DEEP LEARNING LAB(EEE4423-01) Week 6 : FCN and DilatedNet

Hyeon Si Eun[1], 2019142214
[1] Department of Electric and Electronic Engineering, Yonsei University

1 FCN : Fully Convolutional Networks for Semantic Segmentation

The paper "Fully Convolutional Networks for Semantic Segmentation"[2] proposes a novel method for semantic segmentation using fully convolutional networks (FCN).

Semantic segmentation refers to the process of assigning a label to each pixel in an image, such that all pixels with the same label belong to the same object or class. This is a fundamental task in computer vision, with applications in various fields such as autonomous driving, robotics, and medical image analysis.

Traditional approaches for semantic segmentation involve using a combination of handcrafted features and classifiers, which can be time-consuming and limited in their ability to capture complex patterns in the data. In contrast, FCNs are end-to-end trainable networks that can learn to perform pixel-level segmentation directly from the input image.

The main contribution of the paper is the introduction of an end-to-end trainable FCN architecture that can perform dense predictions for semantic segmentation tasks. The authors propose a novel approach that uses upsampling layers to transform low-resolution feature maps into high-resolution segmentation maps.

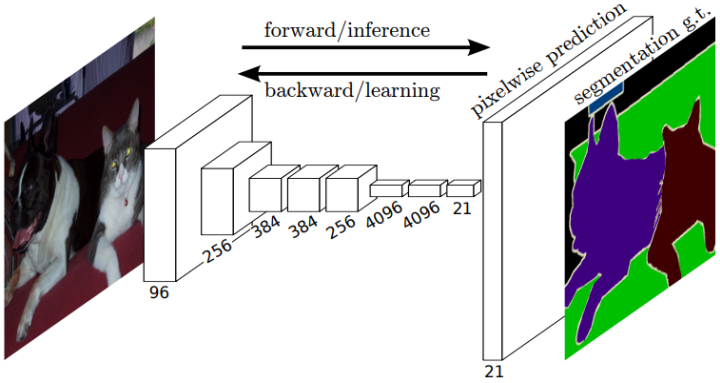

Figure 1: Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

FCN is a purpose-built transformation of CNN-based models (AlexNet, VGG-16, GoogLeNet) that have shown excellent performance in image classification for the Semantic Segmentation model.

This 'Image classification model to Semantic segmentation model' can be largely expressed in the following three processes:

- Convolutionalization
- Deconvolution (Upsampling)
- Skip architecture

1. Convolutionalization

The authors proposed a modification to the popular convolutional neural network (CNN) architecture, which allows it to perform semantic segmentation. Specifically, they replaced the fully connected layers in a CNN with convolutional layers, resulting in a network that takes an image of any size as input and produces a dense classification map as output.

Image classification models basically consist of a Fully-Connected (fc) layer as an output layer for the fundamental goal of the model regardless of the internal structure. This configuration is to extract image features using ConvNet from the input layer to the middle of the network and to classify the

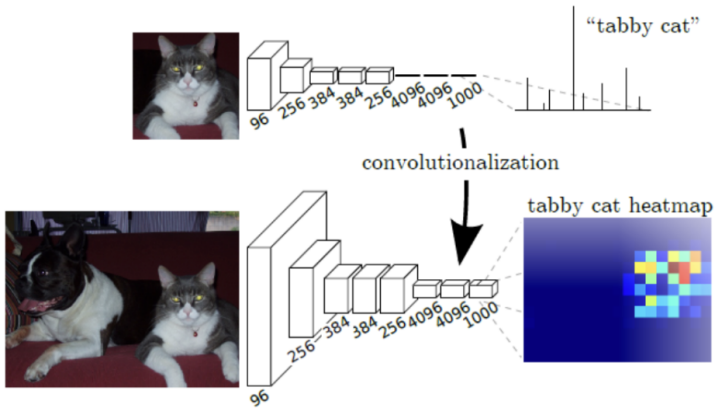

Figure 2: Transforming fully connected layers into convolution layers enables a classification net to output a heatmap. Adding layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end dense learning.

image using fc from the output layer. However, from the viewpoint of Semantic Segmentation, there is a limitation of the fc layer.

