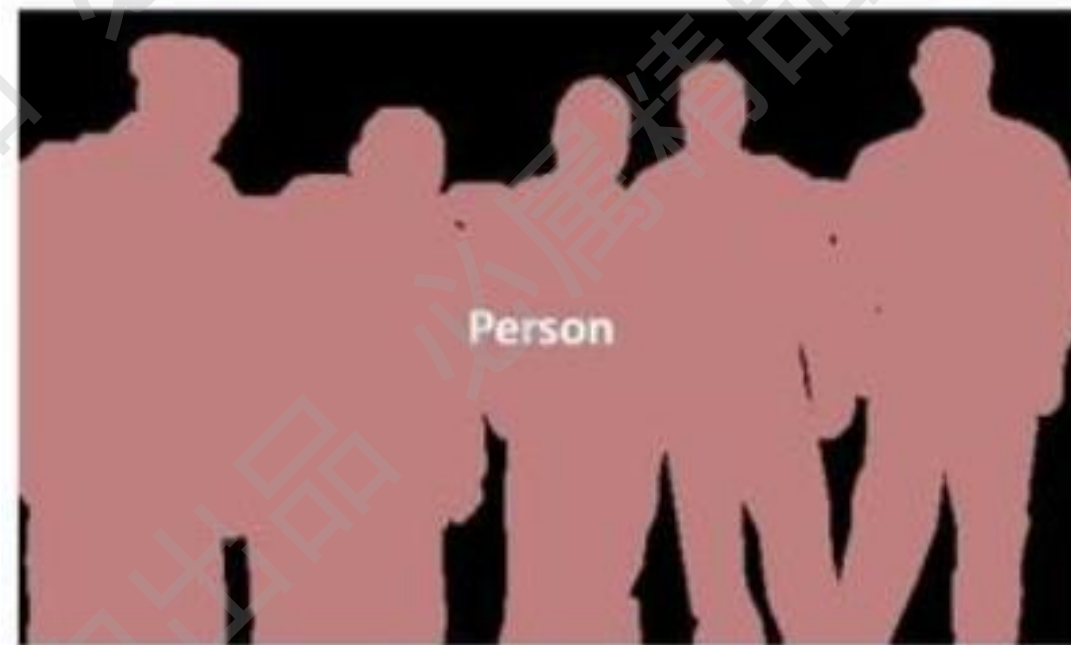


# deeplab系列

✓ 图像分割中的传统做法：

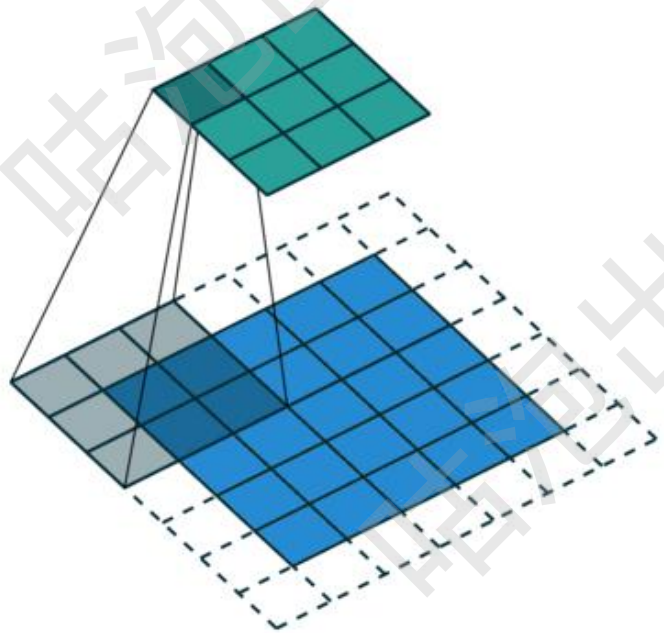
✎ 为了增大感受野，通常都会选择pooling操作，但是也会丢失一部分信息



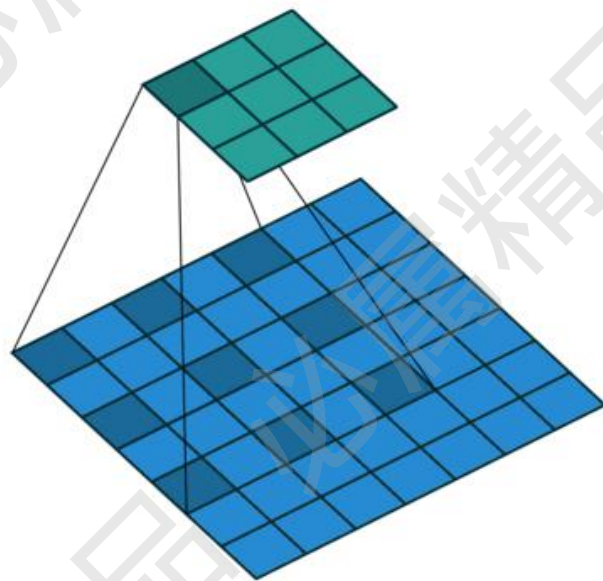
