

RefineNet

论文信息

论文地址：[RefineNet: Multi-Path Refinement Networks with Identity Mappings for High-Resolution Semantic Segmentation](#)

发表时间：20 Nov 2016

创新点

1. 本文提出一种叫做RefineNet的网络模块，它是基于Resnet的残差连接的思想设计的，可以充分利用下采样过程损失的信息，使稠密预测更为精准。
2. 提出chained residual pooling，能够以一种有效的方式来捕捉背景上下文信息。

思想

目前流行的深度网络，比如VGG，Resnet等，由于pooling和卷积步长的存在，feature map会越来越小，导致损失一些细粒度的信息（低层feature map有较丰富的细粒度信息，高层feature map则拥有更抽象，粗粒度的信息）。对于分类问题而言，只需要深层的强语义信息就能表现较好，但是对于稠密预测问题，比如逐像素的图像分割问题，除了需要强语义信息之外，还需要高空间分辨率。

针对这些问题，很多方法都提出了解决方案：

1. 针对pooling下采样过程中的分辨率损失，采用deconvolution恢复。但是却很难恢复位置信息。
2. 使用空洞卷积保持分辨率，增大感受野，但是这么做有两个缺点：A.明显增加了计算代价。B.空洞卷积是一种coarse sub-sampling，因此容易损失重要信息。
3. 通过skip connection来产生高分辨率的预测。

作者认为高级语义特征可以更好地进行分类识别，而低级别视觉特征有助于生成清晰、详细的边界。所以作者认为第3点是很好的思路。基于此，作者提出了RefineNet，其主要贡献为：

1. 提出一种多路径refinement网络，称为RefineNet。这种网络可以使用各个层级的features，使得语义分割更为精准。
2. RefineNet中所有部分都利用residual connections（identity mappings），使得梯度更容易短向或者长向前传，使段端对端的训练变得更加容易和高效。
3. 提出了一种叫做chained residual pooling的模块，它可以从一个大的图像区域捕捉背景上下文信息。

模型

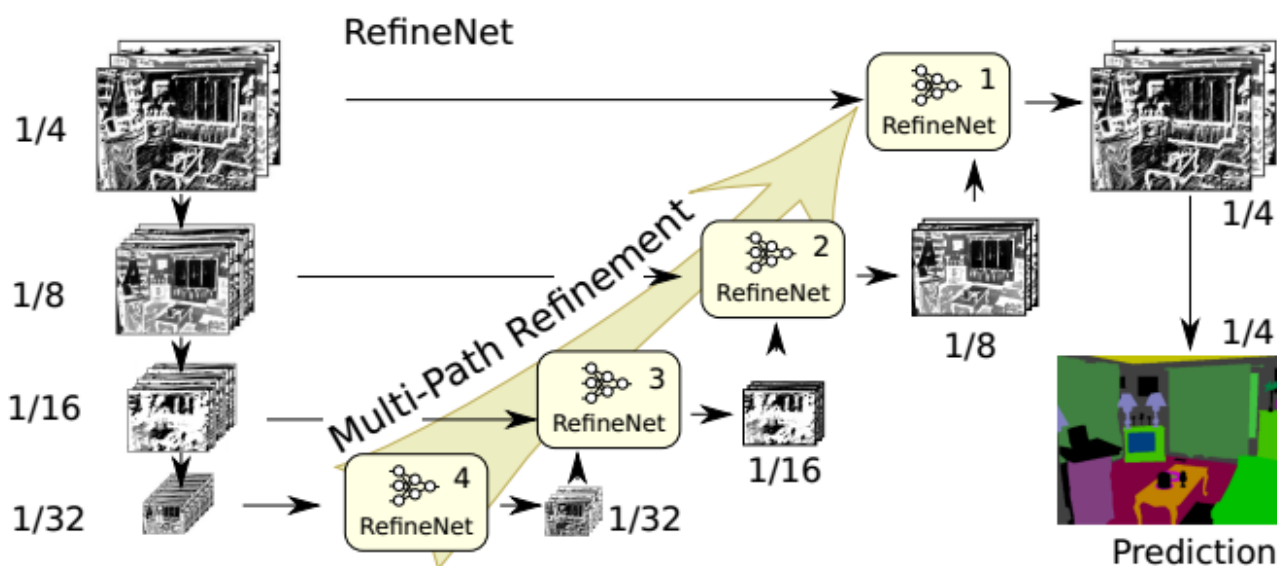
Multi-path refinement

根据feature map的分辨率将预训练RESNET划分为4个block，并采用4个RefineNet的级联结构，每个RefineNet都接收一个相应的RESNET block的输出和之前的RefineNet。

