周报（2023. 3.14-2023.3.21）姓名：孙瑞阳

每日小结

	周一	周二	周三	周四	周五
早	MobileVIT 代码，上课	CCNet 代码，MobileVIT	大数据集处理，上课	上课	Non-Local 代码
中	CCNet 代码，上课	论文阅读	论文阅读	跑代码，上课	Non-Local 代码
晚	上课，CCNet	组会	大数据集处理	大数据集处理	跑 GPT

注：简单表述当前时间段工作，如看文献 1，整理数据等

科研详情

文献阅读

文献1

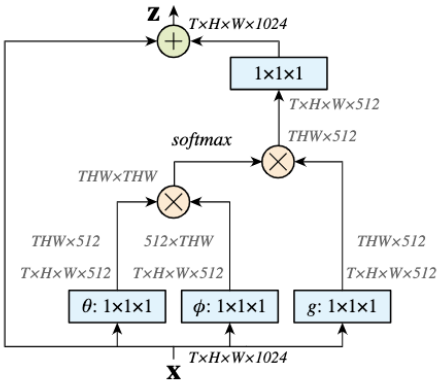
题目： Non-Local neural networks

作者： **Xiaolong Wang, Ross Girshick, Abhinav Gupta, Kaiming He**

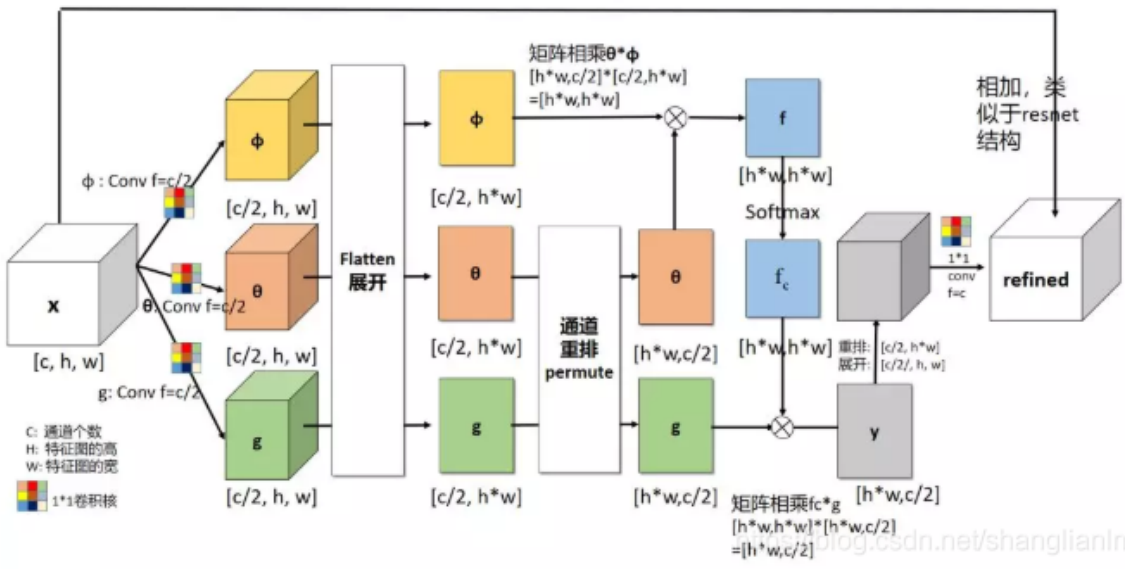
出处： arxiv 2018

方法：

Non-Local Neural Network 和 Non-Local Means 非局部均值去噪滤波有点相似。普通的滤波都是 3×3 的卷积核，然后在整个图片上进行移动，处理的是 3×3 局部的信息。Non-Local Means 操作则是结合了一个比较大的搜索范围，并进行加权。



non-local operations 通过计算任意两个位置之间的交互直接捕捉远程依赖，而不用局限于相邻点，其相当于构造了一个和特征图谱尺寸一样大的卷积核，从而可以维持更多信息。

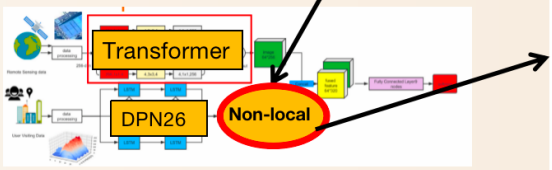


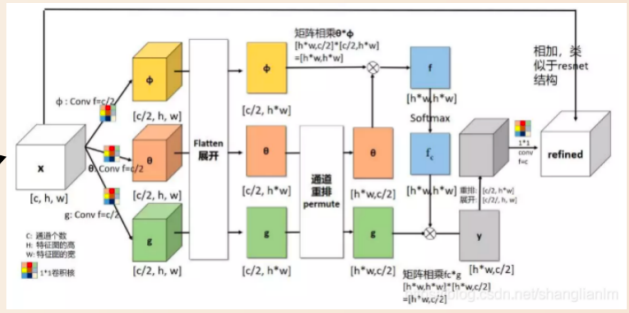
此外，non-local 操作也可以用于含时序的任务中，如视频分类任务，可综合几帧的特征来增强当前帧的特征。消融实验验证了同时在时域和空域上加入 non-local 操作效果会最好