# 语义分割

✓ DeepLab系列方法

✎ 空洞卷积 (dilated convolution) :



(传统卷积)

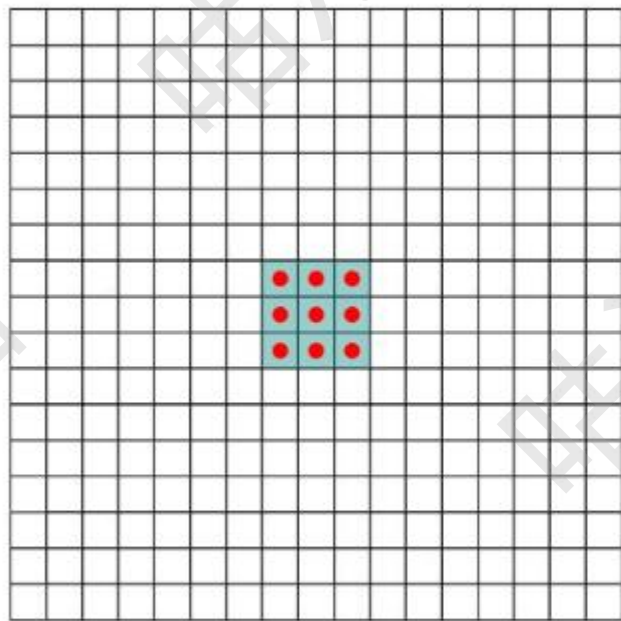


(空洞卷积)

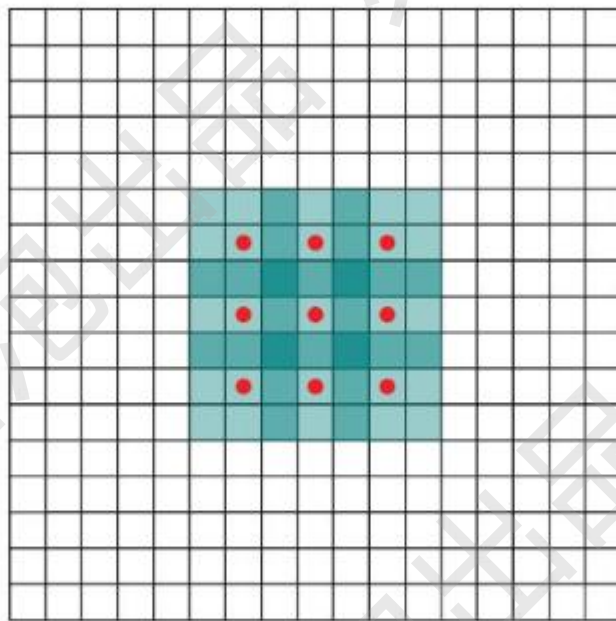
# 语义分割

✓ 空洞卷积 (dilated convolution) :

