(GCN) Large Kernel Matter, Improve Semantic Segmentation by Global Convolutional Network (2017)

C Peng et al.
Resume

March 7, 2019

---

1 Introduction

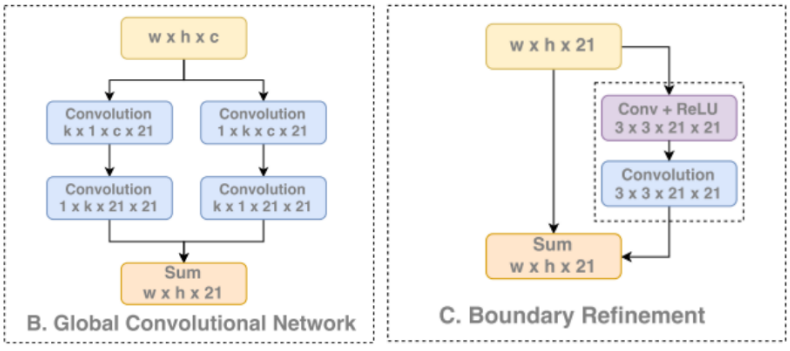
The authors propos a global convolutionnal network to adress the problem of localization and classification in semantic segmentation, GCN uses two asymetric convolutions, for k x k large kernels, they use two sucessive 1 x k and k x 1 kernel with less parameters and even better performances due the lack of overfitting with larget networks. and they also suggest residual based boundary refinement to further refine the object boundaries.

2 Design choices

The authors follow two design principles:

1) from the localization view the model structure should be FC to retain the localization perfomance and no fully connected or global pooling layers should be used given that they discard the localization information.

2) from the classification view, larger kernel size sould be adopted in the network arhitecure to enable dense connections between the feature maps and per pixel classifiers, and to make GC practical, they adopt symetrical, separable large filters to reduce the model parameters and computation cost, with a boundary refinnement block.



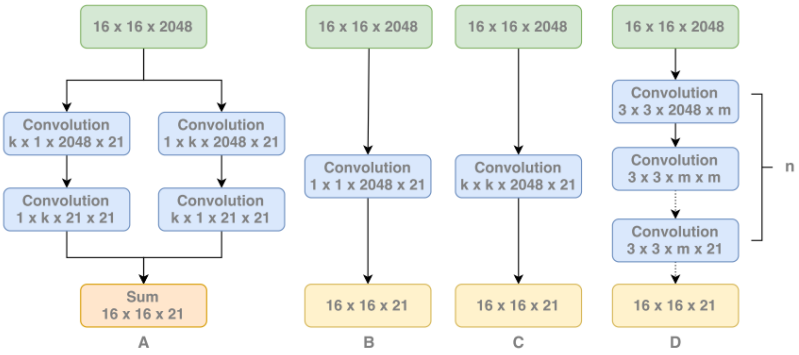
2.1 Is GCN better?

To check if the GCN is really better, they compare their approach with three other ones,

- the baseline with only a simple 1x1 convolution,

- large kernel with different size k x k,
- a sucession of n (three 3x3 convolutions), with different number of n?

These three are compared to the symetric 1xk and kx1 convolutions proposed by the authors.



First they compare the GCN to the baseline, and we seek the larger k, they better the results.

k	base	3	5	7	9	11	13	15
Score	69.0	70.1	71.1	72.8	73.4	73.7	74.0	74.5

This might be due to the number of parameters, given the models better representation capabilites, so to compare the GCN taking into account the number of parameters, they compare it to kxk larger kernels, with even higher number of parameters, and GCN still gives better results, maybe to the overfitting of large kernels.

k	3	5	7	9
Score (GCN)	70.1	71.1	72.8	73.4
Score (Conv)	69.8	70.4	69.6	68.8
# of Params (GCN)	260K	434K	608K	782K
# of Params (Conv)	387K	1075K	2107K	3484K

Lastly they compare the GCN to the wiedly used approach of staking multiple 3x3 convs to obtain similar field of view

k	3	5	7	9	11
Score (GCN)	70.1	71.1	72.8	73.4	73.7
Score (Stack)	69.8	71.8	71.3	69.5	67.5