启发：

1. 在自己的网络 DPN 后接 nonlocal，效果一般，0.682，没有 CBAM 的通道效果好（0.695）

- DPN后增加Non-local注意力





tansformer+DPN+CBAM通道部分	0.695	0.607
tansformer+DPN	0.694	0.606
tansformer+DPN+Non-local	0.682	0.590
tansformer+DPN+三个Non-local	0.630	0.516

2. 在自己的代码复现了 2 个、3 个 nonlocal 堆积使用，效果不好，没有一个 nonlocal 的效果好（本文视频任务说多个堆积会显著提升，不符合我的实验结果）

文献2

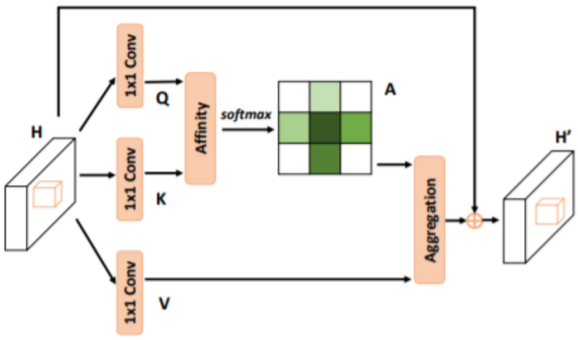
题目：CCNet: Criss-Cross Attention for Semantic Segmentation