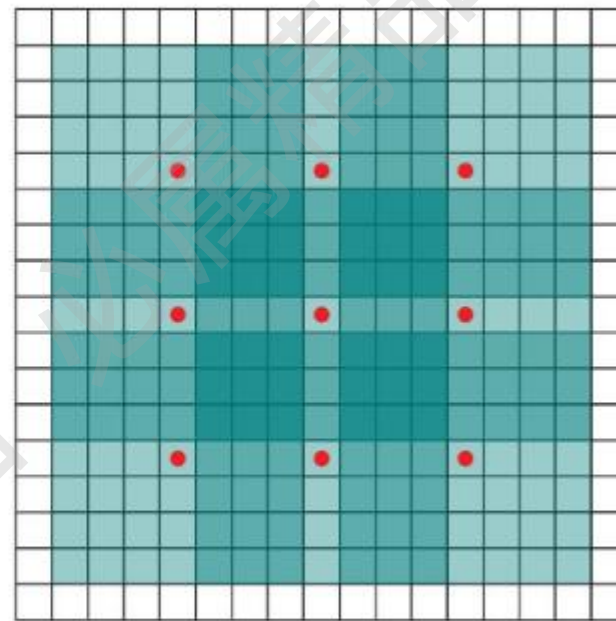
✎ 通过设置dilated参数可以得到不同感受野的特征 ( $3 \times 3$ ,  $7 \times 7$ ,  $15 \times 15$ )



(a)



(b)



(c)

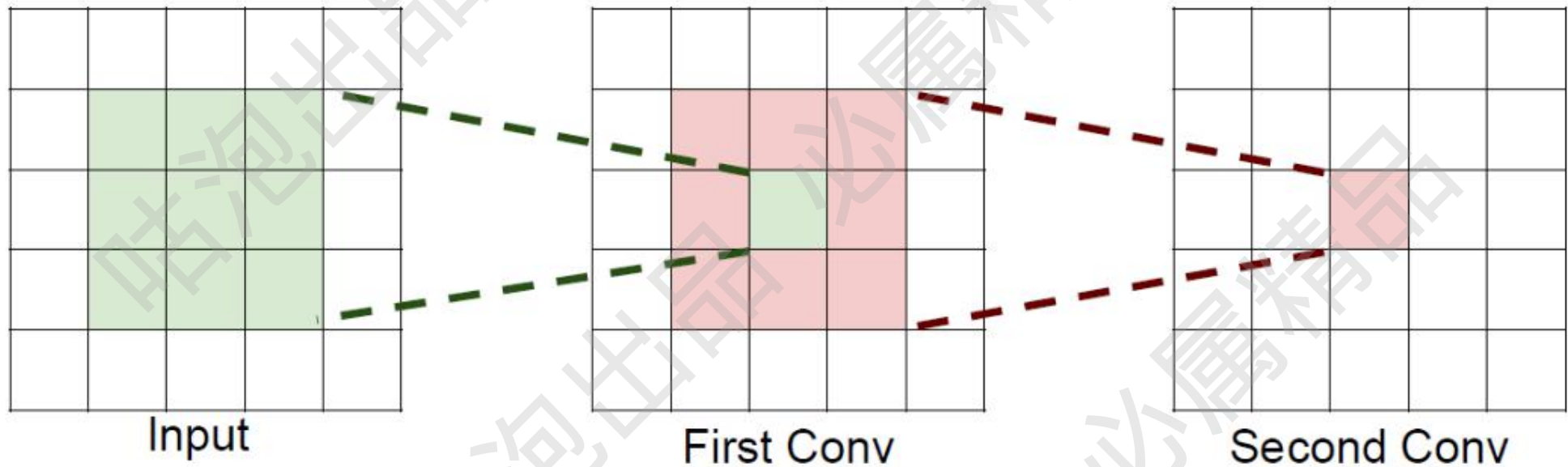
# 语义分割

✓ 空洞卷积的优势：

- ✎ 图像分割任务中（其他场景也适用）需要较大感受野来更好完成任务
- ✎ 通过设置dilation rate参数来完成空洞卷积，并没有额外计算
- ✎ 可以按照参数扩大任意倍数的感受野，而且没有引入额外的参数
- ✎ 应用简单，就是卷积层中多设置一个参数就可以了

