FCN for Semantic Segmentation Summary

Roger Marí [1], Joan Sintes [2], Àlex Palomo [3], Àlex Vicente [4]
Universitat Pompeu Fabra, Image Processing and Computer Vision Group
{ [1]roger.mari01, [2]joan.sintes01, [3]alex.palomo01, [4]alex.vicente01 }@estudiant.upf.edu

***Abstract*—This is a summary of the original paper by Jonathan Long, Evan Shelhamer and Trevor Darrell [1].**

I. Introduction

In recent years deep learning models proved to be extremely successful for image classification. Semantic segmentation, which consists in assigning a label to each pixel (and not just one label to the whole image) is a natural next step.

Semantic segmentation is a more challenging task: it is a dense prediction problem. First works for dense prediction with convnets had some points in common, including:

- Small models, restricting capacity and receptive fields.
- Patchwise training, which is not efficient at all.
- Several post processing techniques (superpixel projection, random field regularization, filtering, etc.).

The authors of the paper proposed to get rid these restrictions and refinement steps by training a fully convolutional network (FCN) end-to-end, pixels-to-pixels. The idea is very simple: adapt a classification network (e.g. VGG or GoogleNet) into a FCN and, using the weights pre-trained for classification as initialization, fine-tune for semantic segmentation.

II. Network Architecture

A. Adapting classifiers for dense prediction

The last block of fully connected layers of most classification networks requires that these nets take fixed-sized inputs, while producing non-spatial outputs (that is, 1D vector with class probabilities, no spatial information preserved).

To convert a classifier model to a FCN capable of dense prediction, the fully connected layers are replaced by conv. layers with kernels that cover the entire input region. Then a 1x1 conv. with depth d equal to the number of classes followed by softmax activation is appended at each of the coarse output locations to predict the d class probabilities for each pixel. This way the model can deal with inputs of any size. The problem is that classification models typically include subsampling via strided convolutions or max pooling to keep computational cost reasonable. Consequently, semantic segmentation requires a method for upsampling to resize the coarse output to the same size as the input (remember, one label per pixel).

B. Upsampling via backwards strided convolution

Upsampling with factor$f$ can be seen as a convolution with an input stride of $1/f$. This is equivalent to a backward convolution with an output stride of f. Such operation simply reverses the forward and backward passes of convolution (i.e. a transposed convolution). Using this trick, upsampling is integrated in-network for end-to-end learning via backpropagation.

C. Combining what and where

The authors finally propose to combine the different layers of the FCN to refine the spatial precision of the output. This is done via skip connections, that fuse the output from shallower layers (which provide precise spatial information, the *where*) with the output of deeper layers (which provide precise semantic information, the *what*). In the fusion (i.e. summing) the coarsest maps are upsampled following the technique explained in section II-B.

III. Training process and Experiments

Mini-batch gradient descent, with batch size 20 and momentum 0.9, was used for weight optimization (at least 175 epochs). Weight decay of 5^{-4} or 2^{-4} was used and dropout was included where used in the original classification models. The learning rate was set to 10^{-3}, 10^{-4} and 5^{-5} for FCN-AlexNet, FCN-VGG16 and FCN-GoogLeNet respectively.

The authors trained with a per-pixel multinomial logistic loss and validated with the standard metric of mean pixel intersection over union. They claim that training from scratch is not feasible, so they use pre-trained weights on image classification for initialization. After that, all the layers of the FCN are fine-tuned by backpropagation through the whole net. Fine-tuning the output classifier alone yelds 70% of the final performance achieved by fine-tuning all layers.

The experiments were carried using PASCAL VOC 2011 segmentation challenge dataset. The best results were achieved using FCN-8s, which uses the VGG16 conv. blocks and fuses the output maps of blocks 3, 4 and 5 for the final dense prediction. FCN-8s results on PASCAL VOC 2011 (test set): mean IoU 62.7%. pixel accuracy 90.3%, mean pixel accuracy 75.9%, inference time around 175 ms. This performance outperformed all the previous state-of-the-art methods.

IV. Conclusion

The authors of the paper proposed to use fully convolutional networks, trained end-to-end, for image semantic segmentation. Unlike previous methods, they extended classification architectures to produce pixel-wise output. Classification pre-trained weights are used for initialization, and then all layers are fine-tuned for segmentation. The feature maps of shallower layers are fused with those of deeper layers to keep compromise between semantic information and spatial precision.

References

[1] 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*, 2015, pp. 3431–3440.