语义分割串烧：从入门到弃坑（上）（截至2017）

- 看这鬼东西的起因：
- 1）在看gaitset时发现里面有个金字塔层级块，我当时就懵逼了，这是什么东西啊，没听说过呀，然后兜兜转转发现这和DeepLab的ASPP长得蛮像。
- 2）在看SwinTransformer时，看不懂，有人说其中滑动窗口和FCN思想很像。
- 3）本身对语义分割也蛮感兴趣，于是跑过来看看。结果没想到这玩意这么抽象。

> 这博客写的是真好:https://blog.qure.ai/notes/semantic-segmentation-deep-learning-review#fcn

语义分割概述

语义分割是个什么呢？简单来说，他是像素级别的图像分类，但又和图像分类不太一样：图像分类是把一张图分类打一个标签，而语义分割是把图像中每个像素都进行分类打标签，类似下图不同颜色说明是不同类别

(a) Image (b) Ground Truth

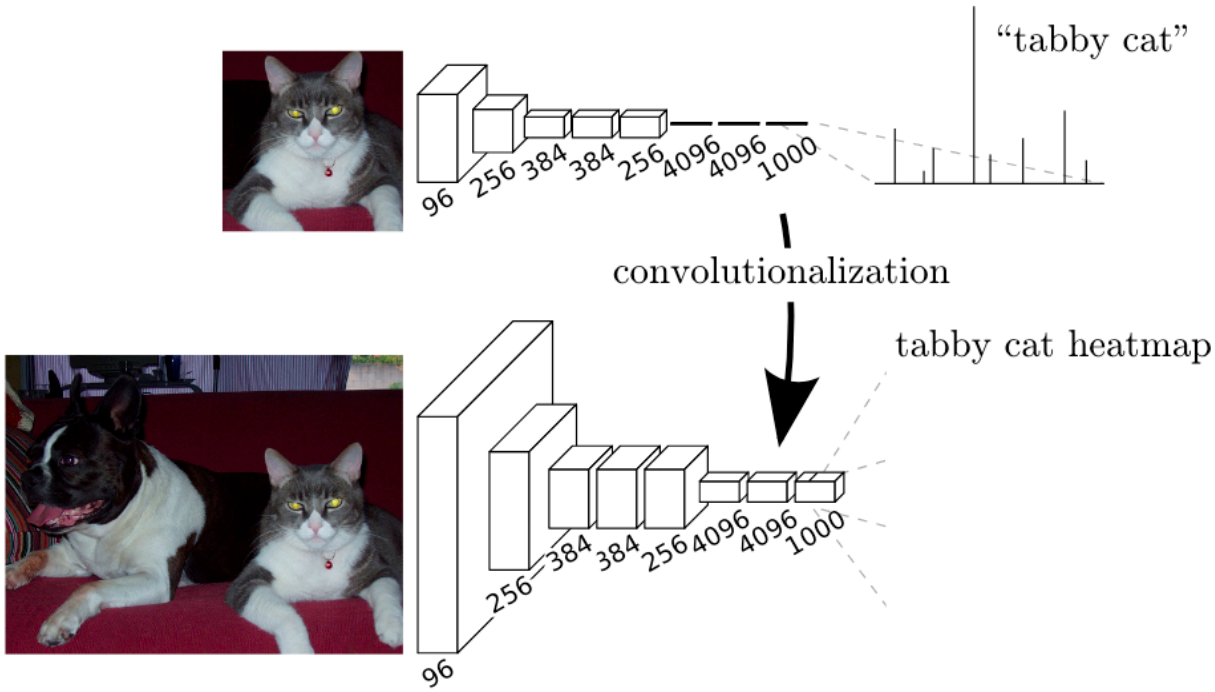
FCN

> 原文：“Fully Convolutional Networks for Semantic Segmentation”（2014）

主要贡献

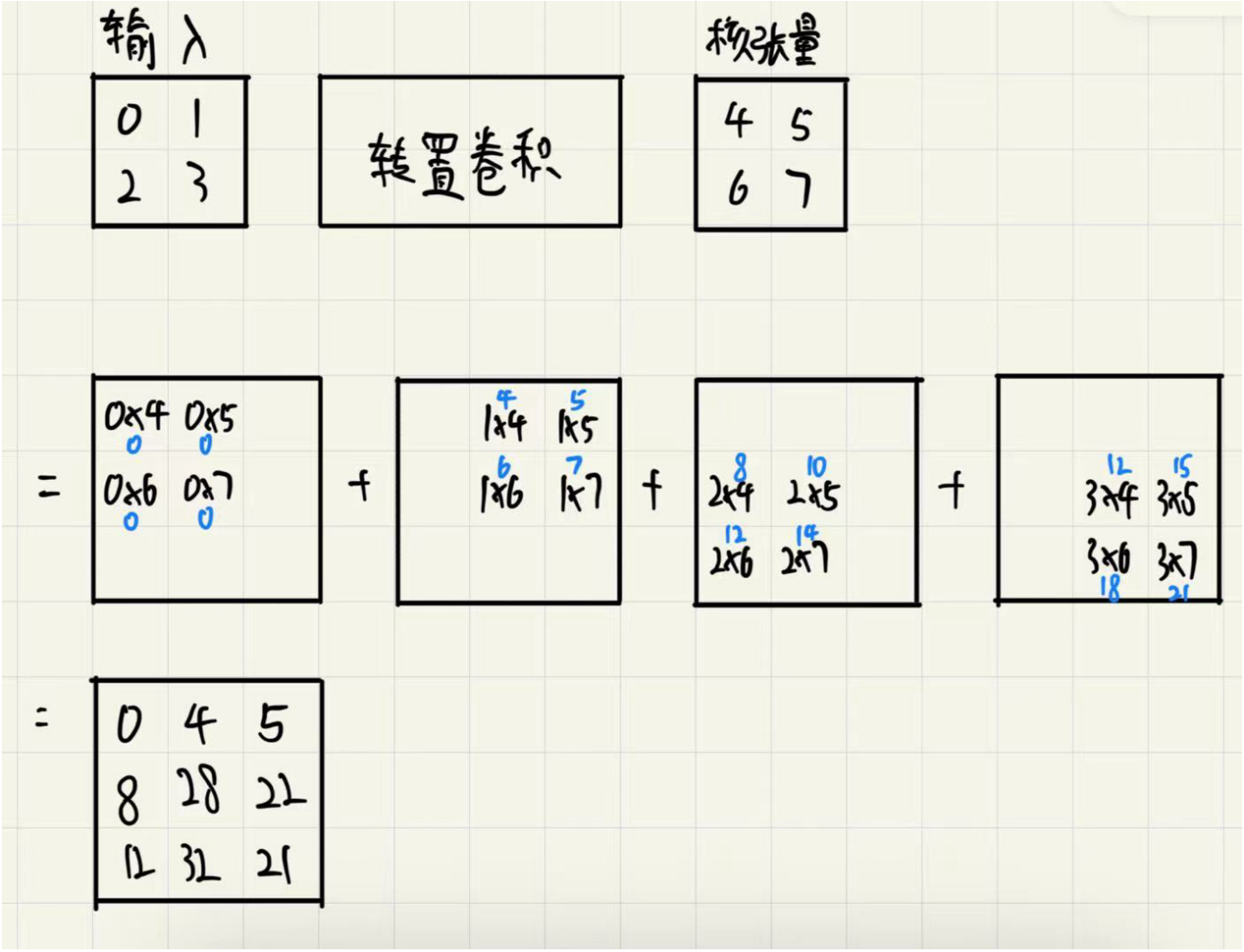
