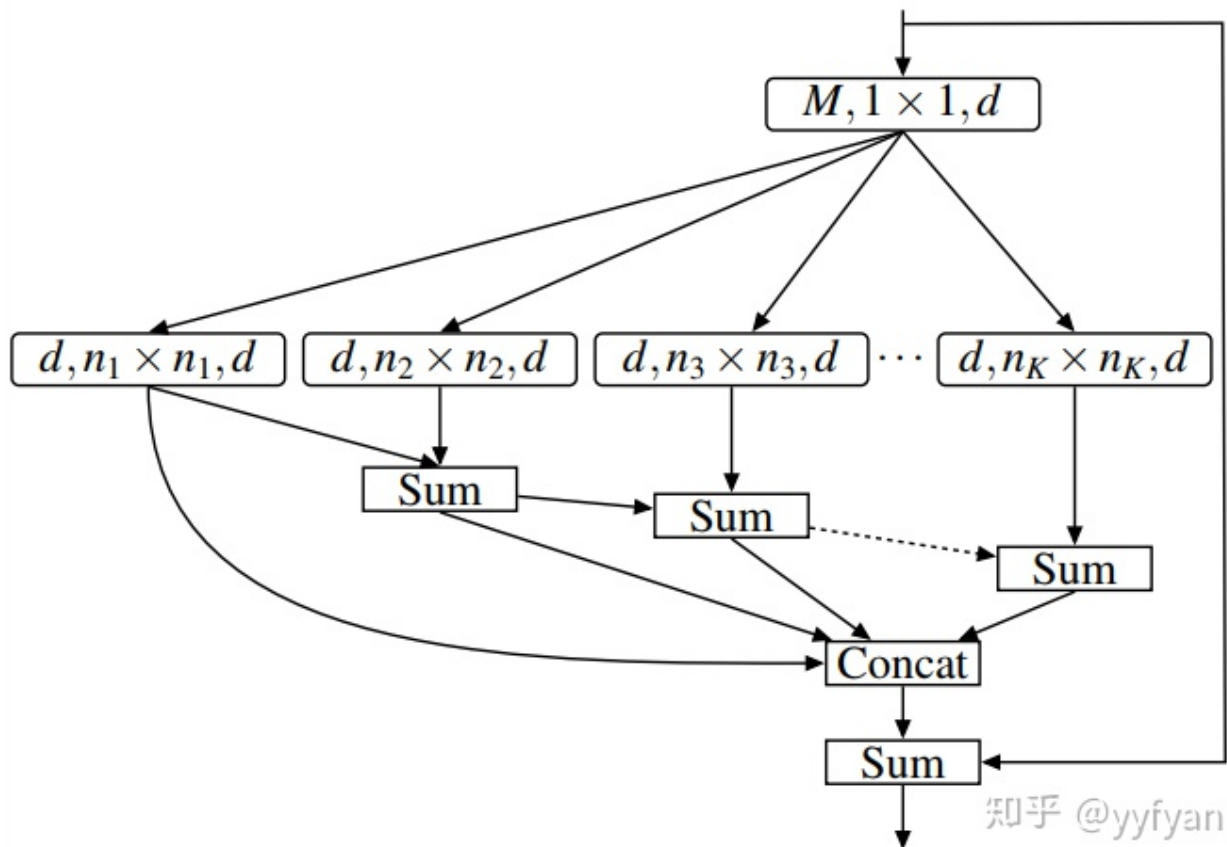


ASPP存在的问题，当使用的空洞率增大时，有效的滤波参数数量逐渐减小。极端的，当r等于特征图大小时，该卷积没有捕获整幅图像的上下文信息，而是退化为1*1卷积。

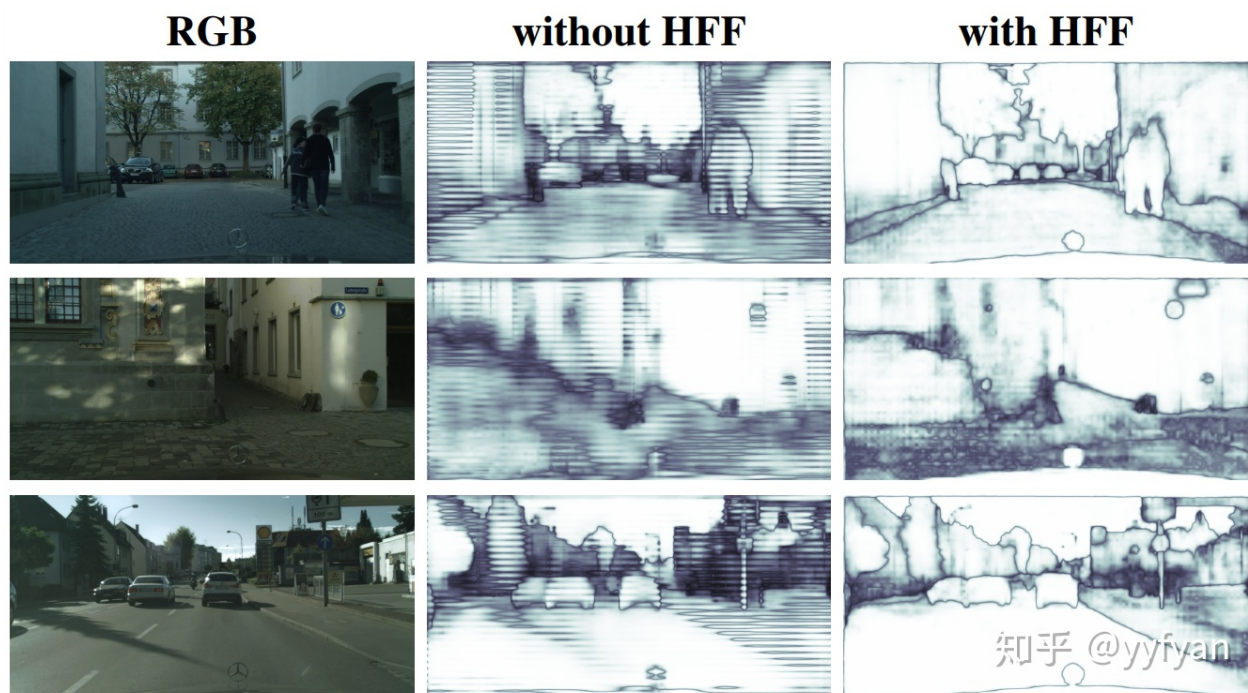
解决方案：**增加图像级特征，使用全局池化获取图像全局信息**，而其他部分的卷积为了捕获多尺度信息，这里的卷积不同于deeplabv2，加了batch normalization。

Sachin Mehta, *et al.* **ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation.** //ECCV 2018

