Addition Residual Learning to Fully Convolutional DenseNets for Semantic Segmentation

Gyeongchan Yun
UNIST
rugyoon@unist.ac.kr

Abstract

Recently, Convolutional Neural Network (CNN) has been developing dramatically in semantic image segmentation. The typical semantic segmentation architecture adopts a downsampling path, an upsampling path, skip connections.

In this paper, I analyze Fully Convolutional DenseNet (FC-DenseNet) which is extended from DenseNet where the main characteristics is an iterative concatenation of previous feature maps. Moreover, I propose to adopt residual learning with dense block in downsampling path to exploit feature reuse and improve connectivity within downsampling and upsampling phases. I evaluate my proposed method with FC-DenseNet103 which is considered as state-of-the-art model. As a result, the proposed method improves state of-the-art performance on challenging urban scene understanding datasets (CamVid).

1. Introduction

Convolutional Neural Network (CNN) is developing dramatically in computer vision tasks such as image classification [8, 9, 16], object detection [13, 14], and semantic segmentation [11, 15].

Fully Convolutional Network (FCN) [11, 15] is proposed as a variant form of CNN for image classification to tackle pixel-wise prediction problem such as semantic segmentation. The main difference is to add upsampling layers to conventional CNN to recover loss of spatial information after pooling. To acheive fine-grained information recovery, FCN adopts a skip connection between their downsampling path and upsampling path.

In a trend of layers of CNN being deeper, *ResNet* [8] is proposed to point out the limitation of very deep CNN architecture [16]. As shown in Figure 1, ResNet adopts residual learning to every few stacked layers inspired by skip connection. Moveover, a new CNN architecture, called *DenseNet*, is introduced in [9]. DenseNet consists of dense block and pooling, where each dense block is an iterative concatenation of previous feature maps. DenseNet and ResNet have been extended to works as FCN [7, 10] for semantic segmentation.

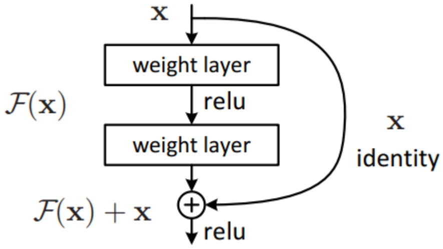
x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 1. Residual learning: a building block with identity mapping

In this paper, I focus on Fully Convolutional DenseNet (*FC-DenseNet*) [10] which is extended from DenseNet to solve semantic segmentation problem. FC-DenseNet can inherit the advantages of DenseNet: (1) parameter efficiency (2) implicit deep supervision (3) feature reuse. FC-DenseNet leverages dense block with transition down, transition up, and skip connection. As lots of feature maps can be produced with the concatenation of dense block, building simply upsampling path would result in a computationally intractable burden. To solve this problem, FC-DenseNet only upsample the feature maps created by last dense block of downsampling path.

I have found out that two opportunities for development in FC-DenseNet. First, due to memory demanding problem, transposed convolution in upsampling path is applied only to the feature maps obtained by the last dense block not all feature maps from previous dense block. Therefore, inserting shortcut connections followed by last dense block from previous dense block helps upsample more information contained feature maps shown in Figure 4. Second, some information from earlier dense blocks is lost in the downsampling path due to pooling. The paper insists that skip connection from downsampling path to upsampling path can prevent the loss of information. Changing 1x1 convolution to residual learning in transition down can contain correlated features. As a result, I expect it would result in less amount of information loss after pooling operation.

As I want to compare the baseline model, FC-DenseNet, with my extended algorithm on the same datasets, I evaluate my model on benchmark for urban scene understanding, CamVid [6] used in [10].

2. Related Work

Recent research in semantic segmentation have been dived into improving architectural designs by improving the upsampling path and increasing the connectivity within FCN [5, 15, 12].

For aspect of the resolution recovery, skip connections from the downsampling to the upsampling path have been adopted to allow for a fine-grained information recovery [15].

In order to reduce the loss of spatial information in downsampling path, stochastic pooling [17] was proposed, where randomly picking the activation within each pooling region.

3. Fully Convolutional DensNets with Residual Learning

As shown in Figure 2, FC-DenseNet are built from DenseNet architecture with downsampling path, upsampling path and skip connections. Skip connections help the upsampling path recover spatially detailed information from the downsampling path by reusing features maps. The goal of addition residual block in the downsampling path is to further exploit the feature reuse by extending the more sophisticated FC-DenseNet architecture, while remaining to avoid the feature explosion at the upsampling path of the network.