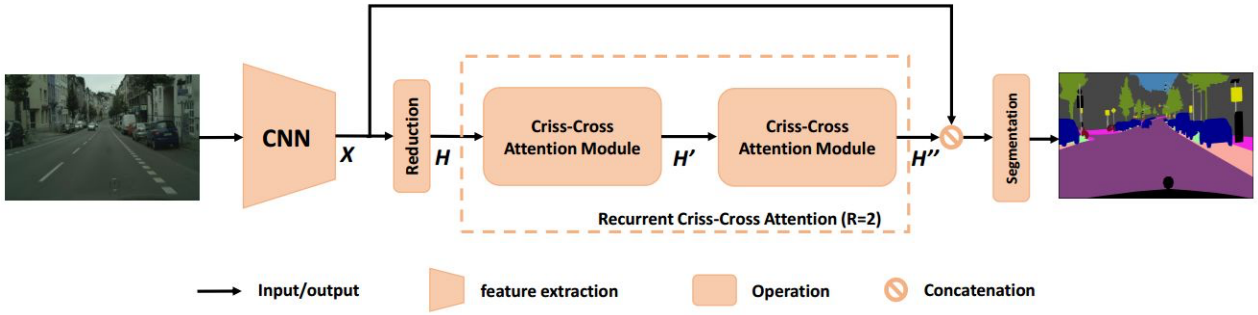
作者：Zilong Huang, Xinggang Wang et al

出处：CVPR2018

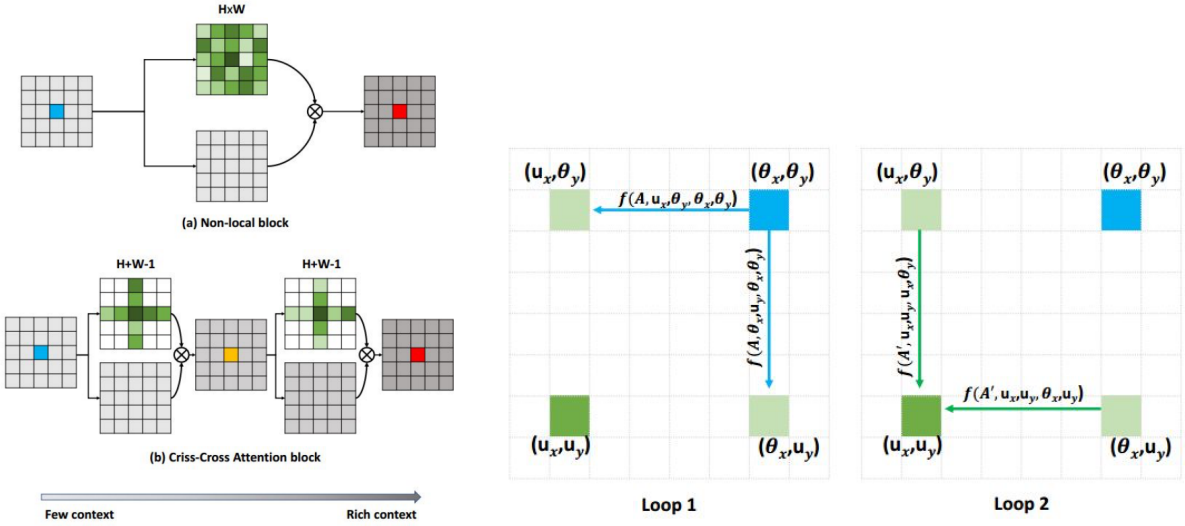
方法：

远距离像素间具有一定的相关性，将这种上下文的信息融合，可以得到更加有效的 feature map，提升模型的语义理解能力。本文提出 CCNet，利用 criss-cross attention module 提取图像的周围像素的上下文信息，从而捕获全局信息。





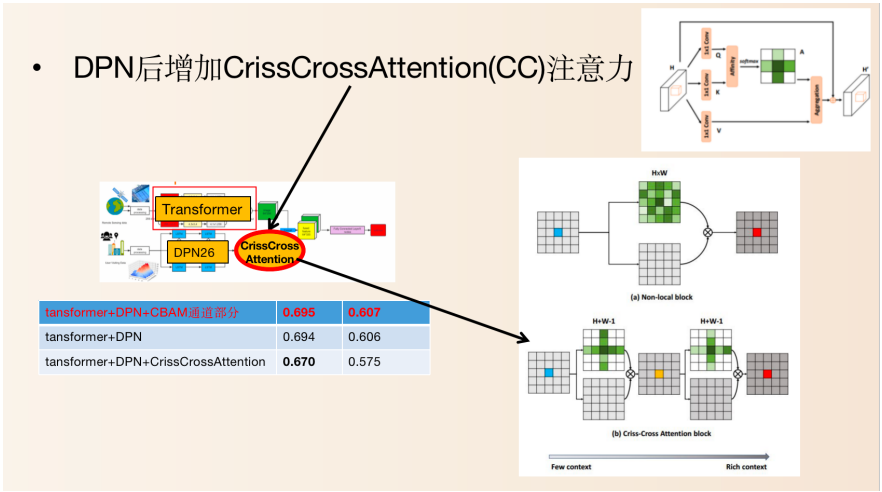
由于卷积操作只针对感受野内的像素进行操作，具有一定的 local 特性，常见的 FCN 网络很难提取到上下文语义信息和全局特征。因此 non-local 就应运而生。类似于注意力机制，针对每个像素点生成不同的权重值，对 feature map 进行加权处理。但是这样操作，计算量大大增加。于是本文就提出了一种十字交叉的注意力机制。如下图左所示，（a）即为 non-local 模块，（b）为 criss-cross 注意力模块。其中有绿色的图就表示提取出的注意力机制权重。



在 criss-cross attention module 中，重复使用了两次 criss-cross 注意力机制，因为只使用一次，该像素点的只能与周围呈十字型的像素点进行信息交互，使用两次之后，较远处的像素点同样可以间接作用于该像素点。信息传播大致如上图右所示。相比与 non-local,计算量大大减少。

启发：

在自己的网络 DPN 后接 **CCNet: Criss-Cross Attention**，效果一般，0.670，没有 CBAM 的通道效果好（0.695），并且 non-local > CrissCrossAttention(CC)：

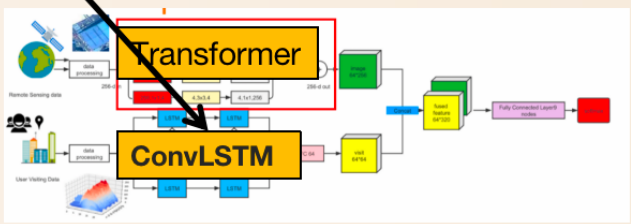


工作进展

1：阅读文献；

2：URFC 小数据集**实验结果：**

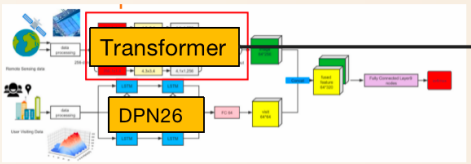
DPN换成了ConvLSTM，效果很差，只有0.46



tansformer+DPN	**0.694**	0.606
transformer+ConvTransformer（1头注意力）	0.658	0.570
transformer+ConvTransformer（4头注意力）	0.649	0.557
tansformer+DPN+Transformer	**0.670**	0.577
tansformer+ConvLSTM	**0.46**	0.326