ESPNet利用分解卷积的思想，先用1*1卷积将通道数降低减少计算量，后面再加上基于空洞卷积的金字塔模型，捕获多尺度信息。



之前的方法都是通过引入新的计算量，学习新的参数来解决grid问题。而这里直接使用了特征分层的思想直接将不同rate的空洞卷积的输出分层sum，其实就是将不同的感受野相加，弥补了空洞带来的网格效应。从结果上看效果不错。

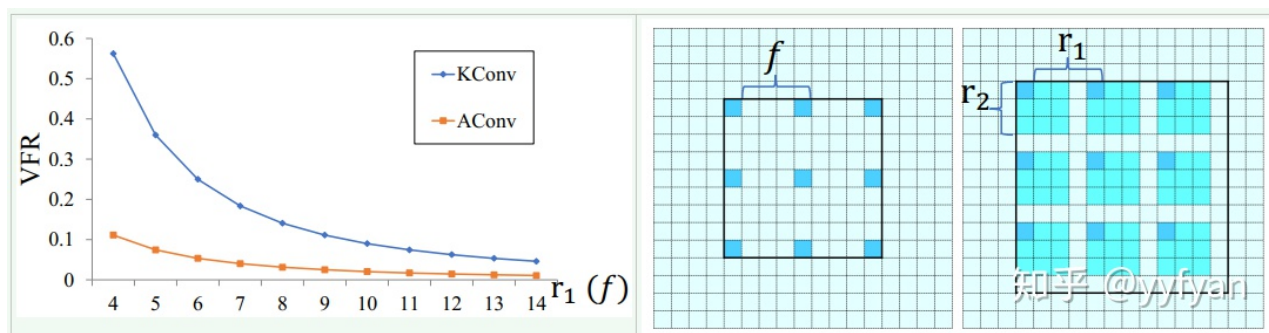


训练技巧：

所有卷积后都使用BN和PReLU，*很多实时分割小网络都使用了PReLU*；

使用Adam训练，很多小网络使用这个；

Tianyi Wu, *et al.* Tree-structured Kronecker Convolutional Networks for Semantic Segmentation. // ICME 2019