# 语义分割

✓ 感受野:



✎ 如果堆叠3个 $3 \times 3$ 的卷积层，并且保持滑动窗口步长为1，其感受野就是 $7 \times 7$ 的了，这跟一个使用 $7 \times 7$ 卷积核的结果是一样的，那为什么非要堆叠3个小卷积呢？



# 语义分割

## ✓ 感受野

✎ 假设输入大小都是 $h \times w \times c$ ，并且都使用 $c$ 个卷积核(得到 $c$ 个特征图)，可以来计算一下其各自所需参数：

一个 $7 \times 7$ 卷积核所需参数：

$$= C \times (7 \times 7 \times C) = \mathbf{49 C^2}$$

3个 $3 \times 3$ 卷积核所需参数：

$$= 3 \times C \times (3 \times 3 \times C) = \mathbf{27 C^2}$$

✎ 很明显，堆叠小的卷积核所需的参数更少一些，并且卷积过程越多，特征提取也会越细致，加入的非线性变换也随着增多，还不会增大权重参数个数，这就是VGG网络的基本出发点，用小的卷积核来完成体特征提取操作。

# 语义分割

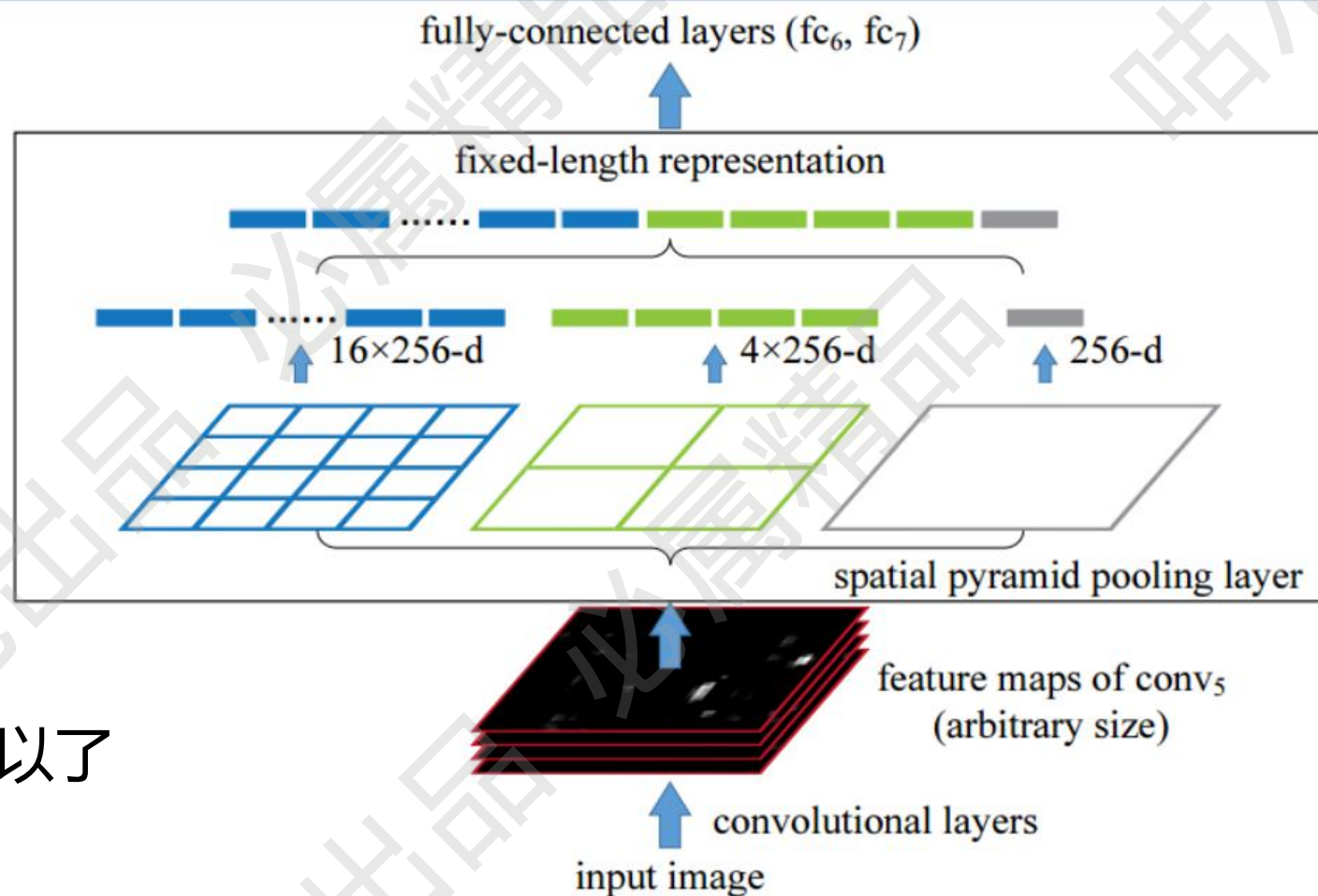
## ✓ SPP-Layer:

