DLLAB WEEK 5 Pre-Report : FCN and Dilated Convolutions

Hyunsoo Cha[1], Kwanghoon Sohn[2]
[1]Yonsei University. [2]Yonsei University.

1 Fully Convolutional Networks for Semantic Segmentation

1.1 Brief Summary

All layers in this paper consist only of convolution layers. This is called the fully convolutional network, FCN. Building a full convolutional network[5] that brings in inputs of any size and produces outputs of matching sizes that make effective estimation and learning is key to this paper. To do so, researchers determine the size of full convolutional networks and explain spatially applying dense preference tasks to the network. Also, it draw connections to the preceding models. Training is conducted by switching AlexNet[5], VGG[9], and GoogLeNet[10] to FCN and transferring representations of datasets learned through different data via fine-tuning for segmentation. Finally, a more accurate segmentation was obtained by combining the segmantic information of the deep and coarse layer and the application information of the shallow and fine layer.

1.2 Learn more about FCN

What is convnet (CNN) The feature map of each layer consists of $h \times w \times d$. If the input image is RGB, d is 3. h, w means spatial dimension. d means feature or channel dimension. A single node is created through the kernel in feature map. The kernel involved in creating a single node is called a receive field. One of the most important properties of CNN is that positioning information is not lost even if it passes through the layers of convolution, pooling, and activation functions in the input image. This is called translation invariant. Only the feature map area corresponding to the receive field is affected.

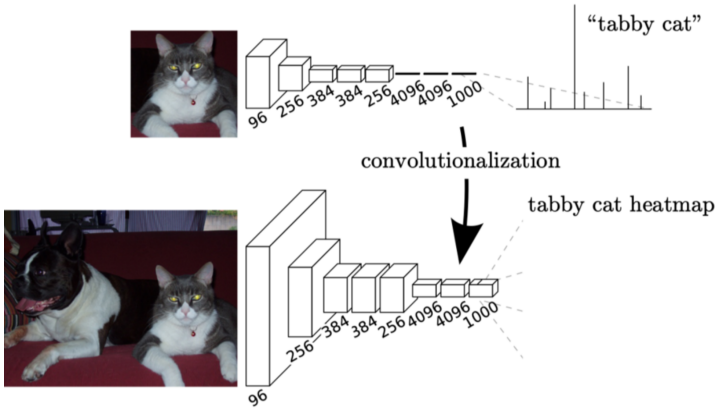

Figure 1: The fully connected layer is also made into a fully convolutional layer through convolutionalization. Adding layers and loss of space enable effective end-to-end dense learning.

Let's say the data vector $x_{i,j}$ corresponding to the (i, j) position of a particular layer. $y_{i,j}$ is a function calculated by the input. The expression is as follows.

$$y_{ij} = f_{ks}(x_{si+\delta i, sj+\delta j}, 0 \delta i, \delta jk) \tag{1}$$

k means the size of the kernel, and s is called stride, subsampling factor f. f_{ks} means the type of layer. The transformation rule is satisfied through the composition below. Applying the synthesis function means moving forward and backward in the entire network. This FCN can compute the size of any input image and produce output images of a corresponding size to the input image through the resampling process.

$$f_{ks} \circ g_{k's'} = (f \circ g)_{k'+(k-1)s', ss'} \tag{2}$$

Dense Prediction Convolutionalized model's spatial output maps are suitable for dense problems such as semantic segmentation. Reinterpretation of classification neural networks with full convolutional allows output maps to be produced for input images of all type of comparable size. Neural networks for classification create subsamples that cause filters to be small and keep the required computations reasonably. These make the output of fully convolutional versions of these neural networks rough.

Shift-and-stitch Dense prediction can be obtained from coarse output by combining the outputs obtained by shifting the input image. When f^2 input images are processed and outputs are deadlocked and combined, the prediction matches the pixels in the center of the reception field. Let input stride be s and filter weights be f_{ij}. To produce a trick, the filter needs to be sparse in the following ways.

$$f'_{ij} = \begin{cases} f_{i/s, j/s} & \text{if } s \text{ divides both } i \text{ and } j; \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Reproduce the output of the trick that repeats the filter large for each layer until all subsamples are deleted.