遥感 VIT 做成了多尺度的，效果略有下降

多尺度融合VIT transformer，效果稍微下降



无多尺度VIT，16*16patch：

Accuracy	F1 score
0.695	0.606

猜测原因：

- vit_b_16和vit_b_32的参数数量也不同。vit_b_16有8.6亿个参数，而vit_b_32有3.4亿个参数[2]。

思路一：
vit_b_16预训练模型，patch规定16*16
vit_b_32预训练模型，patch规定32*32
目前直接把这两个预训练模型微调后的特征**concat**

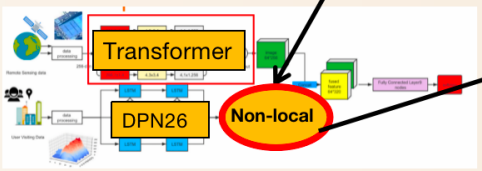
Accuracy	F1 score
0.691	0.598

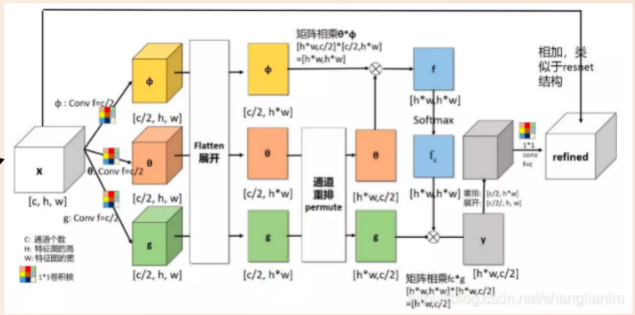
思路二：
vit_b_16预训练模型，patch规定16*16
用16*16的模型，把图像切分为32*32的块，
然后再下采样成16*16，输入模型
得到32的尺度，再跟原本就16*16的块融合

Accuracy	F1 score
0.669	0.572

DPN 后分别增加了文献 1、2 的 non-local、MobileVITAttention、CrissCrossAttention(CC)3 种注意力，效果都不如 CBAM 的通道注意力好，效果：MobileVITAttention > non-local > CrissCrossAttention(CC)：

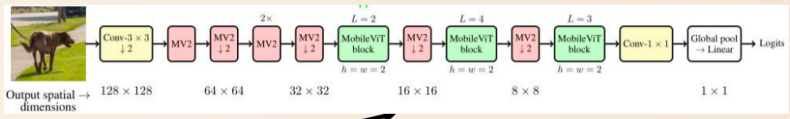
- DPN后增加Non-local注意力，效果0.682

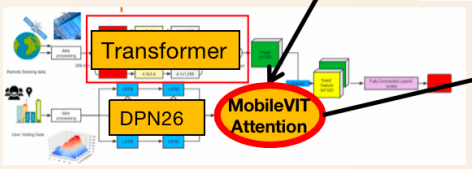




tansformer+DPN+CBAM通道部分	0.695	0.607
tansformer+DPN	0.694	0.606
tansformer+DPN+Non-local	**0.682**	0.590
tansformer+DPN+三个Non-local	0.630	0.516

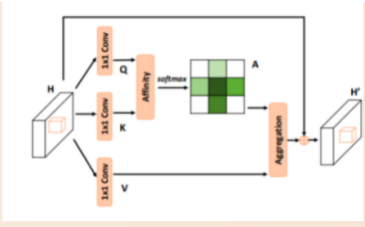
- DPN后增加MobileVITAttention注意力，效果0.685

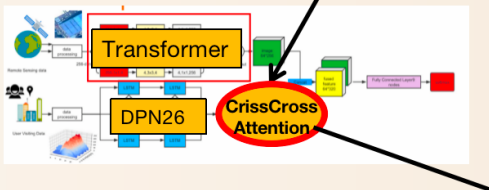


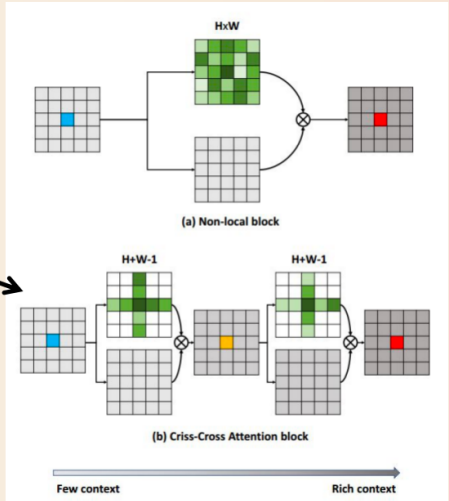


tansformer+DPN+CBAM通道部分	0.695	0.607
tansformer+DPN	0.694	0.606
tansformer+DPN+MobileVITAttention	**0.685**	0.589