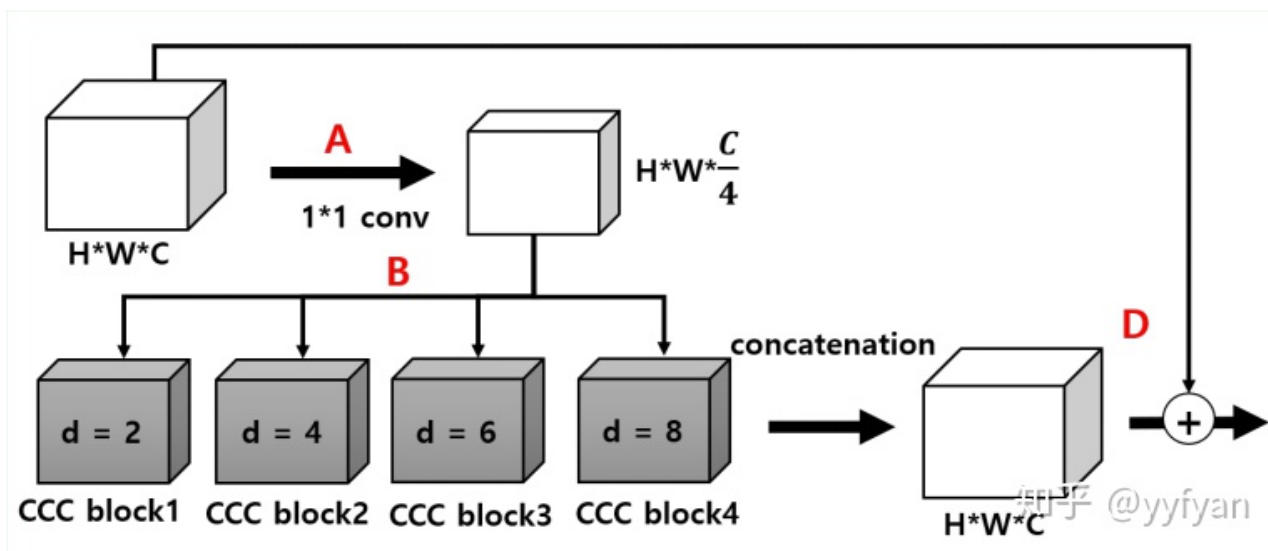
使用Kronecker convolution来解决空洞卷积局部信息丢失问题，以 $r_1=4$ 、 $r_2=3$ 为例，KConv将每个标准卷积的元素都乘以一个相同的矩阵，该矩阵由0,1组成，这样参数量是不增加的。该矩阵为：

这样每个元素乘以矩阵后变为上面右图所示的图。因此，可以看出 r_1 控制空洞的数量，也即扩大了感受野，而 r_2 控制的是每个空洞卷积忽视的局部信息。当 $r_2=1$ 时，其实就是空洞卷积，当 $r_2=r_1=1$ 时就是标准卷积。

总体效果mIOU提升了1%左右。

除此之外，提出了一个TFA模块，利用树形分层结构进行多尺度与上下文信息整合。**结构简单，但十分有效**，精度提升4-5%。

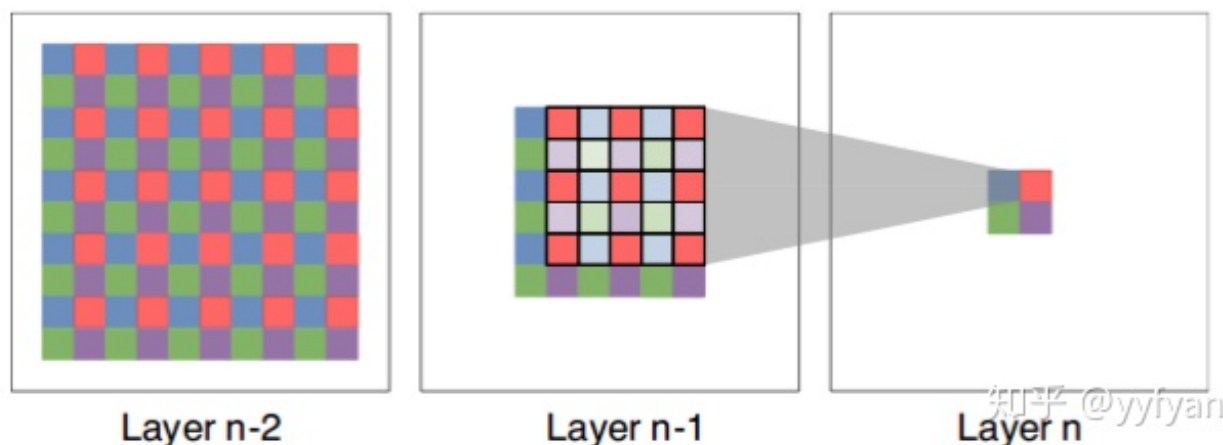
Hyojin Park, *et al.* Concentrated-Comprehensive Convolutions for lightweight semantic segmentation. // 2018



针对实时语义分割提出的网络结构，深度分离卷积与空洞卷积的组合，在ESPNet上做的实验。并且说明简单的组合会带来精度的降低，由于局部信息的丢失。为此，在深度分离空洞卷积之前，使用了两级一维分离卷积捕获局部信息。

网络结构上与ESPNet保持一致，其中，并行分支结果直接Cat，不需要后处理，每个分支不需要bn+relu。消融实验表明，在一维卷积之间加入BN+PReLU，精度会增加1.4%。

Efficient Smoothing of Dilated Convolutions for Image Segmentation.//2019



编辑于 2019-09-03

深度学习 (Deep Learning)

图像分割

卷积神经网络 (CNN)

▲ 赞同 366

▼ ● 34 条评论

🔗 分享

 喜欢

 收藏





赞同 366



分享

文章被以下专栏收录