- 使用了完整的end-to-end网络（就是给一个输入，中间你什么都不用管，直接得到一个输出）

- 将图像分类模型迁移到语义分割
- 上采样时不采用双线插值，使用了转置卷积代替了最后的全连接层
- 采用了短路连接来提升上采样效果（类似ResNet的短路）



详解

1）首先是转置卷积。作者认为，全连接层无法很好关注全局特征，并且下采样会降低图像大小，丢失信息，因此在上采样（上采样详情请看https://zhuanlan.zhihu.com/p/428523385）的时候会难以复原像素语义信息，导致重建图像时效果不好。于是作者提出了转置卷积，简单来说，



分别拿输入中的每个元素和核张量所有元素相乘,最后相加得到结果，这样就可以在提升感受野的同时，上采样更好还原像素信息。（相对于双线插值而言）那么你估计又想问了，为什么叫转置呢？我们可以把一个nxk的输入矩阵等效降维成一维1xnk，把卷积后得到的输出矩阵mxk等效降维为mkx1，那么根据矩阵乘法，卷积矩阵等效的大小应该就是nkxmk。那么当我上采样，从mxk还原为nxk时，卷积矩阵等效大小就应该转置为mkxnk。（详情可见这两篇文章https://zhuanlan.zhihu.com/p/158933003和https://zhuanlan.zhihu.com/p/549164774）