2.2 The effect of refinnement block

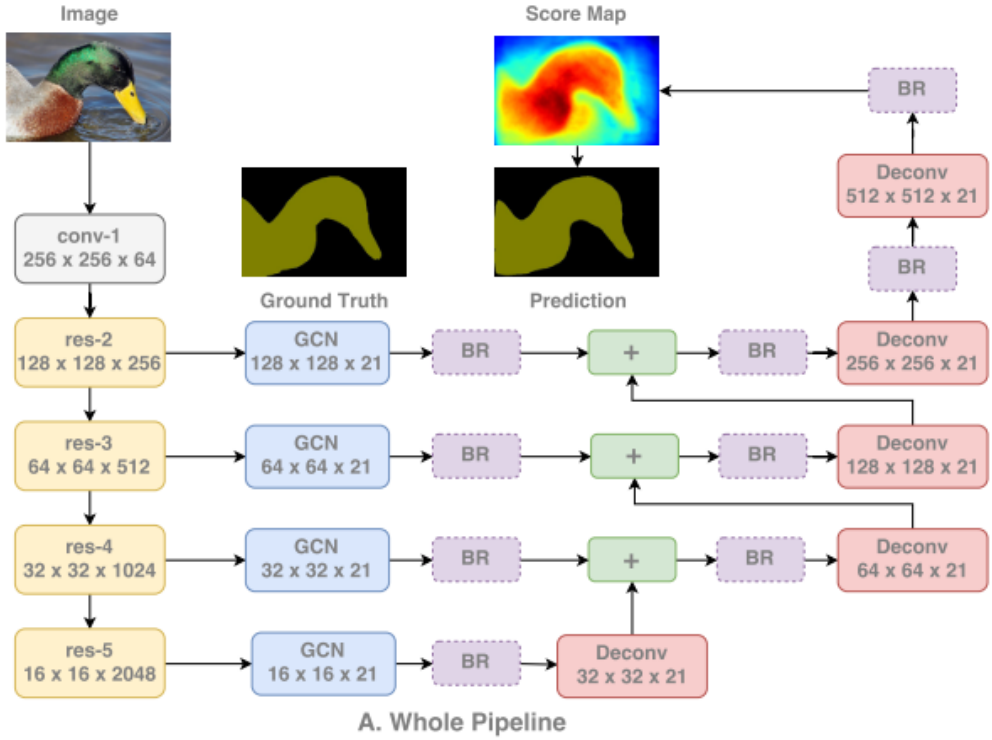
To verify the effect of the refinnement block, the authors divide the score map into two parts: a) bou,dary region whose pixels locate close to objects boundary (distance ¡ 7), and b) the internal region as other pixels, and they evaluate the segmentation with or without the regfinnemnet block, and with different values of k.

Model	Boundary (acc.)	Internal (acc.)	Overall (IoU)
Baseline	71.3	93.9	69.0
GCN	71.5	95.0	74.5
GCN + BR	73.4	95.1	74.7

And the results shows that the usage of the refinnement block increase the boundary accuracy and GCn the internal accuracy.

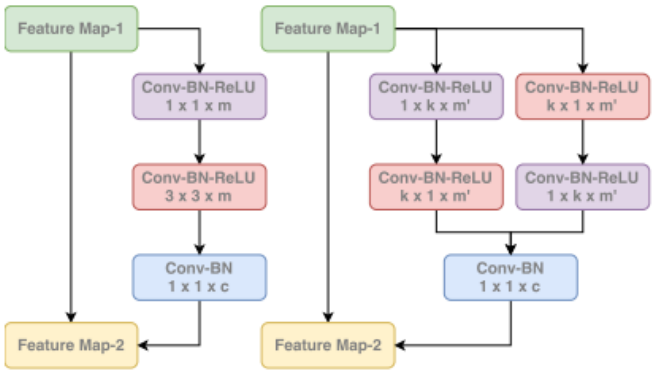
3 Model & Training

For the model, they use a resnet backbone, and they extract the feature at different stages with different spatial dimensions, and passed them through the GCN and BR, and smaller one are upsampled and added to the larger ones, until we get the score map with the same size as the input.



3.1 ResnetGCN

The authors propose to modify the resnet architecture and to use 1xk followed by kx1 in the bottle neck of the last two convolutinnal blocks of resnet, they retrain the model on imagenet, and even if the performances on imagenet are worse than the regular resnet, when used in segmentation, the preformances are better.



Component	Output Size	ResNet50	ResNet50-GCN
conv1	112×112	7×7, 64, stride 2	
res-2	56×56	3×3 max pool, stride 2	
		$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
res-3	28×28	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
res-4	14×14	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} (1 \times 5, 85) & (5 \times 1, 85) \\ (5 \times 1, 85) & (1 \times 5, 85) \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
res-5	7×7	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} (1 \times 7, 128) & (7 \times 1, 128) \\ (7 \times 1, 128) & (1 \times 7, 128) \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
ImageNet Classifier	1×1	global average pool, 1000-d fc, softmax	
MFlops (Conv)		3700	3700

Resutls of comparaison:

Pretrained Model	ResNet50	ResNet50-GCN
ImageNet cls err (%)	7.7	7.9
Seg. Score (Baseline)	65.7	71.2
Seg. Score (GCN + BR)	72.3	**72.5**

3.2 Training

They use regular SGD, with a batch of 1, and momentum of 0.99 and decay of 0.0005. the data augmentation used is mean substraction and H fliping, they first train the network with a dataset containing Pascal VOC, COCO & SBD for the first section, then only PASCAL and SBD and finally only PASCAL for the last stage.

Phase	Baseline	GCN	GCN + BR
Stage-1(%)	69.6	74.1	75.0
Stage-2(%)	72.4	77.6	78.6
Stage-3(%)	74.0	78.7	80.3
Stage-3-MS(%)			80.4
Stage-3-MS-CRF(%)			**81.0**