3.1. Review of FC-DenseNets

FC-DenseNet has been extended from DenseNet to work as FCN. Note that the downsampling part of FC-DenseNet is the same architecture of DenseNet. Note that, in the downsampling path, the linear growth in the number of features is compensated by the reduction in spatial resolution of each feature map after the pooling operation. A *transition down* is introduced to reduce the spatial dimensionality of the concatenated feature maps from dense block. In FC-Densenet, the last layer of the downsampling path is referred to as *bottleneck.*

In FC-DenseNet, they replace transposed convolution operation by a dense block and an upsampling operation referred to as *transition up*. Transition up modules consist of a transposed convolution that upsamples the previous feature maps. The upsampled feature maps are then concatenated to the ones coming from the skip connection to form the input of a new dense block.

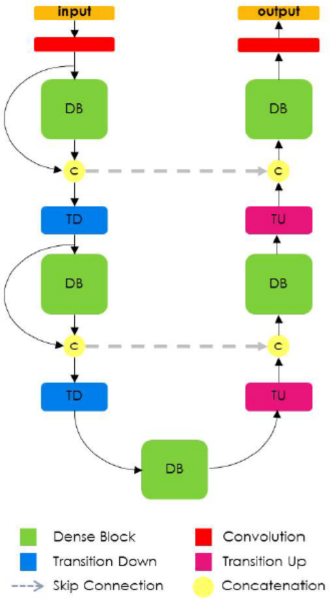

Figure 2. Diagram of FC-DenseNet's architecture.

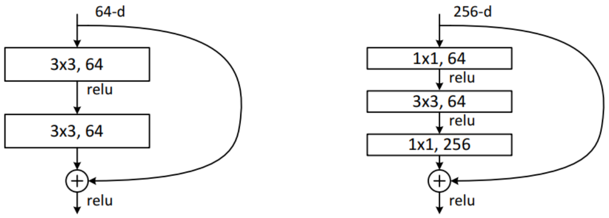

Figure 3. Left: Residual building block. Right: Residual ”bottleneck” block.

3.2. Residual Block

A residual building block is shown in Figure 1. The operation F + x is performed by a shortcut connection and element-wise addition.

A left part of Figure 3 illustrates insertion of shortcut connections to few stacked layers. Identity shortcuts are particularly important for not increasing the complexity of the bottleneck architectures that are introduced in [8]. As shown in right part of Figure 3, the three layers are 1 × 1, 3 × 3, and 1 × 1 convolutions, where the 1 × 1 layers are responsible for reducing and then increasing dimensions, leaving the 3×3 layer a bottleneck with smaller input/output dimensions.

3.3. Residual Block with Bottleneck

Since the upsampling path increases the feature maps spatial resolution, the linear growth in the number of features would be too memory demanding. In order to tackle this problem, transposed convolution in transition up is ap-

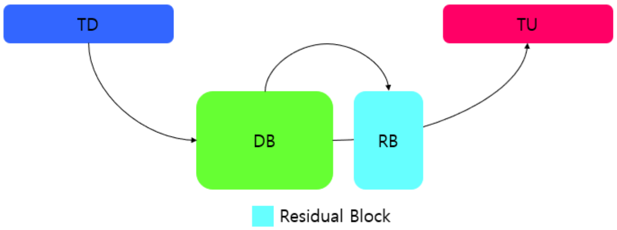

Figure 4. Diagram of Residual Block with Bottleneck.

Transition Down (TD)
Batch Normalization (BN)
ReLU
1 x 1 Convolution
Dropout p = 0.2
2 x 2 Max Pooling

New Transition Down
Residual Block
2 x 2 Max Pooling

Table 1. Left: Transition Down (TD) in FC-DenseNet. Right: Residual block in Transition Down (TD).

plied only to the feature maps obtained by the last dense block not all feature maps from previous dense block shown in bottom part of Figure 2.

Rather than applying only the most recent feature map from last dense block to upsampling path due to memory problems, it is helpful to place the residual block directly followed by the bottleneck (last dense block in downsampling path) to establish a shortcut connection with the previous dense block shown in Figure 4. Therefore, it can hand over the more correlated feature map to the transition up.

3.4. Residual Block in TD

The FC-DenseNet paper explains that some information from earlier dense blocks is lost in the downsampling path due to pooling. The paper insists that skip connection from downsampling path to upsampling path can pass the available information in transition down.

There is a question that cross on my mind: What if further information contained feature maps is passed over transition down not to modify upsampling path? In Table 1, transition down in FC-DenseNet is composed of BN, followed by ReLU, a 1 × 1 convolution, dropout with p = 0.2 and a non-overlapping max pooling of size 2 × 2. Therefore, I propose to change 1x1 convolution to residual learning in transition down, and it can contain correlated features.

3.5. Implementation

An official code of FC-DenseNet [2] is implemented by Theano [3]. As I am familar with Tensorflow [4], I use the baseline public code followed by link [1]. I only modify FC_DenseNet_Tiramisu.py in the baseline code. I implement residual bottleneck block illustrated in Figure 3, named ResidualBlock(), using python and tf.contrib.slim API. Then, ResidualBlock() is called in TransitionDown() and followed by bottelneck.