2）接着是使用了短路连接。上文也说过，下采样时会丢失像素信息，因此作者想了个妙方法，它在上采样时将上一层的上采样特征图和对应层级的下采样特征图拼接在一起，这样就可以从原先的下采样图像中补充原图像细节特征。

SegNet

> 原文：“SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation”（2015）

[4]

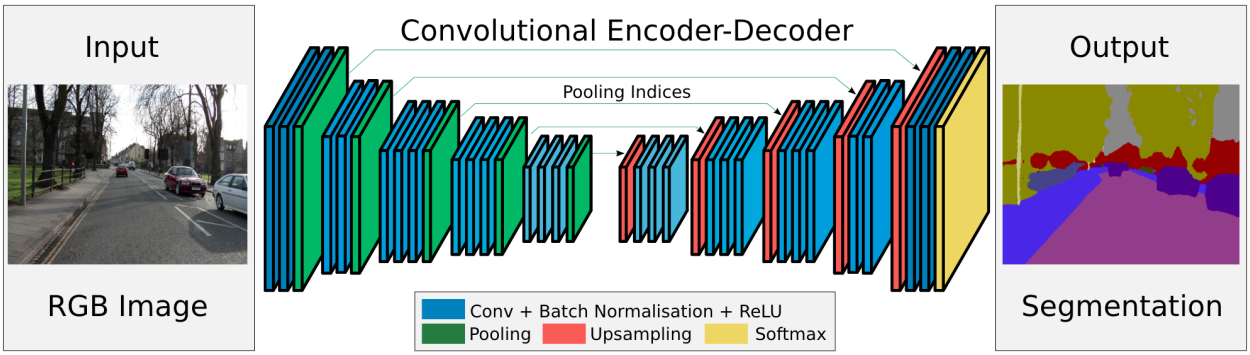


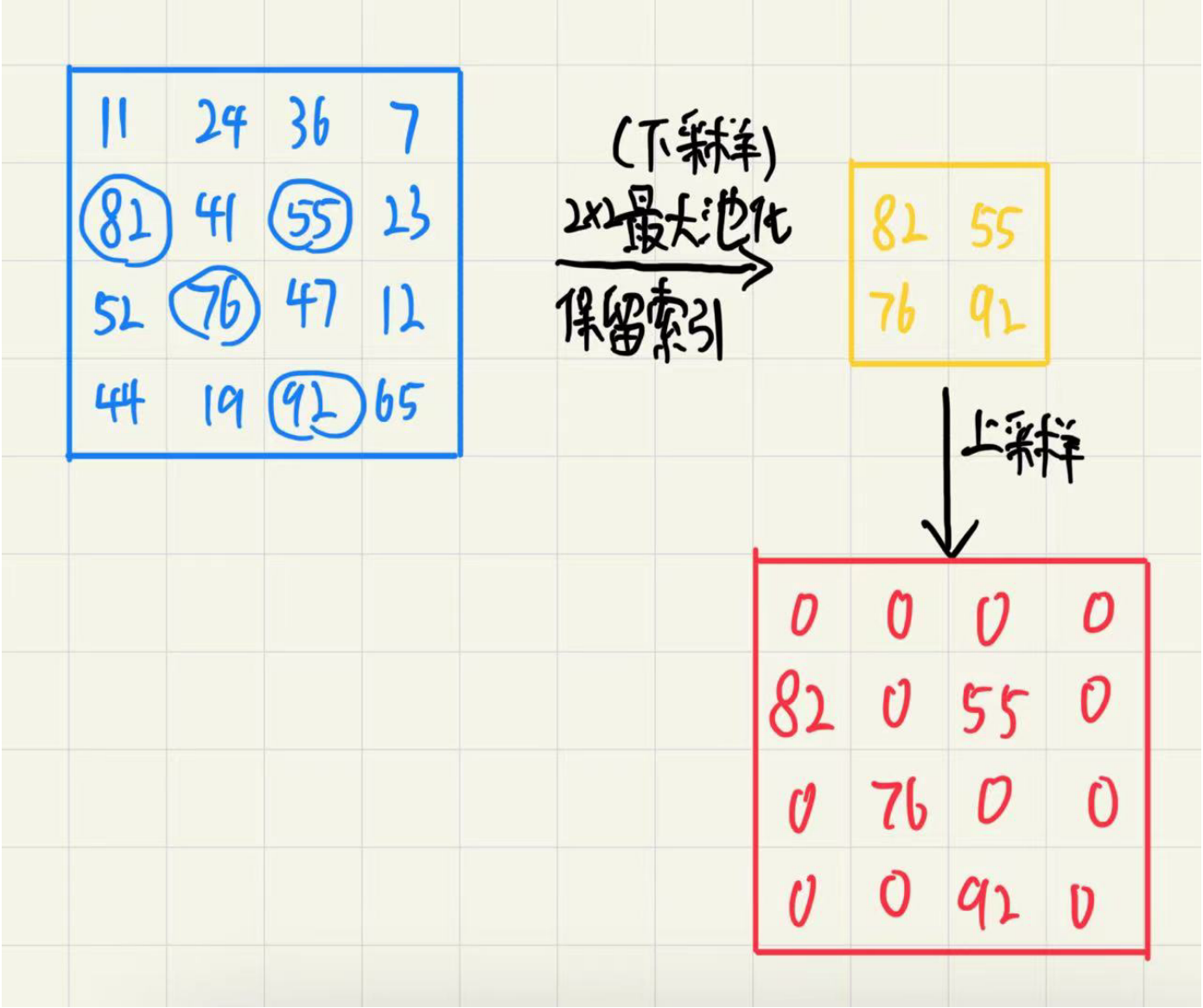
Fig. 2. An illustration of the SegNet architecture. There are no fully connected layers and hence it is only convolutional. A decoder upsamples its input using the transferred pool indices from its encoder to produce a sparse feature map(s). It then performs convolution with a trainable filter bank to densify the feature map. The final decoder output feature maps are fed to a soft-max classifier for pixel-wise classification.