1. The location information of the image disappears.
2. The input image size is fixed.

The purpose of segmentation is to classify each pixel of the original image and to segment instances and backgrounds. So, location information is very important. In order to compensate for the limitations of these fc-layers, a method of replacing all fc-layers with 1x1 Conv-layers was chosen like Figure2.

2. Deconvolution

As the input passes through conv + pooling, its size decreases. The shift-and-stitch method was also considered as a method of restoring this, but upsampling was judged to be more effective in this paper. In FCN, Bilinear interpolation and Backwards convolution are used to upsample coarse feature maps into dense maps. Upsampling is performed on the network for end-to-end learning by backpropagation at per-pixel loss. Depending on their use in deconvolution networks, these layers are sometimes called deconvolution layers. In experiments, the authors found that upsampling within the network was effective in learning fast and dense predictions. And with patchwise training, the class imbalance problem arises, and fully convolutional training is better in terms of speed and efficiency.

3. Skip architecture

To further refine the segmentation results, the authors introduced a technique called "skip architecture", which involves merging the feature maps from lower layers of the network with those from higher layers. This allows the network to capture both local and global context, leading to more accurate segmentation results. The authors define a skip architecture that combines the semantic information of the Deep and Coarse layer and the apparent information of the Shallow and Fine layer to obtain accurate and detailed segmentation.

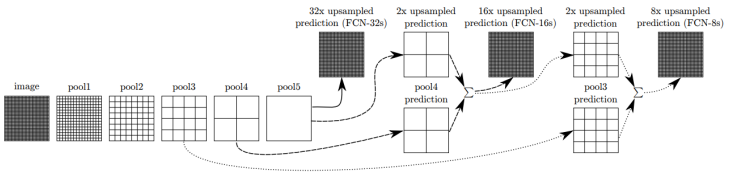

Figure 3: The Skip architecture

FCN-32 in the paper refers to the result of upsampling to 32x32 after taking a 32x32 image as input and extracting a 1x1 feature map. In this way,

This is an extended abstract. The full paper is available at the this paper and this paper

when a lot of pooling processes are performed, information loss is large, so even if upsampling is performed to match the size of the image, it is difficult to show good performance in segmentation due to the lack of fine information. Therefore, the key to skip architecture is to utilize intermediate feature maps that have undergone less pooling.

The FCN architecture is trained using a modified version of the cross-entropy loss function, which takes into account the sparsity of the ground truth annotations. It is trained with per-pixel multinimial logistic loss and evaluated with the standard metric of mean pixel intersection over union. Learning ignores pixels that deviate from the ground truth.

The authors evaluated their approach on several benchmark datasets for semantic segmentation, including PASCAL VOC, NYUDv2, and SIFT Flow. They showed that their approach outperformed previous state-of-the-art methods on these datasets in terms of segmentation accuracy and speed. If you want to see detailed experimental results, I recommend viewing the full paper.

Overall, the paper presents a simple and effective method for semantic segmentation that has since become a widely used baseline in the field of computer vision.

2 DilatedNet : Multi-Scale Context Aggregation by Dilated Convolutions

The paper "Multi-Scale Context Aggregation by Dilated Convolutions"[3] proposes a novel method for capturing multi-scale context in convolutional neural networks (CNNs) using dilated convolutions.

The main motivation for this work was the observation that CNNs with large receptive fields (i.e., the region of the input that a filter is sensitive to) can capture context at multiple scales. However, using large filters can be computationally expensive and can lead to overfitting. To address these issues, the authors propose dilated convolutions, which allow for large receptive fields without increasing the number of parameters or the computational cost.