✎ 网络中通常要求输入固定

✎ SPP通用不同的池化层

✎ 再进特征拼接

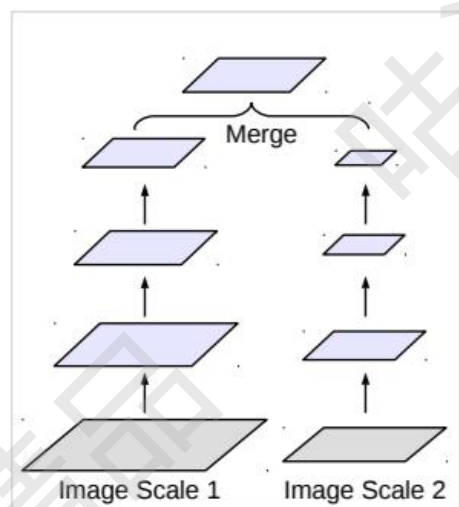
✎ 只要保障输入特征固定就可以了



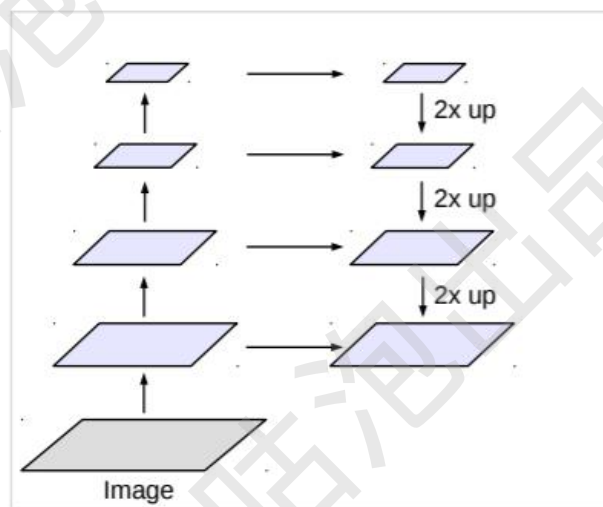
# 语义分割

✓ 常用的多尺度特征提取方法:

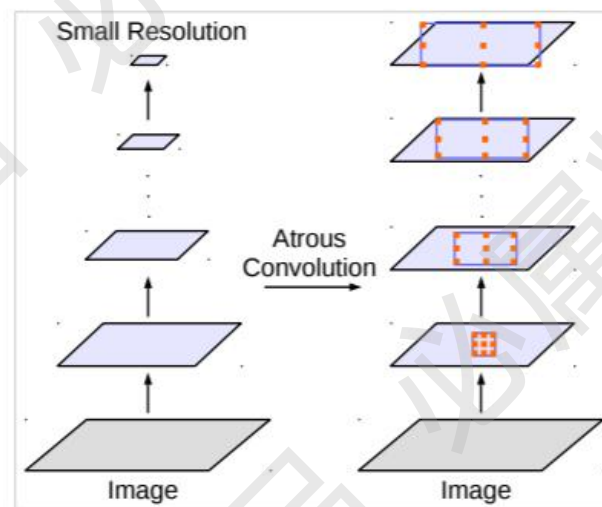
📌 这些方法都比较通用，在各项视觉任务中均可以使用



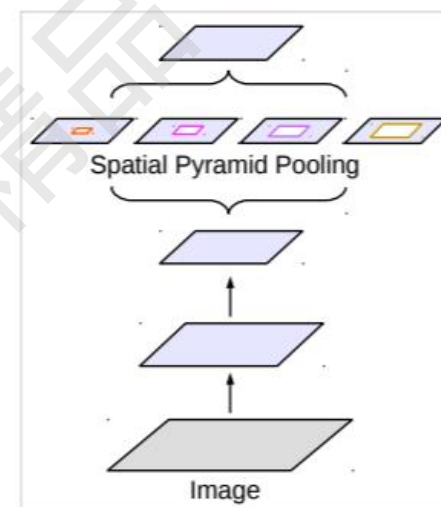
(a) Image Pyramid



(b) Encoder-Decoder



(c) Deeper w. Atrous Convolution



(d) Spatial Pyramid Pooling

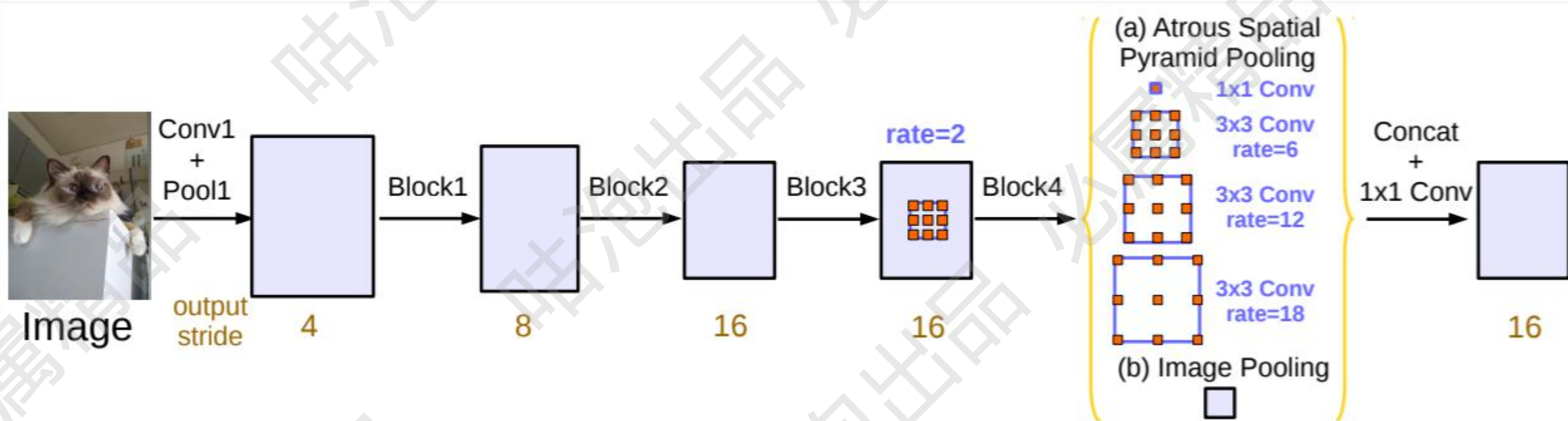
Figure 2. Alternative architectures to capture multi-scale context.



# 语义分割

✓ ASPP (atrous convolution SPP) :

✎ 其实就是跟SPP差不多，只不过引入了不同倍率的空间卷积



# 语义分割

✓ deepLabV3+

✎ 整体网络架构:

✎ 效果提升不算多

✎ Backbone可以改进

✎ 创新不多所以不是4版。。