主要贡献

- 使用了更多的短路连接
- 使用了最大池化层索引保留技术

详解

最大化池化层索引保留技术，其实就是在下采样时保留每个pool里最大数的索引位置，在上采样时再把他们放回去记录的索引位置上。



但是这个方法会消耗内存记录索引位置，所以看看就好。

Dilated Convolutions

> 原文：“Multi-Scale Context Aggregation by Dilated Convolutions”(2015)

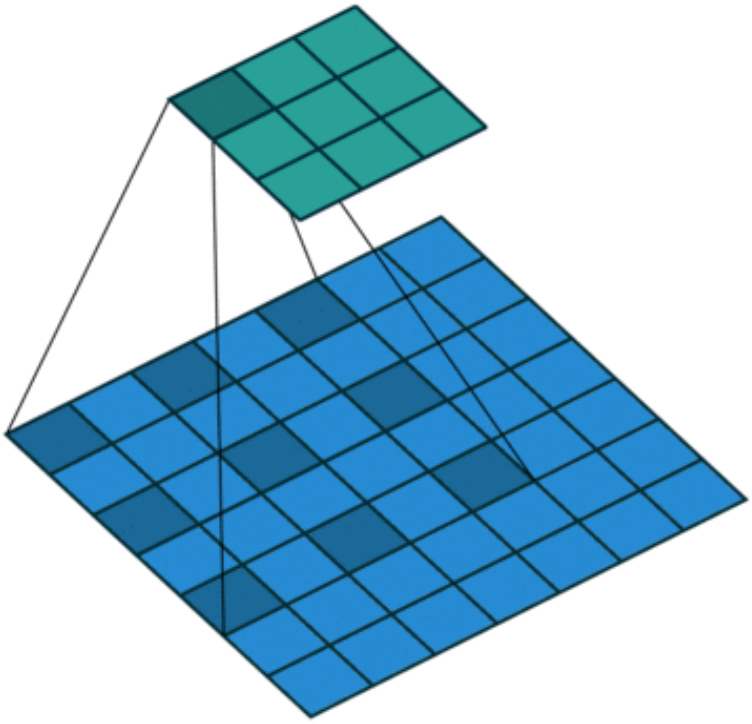
主要贡献

- 提出了神器空洞卷积
- 提出使用扩张卷积进行多尺度聚合的“上下文模块（context module）”

详解

1）空洞卷积。我觉得没有比这讲得更好了的https://www.zhihu.com/question/54149221/answer/1838484189。值得注意的是，在后续实验中，人们发现不同空洞卷积的膨胀率应该遵循一定规律，这样可以利用好每一个像素，并且节约计算资源。详情可看这篇https://zhuanlan.zhihu.com/p/453558537。