注意：

1. 这样的设计不是唯一的。实际上每个RefineNet可以接收多个RESNET blocks。不过这里只将前者。

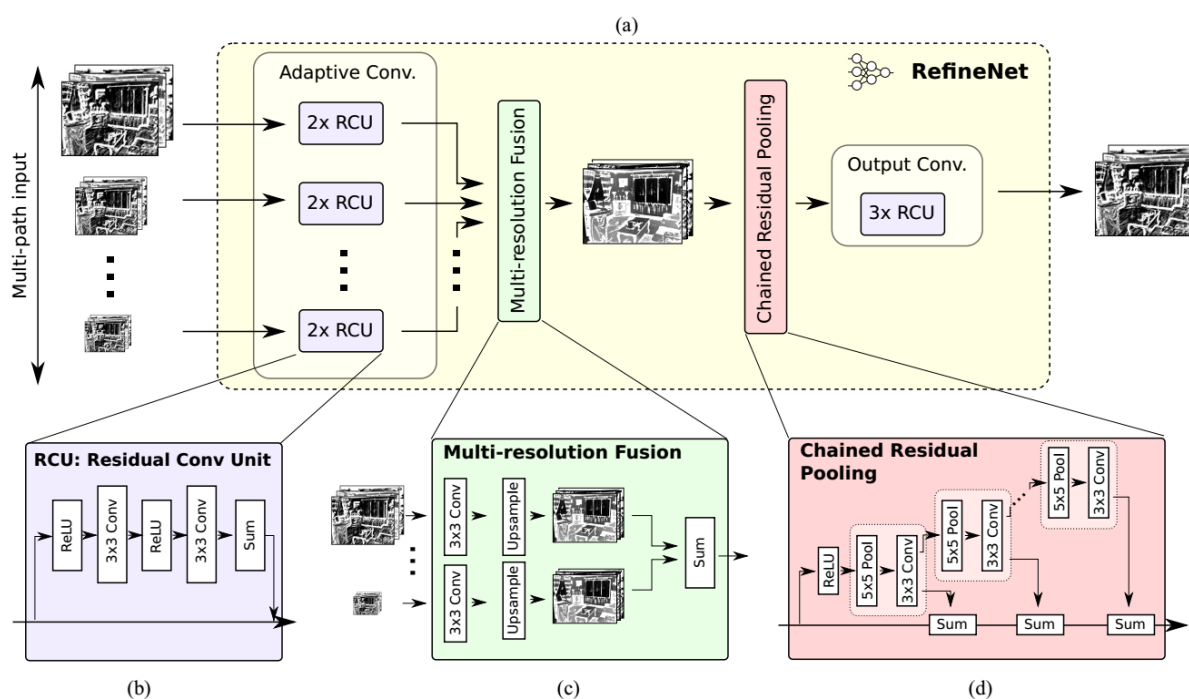
2. 虽然所有的RefineNet都具有相同的内部结构，但是它们的参数没有不一样，这样允许更灵活地适应各个级别的细节信息。



为了解决深度网络缺少细粒度信息的这个限制。

多路径refinement

RefineNet



RefineNet包括以下几种小模块：

1. Residual convolution unit：对ResNet block进行2层的卷积操作。注意这里有多多个ResNet block作为输入。
2. Multi-resolution fusion：将1中得到的feature map进行加和融合。
3. Chained residual pooling：该模块用于从一个大图像区域中捕捉背景上下文。注意：pooling的stride为1。
4. Output convolutions：由三个RCUs构成。

实验

本文在很多数据集上做了实验，效果比较好，这里仅看PASCAL VOC2012的结果：



Figure 5. Our prediction examples on VOC 2012 dataset.

与其他方法的对比：

Table 5. Results on the PASCAL VOC 2012 test set (IoU scores). Our RefineNet archives the best performance (IoU 83.4).

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	potted	sheep	sofa	train	tv	mean
FCN-8s [36]	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
DeconvNet [38]	89.9	39.3	79.7	63.9	68.2	87.4	81.2	86.1	28.5	77.0	62.0	79.0	80.3	83.6	80.2	58.8	83.4	54.3	80.7	65.0	72.5
CRF-RNN [47]	90.4	55.3	88.7	68.4	69.8	88.3	82.4	85.1	32.6	78.5	64.4	79.6	81.9	86.4	81.8	58.6	82.4	53.5	77.4	70.1	74.7
BoxSup [10]	89.8	38.0	89.2	68.9	68.0	89.6	83.0	87.7	34.4	83.6	67.1	81.5	83.7	85.2	83.5	58.6	84.9	55.8	81.2	70.7	75.2
DPN [35]	89.0	61.6	87.7	66.8	74.7	91.2	84.3	87.6	36.5	86.3	66.1	84.4	87.8	85.6	85.4	63.6	87.3	61.3	79.4	66.4	77.5
Context [30]	94.1	40.7	84.1	67.8	75.9	93.4	84.3	88.4	42.5	86.4	64.7	85.4	89.0	85.8	86.0	67.5	90.2	63.8	80.9	73.0	78.0
DeepLab [5]	89.1	38.3	88.1	63.3	69.7	87.1	83.1	85.0	29.3	76.5	56.5	79.8	77.9	85.8	82.4	57.4	84.3	54.9	80.5	64.1	72.7
DeepLab2-Res101 [6]	92.6	60.4	91.6	63.4	76.3	95.0	88.4	92.6	32.7	88.5	67.6	89.6	92.1	87.0	87.4	63.3	88.3	60.0	86.8	74.5	79.7
CSupelec-Res101 [4]	92.9	61.2	91.0	66.3	77.7	95.3	88.9	92.4	33.8	88.4	69.1	89.8	92.9	87.7	87.5	62.6	89.9	59.2	87.1	74.2	80.2
RefineNet-Res101	94.9	60.2	92.8	77.5	81.5	95.0	87.4	93.3	39.6	89.3	73.0	92.7	92.4	85.4	88.3	69.7	92.2	65.3	84.2	78.7	82.4
RefineNet-Res152	94.7	64.3	94.9	74.9	82.9	95.1	88.5	94.7	45.5	91.4	76.3	90.6	91.8	88.1	88.0	69.9	92.3	65.9	88.7	76.8	83.4