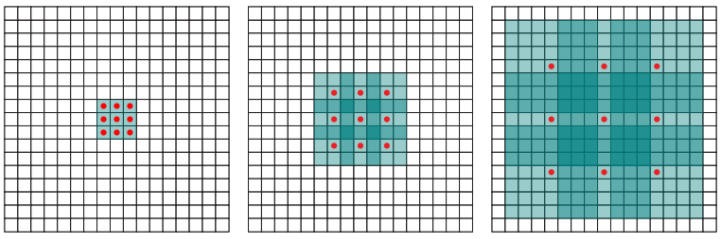

Figure 4: Systematic dilation supports exponential expansion of the receptive field without loss of resolution or coverage. The number of parameters associated with each layer is identical. The receptive field grows exponentially while the number of parameters grows linearly.

The filter size of (a) is 3x3 and has 9 parameters. (b) is the case where the dilated factor is 2. Except for the red dot, the weights of the filter are filled with 0, so it has 9 parameters and the receptive field is 7x7. In other words, without applying maxpooling or conv with a stride of 2, the receptive field can be expanded, reducing the loss of spatial information, and having the advantage of maintaining the amount of computation and expanding the receptive field.

Dilated convolutions work by inserting gaps between the filter elements, effectively expanding the filter's receptive field. Specifically, a dilated convolution with dilation rate r can be seen as a regular convolution with a filter that has r-1 zeros inserted between every two adjacent elements. By using dilated convolutions with different dilation rates in parallel, the authors were able to capture multi-scale context efficiently.

The discrete convolution operator $*$ can be defined as:

$$(F * k)(\mathbf{p}) = \sum_{s+\mathbf{t}=p} F(\mathbf{s})k(\mathbf{t}) \tag{1}$$

And They generalize this operator. Let l be a dilation factor and let $*_l$ be defined as

$$(F *_l k)(\mathbf{p}) = \sum_{s+l\mathbf{t}=p} F(\mathbf{s})k(\mathbf{t}) \tag{2}$$

They refer to $*_l$ as a dilated convolution or an l-dilated convolution. The familiar discrete convolution $*$ is simply the 1-dilated convolution.

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: Context network architecture. The network processes C feature maps by aggregating contextual information at progressively increasing scales without losing resolution.

DilatedNet proposes the context module. The context module improves the performance of dense prediction by synthesizing multi-scale contextual information. The feature of this module is that it outputs the feature map of the C channel when it receives the feature map of the C channel. Since the size and number of channels of input and output are the same, it can be used for other dense prediction structures. It consists of 8 layers of dilated conv. Truncation means use ReLU after conv operation. You can see the Context network architecture in Figure5. And they found that random initialization schemes were not effective for the context module. We found an alternative initialization with clear semantics to be much more effective:

$$k^b(\mathbf{t}, a) = 1_{[t=0]} 1_{[a=b]} \tag{3}$$

where a is the index of the input feature map and b is the index of the output map. This is a form of identity initialization, which has recently been advocated for recurrent networks.[1]

The 'front end module' is a kind of backbone connected to the front end of the context module. It receives 3-channel images as input and outputs a feature map of C=21. It uses the VGG-16 net and removes the last two pooling operations to make the output size 64x64. Also, change all convolutions in conv block 5 to 2-dilated convolutions. Change the first fc-1 after conv block 5 to a 4-dilated convolution. It receives input images of 3 channels, creates 64x64 feature maps of 21 channels, and passes the generated feature maps to the context module.
Front end module and context module of this paper were learned in the following order. At this time, the joint training did not have much effect.
1. Learn Front end module by inputting RGB image.
2. Learn the context module by inputting the output feature map of the learned front end module.

The authors evaluated their approach on several image recognition tasks, including object recognition, semantic segmentation, and depth estimation. They found that dilated convolutions outperformed traditional convolutions and other methods for capturing multi-scale context, such as pooling and pyramid architectures. If you want to see detailed experimental results, I recommend viewing the full paper.

In summary, the paper "Multi-Scale Context Aggregation by Dilated Convolutions" proposes dilated convolutions as an efficient way to capture multi-scale context in CNNs. The approach was shown to outperform traditional convolutions and other methods on several image recognition tasks.

[1] Quoc V Le, Navdeep Jaitly, and Geoffrey E Hinton. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*, 2015.

[2] 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.

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