- DPN后增加CrissCrossAttention(CC)注意力效果0.670

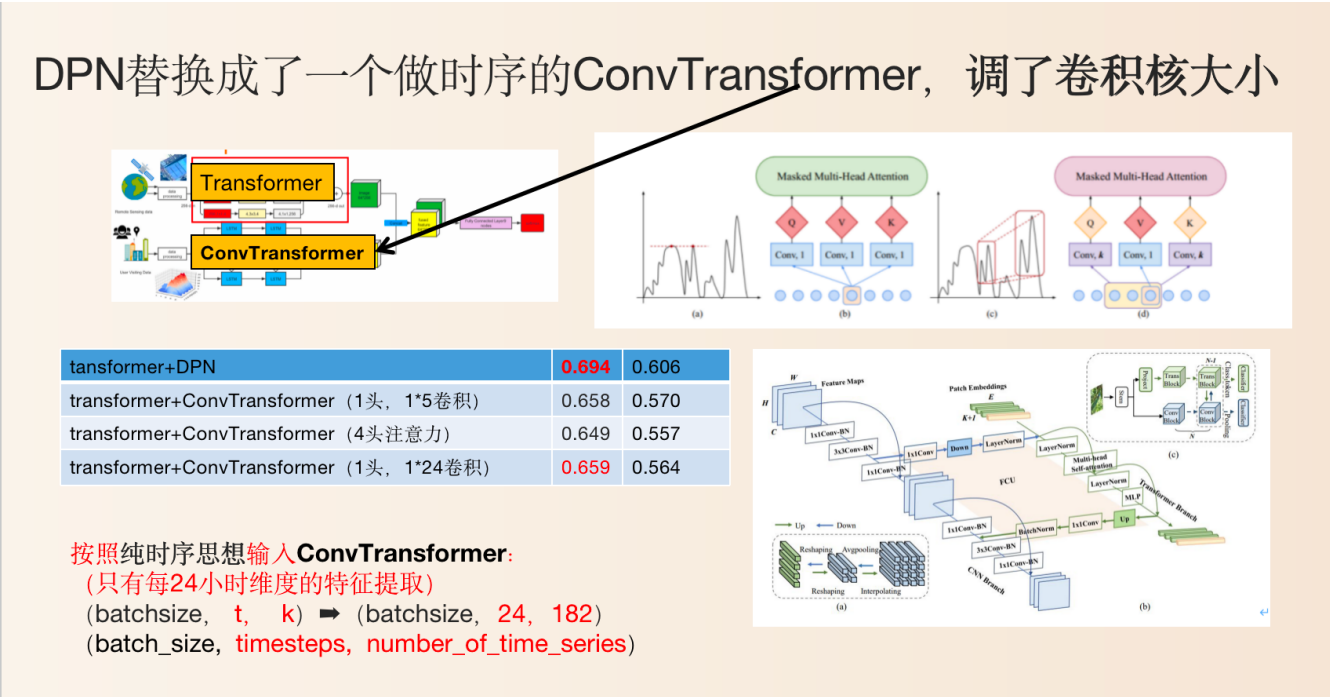






tansformer+DPN+CBAM通道部分	0.695	0.607
tansformer+DPN	0.694	0.606
tansformer+DPN+CrissCrossAttention	**0.670**	0.575

ConvTransfomer 卷积核调了，1*24 效果最好（符合 24 小时的时序特点）



tansformer+DPN	0.694	0.606
transformer+ConvTransformer（1头，1*5卷积）	0.658	0.570
transformer+ConvTransformer（4头注意力）	0.649	0.557
transformer+ConvTransformer（1头，1*24卷积）	0.659	0.564

按照纯时序思想输入**ConvTransformer**：
（只有每24小时维度的特征提取）
（batchsize，t，k）➡（batchsize，24，182）
（batch_size，timesteps，number_of_time_series）

3：URFC 大数据集处理完了，跑的很慢，在跑

下周计划

1：社交网络：卷积+循环网络的形式也试了好几种了，目前看来还是 DPN 最好，看能不能尝试复杂的卷积（ResNet）+Transformer

2：跑完URFC大数据集的实验，说不定时序convTransfomer在大数据集上比DPN更好

3：修改loss，目前还在学习