1.3 Architecture of FCN for Segmentation

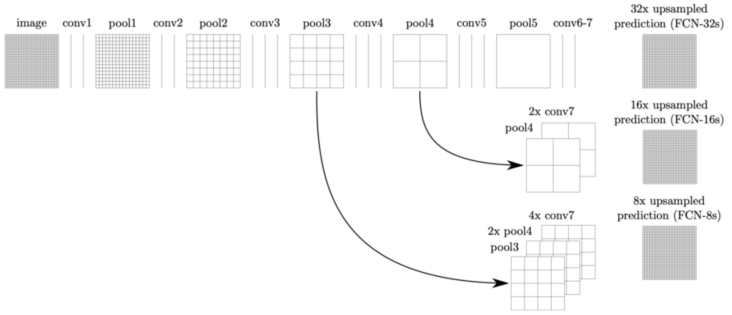

Figure 2: The top-end network is a single network that does not have a skip architecture. 32x upsamples were applied immediately. The second neural network is a 16x upsampled prediction. Upsampling twice as much before 16 times. The output image can be obtained equal to the size of the input image. For the last FCN-8s neural network, after 2x upsampling in pool3, it is combined by applying 1×1 convolutional layer. It also allows to obtain output images of the same size through 8x upsampling.

Skip Connection In this paper, researchers transform the classifier of ILSVRC into FCN. The modified method is to perform upsampling for pixel-specific loss calculation. It was followed by learning for segmentation through fine-tuning[2] and adding a skip connection between the coarse and semantic layer and the local and application layer. This structure is called a skip architecture. Calculated using a multinomic logistic loss, the expression is similar to the binary cross-entropy.

1.4 Conclusion

In the FCN paper, the classification model based on the existing deep learning was fine tuning to image processing for sematic segmentation. Through this, they perform transfer learning and compare and analyze the performance of each network. The size and shape of the input and output images must be the same to obtain per pixel loss of segmentation. To this end, there was a problem of loss of location information during upsampling. To solve this problem, they address the problem using the skip connection from fine layer to coarse layer. As a result, it gave better performance.

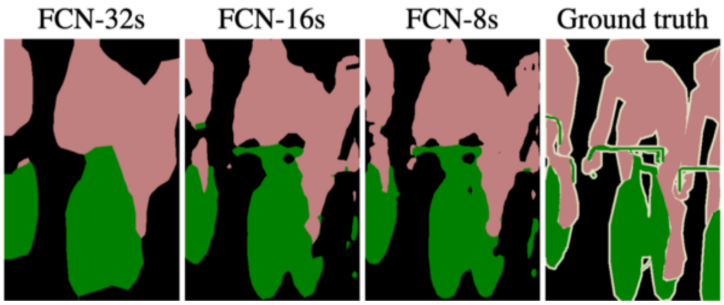

Figure 3: They can see that the performance gets better as they do Skip Connection[7]. To prevent loss of information, it has been shown that applying the skip connection from the fine layer to the coarse layer can produce better results.

2 Multi-Scale Context Aggregation By Dilated Convolutions

2.1 Brief Summary

Semantic segmentation have been developed based on convolutional networks. Image classification problems and structurally dense prediction problems are different. This is also related to semantic segmentation. To solve the dense prediction problem, there was a clear threshold for existing models, and they developed a new neural network. The key strategy is to widen the receptive region (field) with a relatively small increase in the number of parameters using extended convolutions. Although there are many different ways to obtain multi-scale context information, it has not existed before without losing its resolution. Dilated convolution helps to exponentially expand acceptance fields without loss of resolution or coverage. This module not only improves accuracy but also greatly helps simplify this network.

2.2 Dilated Convolution

Let F be a discontinuous function; let k be a discontinuous filter of size $(2r+1)^2$. Let l be the dilation factor. Then, let's redefine the operator as follows.

$$(F *_l k)(p) = \sum_{s+lt=p} F(s)k(t) \tag{4}$$

Let $*_l$ be a dilated convolution or l-dilated convolution. A familiar convolution operation is called a 1-dilated convolution. The important concept used in "algorithme a trous", convolution with a built filter, was first used in 1992[8]. In any case, the researchers wrote this paper based on the fact that there are no loss of resolution or coverage, the dilated convolution exponentially extends the receptive field. Applying the filter to the exponential increase in dilation is as follows.

