SegNet 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 Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla [1].**

I. Introduction

The authors of SegNet first started working on this segmentation network in 2015, shortly after the break in of fully convolutional architectures for image segmentation [2]. In 2017 they presented the latest version of the model.

The SegNet was motivated by the fact that the previous segmentation results, although encouraging, appeared to be coarse. SegNet was primarily conceived for road scene understanding applications.

II. Network Architecture

SegNet has an encoder network and a corresponding decoder network, followed by a final pixelwise classification layer. The encoder net consists of 13 convolutional layers which correspond to the first 13 convolutional layers in the VGG16 model for classification (the fully connected layers are discarded). Each encoder layer has a corresponding decoder layer and hence the decoder network has also 13 layers. The final decoder output is fed to a multi-class soft-max classifier to produce class probabilities for each pixel independently.

Each encoder block performs a convolution, followed by a batch normalization and an ReLU non-linearity. At the end of each VGG block, a max-pooling with 2x2 window and stride 2 (non-overlapping window) is applied to subsample the feature maps by a factor of 2. The increasingly lossy representation is not beneficial for segmentation where boundary delineation is vital. It is necessary to capture the boundary information somehow. If memory constraints are not present, the authors recommend to store all the encoder feature maps. Otherwise, they propose to store only the indices of the max-pooling, which is an efficient way to keep track of spatial information.

The decoder blocks in the decoder network upsample the input feature maps using the pooling indices from the corresponding encoder feature maps. This step produces sparse feature maps, which are convolved to produce dense feature maps. A batch normalization step is then applied to each of these maps. Note that the decoder block corresponding to the first encoder block (the closest to the input image) produces a multi-channel output, and not a 3-channel output (like the input RGB image received by the first encoder block). This is unlike the rest of decoders, which produce feature maps with the same size and channels as their encoder inputs. The output of the last decoder is fed to a soft-max classifier so that the final output of the net is a d channel image of probabilities, where d is the number of classses.

A. Comparison with U-Net

Compared to SegNet, U-Net does not reuse pooling indices but instead transfer the entire feature map (at the cost of more memory) to the corresponding decoders and concatenates them to upsampled (via deconvolution) decoder feature maps. Also, the U-Net does not use a 5th conv. block, while the SegNet uses all VGG16 blocks.

III. Training process and Experiments

The authors trained the SegNet on CamVid road scenes dataset and SUN RGB-D indoor scenes dataset.

The VGG16 pre-trained weights for image classification can be used to initialize the encoder network. Mini-batch gradient descent, with batch size 12 and momentum 0.9, was used for weight optimization (1000 epochs maximum, stop at convergence). The learning rate was set to 0.1.

Cross-entropy loss was used to train. The loss is summed up over all pixels in a mini-batch. Since there is large variation in the number of pixels in each class (e.g. road and sky classes have more pixels than car or pedestrian classes), *class balancing* is used. In particular, *median frequency balancing*: the weight assigned to each class in the loss function is the ratio of the median of class frequencies (computed over the entire training set) divided by the class frequency.

The authors also carried experiments without class balancing, or with other variants to upsample the feature maps at the decoding stage (upsample via replication or a fixed indices array, etc.). They also propose a reduced and compact version of the model, *SegNet-Basic*, with only four encoders and four decoders. No ReLU is used in the decoders in this variant, and a constant kernel size of 7x7 is used in all convolutions to provide a wide context for smooth labelling.

IV. Conclusion

The authors of the paper proposed an encoder-decoder network with almost symmetric structure for image semantic segmentation. The VGG16 blocks are recycled for the encoder part, and unpooling via stored indices is used in the decoder to improve contours delineation. The network can be trained end-to-end just like any fully convolutional model.

References

[1] 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*, vol. 39, no. 12, pp. 2481–2495, 2017.

[2] 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.