总结下，空洞卷积可以在提升感受野的同时灵活多尺度，并且计算量不会有很高提升。



2）上下文模块。上下文模块被单独训练，其输入是前端模块的输出。该模块是不同扩张率的扩张卷积级联，以便聚合多尺度上下文信息，并改善前端模块的预测结果。

剩下的DeepLab系列，RefineNet，PSPNet和Large Kernel分两次上传吧。哭，好多要看的。

DeepLab (v1)（感觉没看懂）

> V1原文:"Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs"

主要贡献

- 使用了空洞卷积

- 使用了全连接随机生成场（Fully connected CRF）来对分割结果进行全局一致性调整

详解

1）DeepLab使用了空洞卷积，不增加参数量就可以增大感受野

2）CRF。详情可以看这篇文章https://zhuanlan.zhihu.com/p/104562658。通俗来理解，类似NLP任务中通过隔壁的词预测词性，鼓励网络学习相邻像素的关系和标签。

DeepLab（v2）（感觉没看懂）

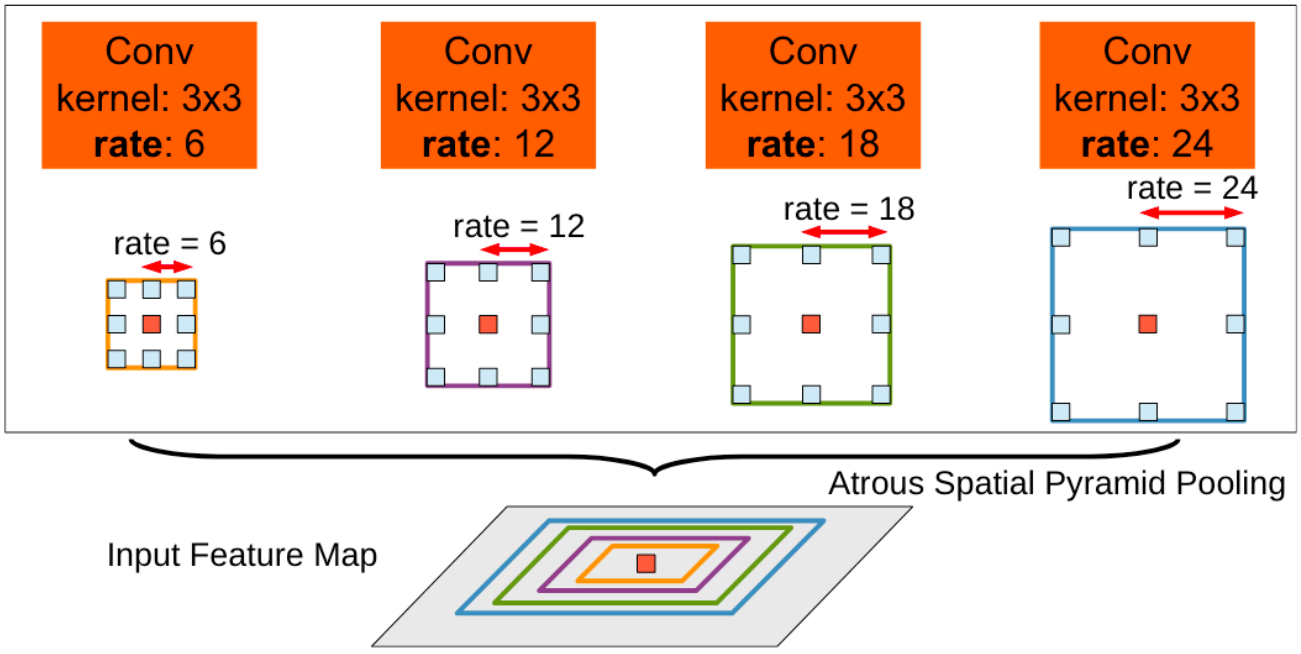
> V2原文:"DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs"

主要贡献

- 提出了空洞空间金字塔池化Atrous spatial pyramid pooling (ASPP)
- 将VGG替换为ResNet
- 使用了多尺度融合

详情

直接引用这个博客里的吧https://zhuanlan.zhihu.com/p/75333140，“其中ASPP的引入是最大也是最重要的改变。多尺度主要是为了解决目标在图像中表现为不同大小时仍能够有很好的分割结果，比如同样的物体，在近处拍摄时物体显得大，远处拍摄时显得小。具体做法是并行的采用多个采样率的空洞卷积提取特征，再将特征融合，类似于空间金字塔结构，形象的称为Atrous Spatial Pyramid Pooling (ASPP)。具体形式如下图所示”