4. Experiments

I choose the baseline model as FC-DenseNet103 which is the state of arts model in [10]. I evaluate my method on urban scene understanding datasets: CamVid [6]. I report the results using the global accuracy (pixel-wise accuracy on the dataset).

4.1. Training Details

All hyper-parameters (optimizer, learning rate, weight decay) used for training is the same as in [10]. I run the experiments for 110 epoch where validation accuracy doesn't increase at all.

4.2. Results on CamVid dataset

In Table 2, the baseline model is FC-DenseNet103. *FC-DenseNet103-RBB* denotes FC-DenseNet103 consisting of Residual Block with Bottleneck. *FC-DenseNet103-RB* denotes two proposed additional residual block mentioned in Section 3.3, 3.4 is applied to FC-DenseNet103.

I also report the results of FC-DenseNet103 as baseline from the public baseline code [1] not that of the paper [10]. As expected, the result of FC-DenseNet103-RBB is superior to FC-DenseNet103. Interestingly, FC-DenseNet103-RB is worse the aspects of parameter efficiency and accuracy compared to FC-DenseNet103-RBB. However, it still improves performance compared to FC-DenseNet103. Residual block in transition down doesn't have effect of mitigating the loss of spatial information as much as I expected. I have analyzed that it may have the effect of offsetting the advantages of concatenation of feature maps in dense block.

Figure 5 shows one of the qualitative segmentation results on the CamVid dataset.

5. Discussion

In the case of *FC-DenseNet103-RBB* in Table 2, the model has parameter efficiency even addition to residual block while increasing global accuracy compared to FC-DenseNet103.

6. Conclusion

In this paper, I propose addition residual learning to FC-DenseNet which made DenseNet fully convolutional to tackle the problem of semantic image segmentation. I design placing residual block in downsampling path of FC-DenseNet to exploit feature reuse and connectivity within FCN-like model. In conclusion, my proposed method improves state of-the-art performance on challenging urban scene understanding datasets (CamVid).

Model	# parameters (M)	Building	Tree	Sky	Car	Sign	Road	Pedestrian	Fence	Pole	Sidewalk	Cyclist	Global accuracy
FC-DenseNet103	9.2	86.2	**97.4**	**98.4**	**94.3**	43.5	98.6	**80.6**	77.4	29.8	94.5	38.5	91.0
FC-DenseNet103-RBB	9.3	**92.2**	96.7	98.3	93.7	**46.4**	**98.9**	64.4	**88.7**	28.7	94.5	50.2	**92.5**
FC-DenseNet103-RB	18.0	89.8	95.7	98.3	88.7	43.3	98.8	61.0	87.0	**40.1**	**96.5**	**56.9**	92.3

Table 2. Results on CamVid dataset with 11 classes.

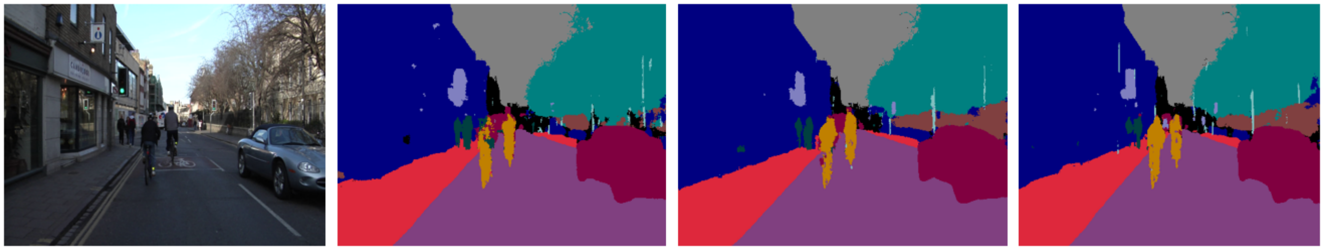

Figure 5. Qualitative results on the CamVid test set. From left to right: original, FC-DenseNet103, FC-DenseNet103-RBB, FC-DenseNet103-RB.

References

[1] https://github.com/AI-slam/FC-DenseNet-Tiramisu. 3

[2] FC-DenseNet. https://github.com/SimJeg/FC-DenseNet. 3

[3] Theano. https://github.com/Theano/Theano. 3

[4] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. 2016. 3

[5] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 2

[6] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *European Conference on Computer Vision (ECCV)*, 2008. 2, 3

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. 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. 1

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 2

[9] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 4700–4708, 2017. 1

[10] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 11–19, 2017. 1, 2, 3

[11] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[12] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015. 2

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[14] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1

[15] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1, 2

[16] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[17] M. D. Zeiler and R. Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*, 2013. 2