$$F_{i+1} = F_i *_{2^i} k_i \ \ for\ i = 0,1,...n-2 \tag{5}$$

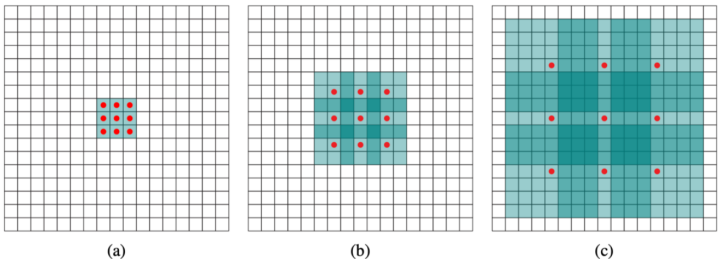

Figure 4: (a) means a 1-dilated convolution. There is no difference from the normal convolutional layer. Receptive field is 3×3. (b) 2-dilated convolutional layer. I can see that the receptive field is stretched to 7×7. (c) 4-dilated convolution layer. I can see that the receptive field has been extended to 15×15.

The receptive field increases exponentially, but the parameters increase linearly. In other words, it allows them to benefit sufficiently in terms of computation.

2.3 Context Aggregation in multi-scale

Inputs and outputs must be of the same shape, and the module is connected to the existing dense preference structure. Each layer has a C channel. Each layer's representation is the same, and even if the loss is not defined within the module and the feature map is not normalized, it gives direct dense per-class predictions.

Layer	1	2	3	4	5	6	7	8
Convolution	3×3	3×3	3×3	3×3	3×3	3×3	3×3	1×1
Dilation	1	1	2	4	8	16	1	1
Truncation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Receptive field	3×3	5×5	9×9	17×17	33×33	65×65	67×67	67×67
Output channels								
Basic	C	C	C	C	C	C	C	C
Large	$2C$	$2C$	$4C$	$8C$	$16C$	$32C$	$32C$	C

Figure 5: This table shows for context network architecture. As has been mentioned, the size increases, but passes through C feature maps without loss of resolution.

They apply 3×3 convolution with different dilation factors. Figure 6 shows that dilation is applied as 1, 1, 2, 4, 8, 16, 1. convolution is applied by following the cutting of each point by $\max(\cdot, 0)$. The last layer is performed with a $1 \times 1 \times C$ convolution. The front-end module outputs a feature map at a resolution of 64×64. The researchers eventually stopped increasing the number of receptive fields after layer 6.

Experiments[5][3] have shown that standard initialization did not particularly help train modules. Also, random initialization schemes were not particularly effective in context modules. The following expression succeeded in initializing more effectively. $k^b(t,a) = 1_{[t=0]} 1_{[a=b]}$

2.4 Front-End Module

In this paper, the neural network for inserting the dilated convolution is somewhat modified in VGG16[7][9]. Two pooling and striding layers at the end of the network were deleted. Specifically, all subsequent layers were dilated by 2. The convolution of the last layer was dilated by four. It can produce high-resolution outputs but initialize parameters of the original classification neural network. The front-end module produces a feature map with a resolution of 64×64, and pads images as input.

The training set used Pascal VOC 2012[4]. Training used SGD, mini-batch size used 14, and learning rate used 10^{-3}. The momentum is 0.9. The network trained a total of $60K$ iterations. We found out how well the front-end module performs compared to the FCN-8s[7]. It also used DeepLab network[1].

2.5 Experiments

They used the Caffe library. The additional images used Microsoft's COCO dataset[6]. Training consists of two main stages. The first step is to train VOC-2012 and Microsoft COCO images together. they iterate $100K$ for a learning rate of 10^{-3}, and solve a learning rate of 10^{-4} repeatedly as much as $40K$ subsequent. The second step is that only VOC-2012 images get fine-tuning. Fine-tuning trains $50K$ with a learning rate of 10^{-5}. On the training set, IoU was calculated at 69.8%, and on the test set, it was calculated at 71.3%.

2.6 Conclusion

A convolutional network is newly configured based on VGG16 for dense preference[11]. The researchers showed that the dense preference suit the dilated convolution operator. This is because any loss of coverage or resolution, it has the ability to expand the receptive field. The new network improves the accuracy of the existing semantic segmentation system. Finally, eliminating vestigial components was able to increase the accuracy of the existing convolutional neural networks of semantic segmentation.

3 Reference

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[2] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655. PMLR, 2014.

[3] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[4] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *2011 International Conference on Computer Vision*, pages 991–998. IEEE, 2011.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[7] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[8] Mark J Shensa et al. The discrete wavelet transform: wedding the a trous and mallat algorithms. *IEEE Transactions on signal processing*, 40(10):2464–2482, 1992.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[10] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[11] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.