RefineNet

主要贡献

- 提出了多路径网络，利用多级别的抽象用于高分辨率语义分割

- 大量采用ResNet进行连接，使得梯度能在长距离和短距离上进行传播
- 实现了端到端训练

详情

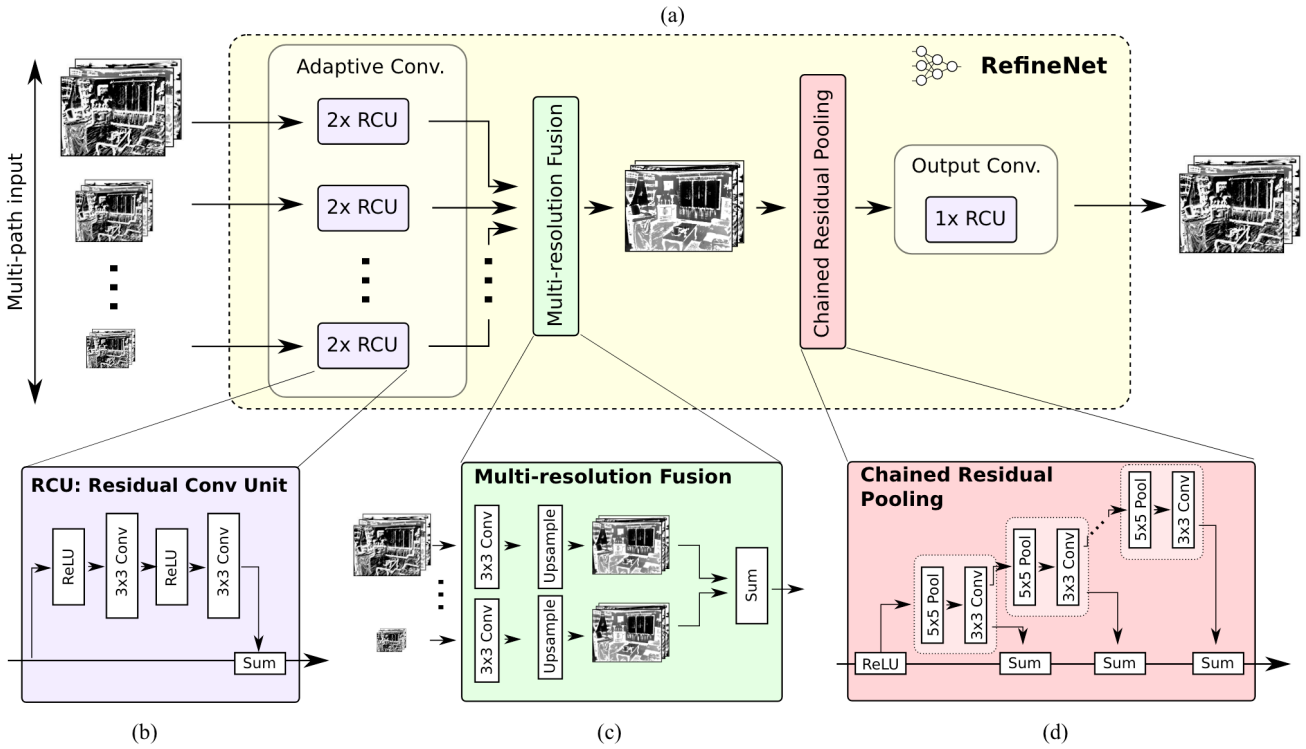
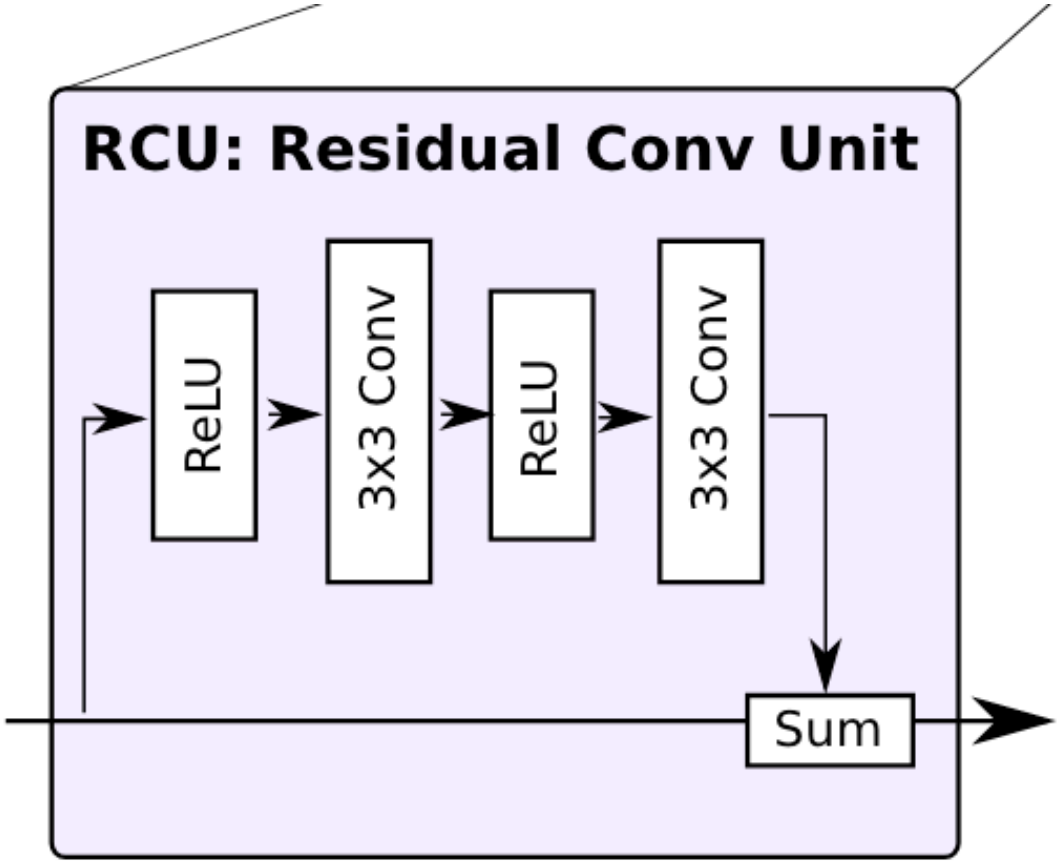


Figure 3. The individual components of our multi-path refinement network architecture RefineNet. Components in RefineNet employ residual connections with identity mappings. In this way, gradients can be directly propagated within RefineNet via local residual connections, and also directly propagate to the input paths via long-range residual connections, and thus we achieve effective end-to-end training of the whole system.

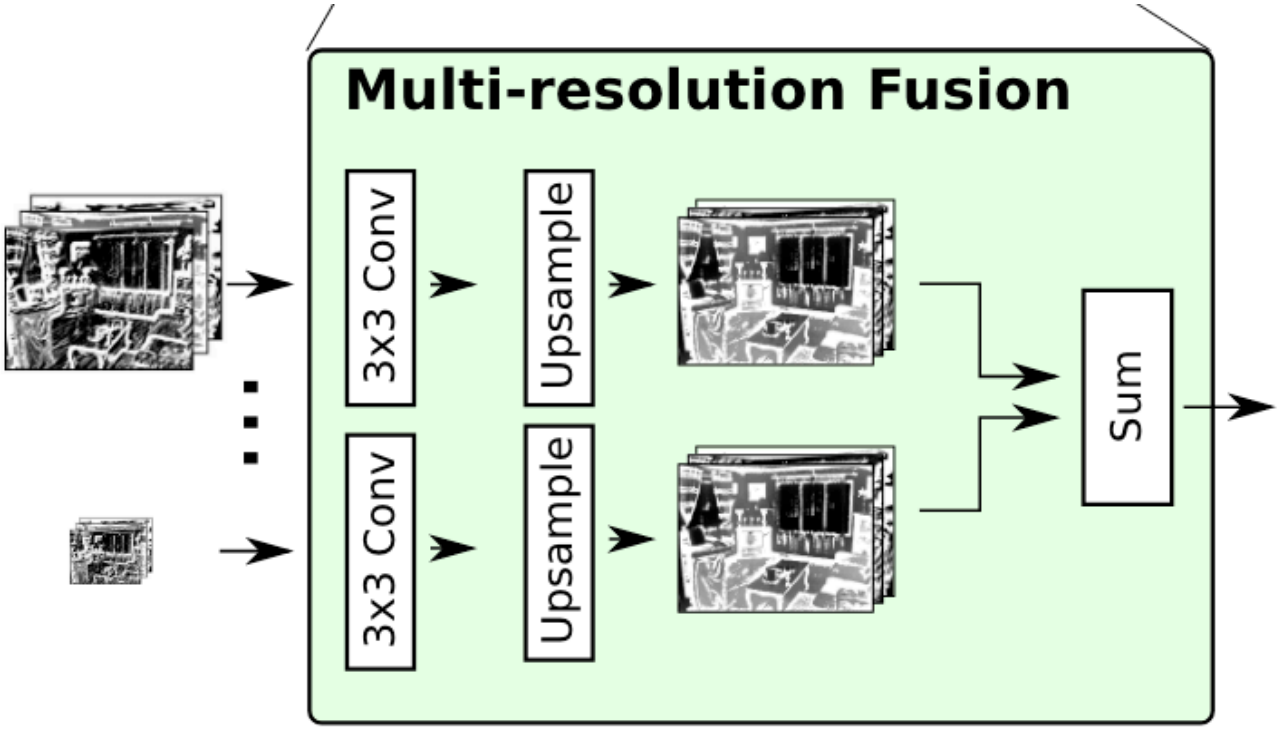
1）ResNet模块



(b)

将输入和经过处理后的特征图进行融合，和ResNet的F（x）=x+R（x）相同。

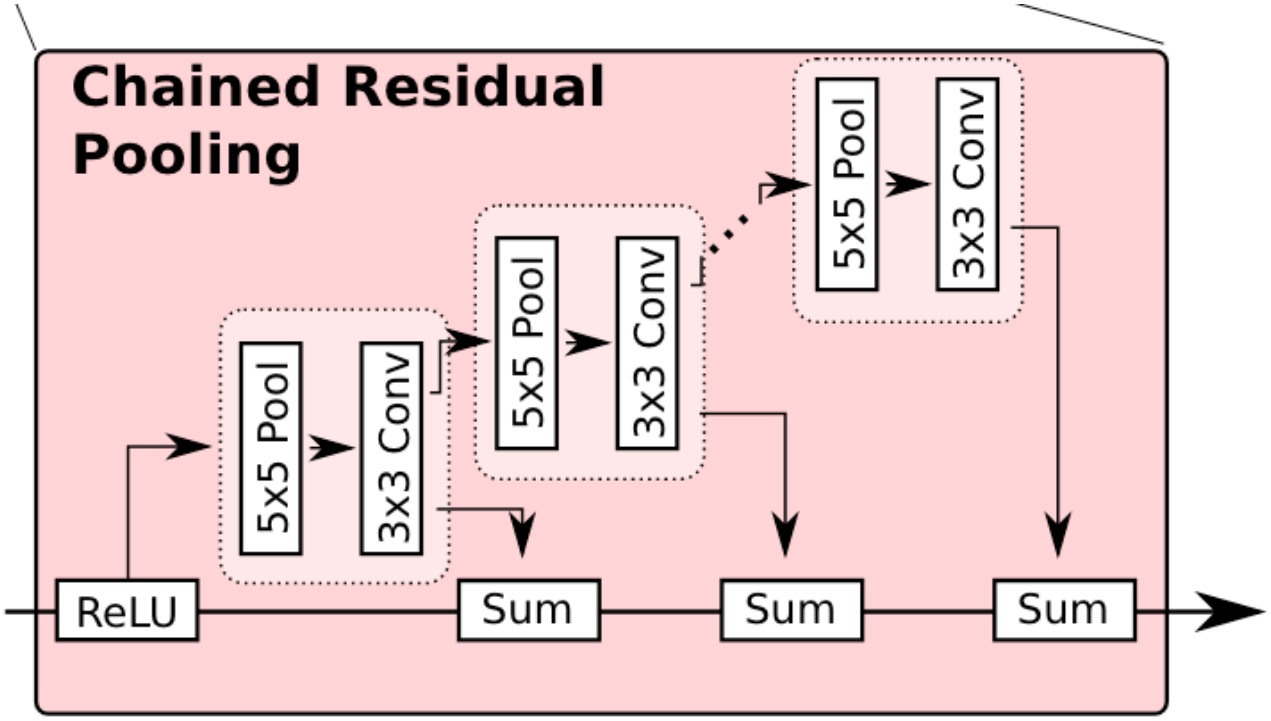
2）多尺度融合模块



(c)

输入是不同尺寸，卷积时特征图尺寸不变，但是在上采样后尺寸相同，这样就可以得到不同分辨率的语义信息，最后相加得到特征。

3）链式残差池化模块



(d)

通过分批次的多个窗口池化操作，学习不同等级的特征，并且逐步融合。