Fully Convolution Networks for Semantic Segmentation

Dimitrios Dagdilelis (s182589), Mitul Vekariya (s181344)

Technical University of Denmark

Semantic Segmentation

Fully **C**onvolutional **N**etworks[3] provide a solution to the image segmentation task, which is trying to answer to the question of 'where' is something instead of just 'what'. The FCN complements the CNN architecture by following up the **"downsampling path"** with an **"upsampling path"**. Upsampling methods can include both learnable and non-learnable parameters.

Key points

- The use of **CNN** layers in the vanilla **FCN** architecture is reducing the structural information in the image as data flows through the *"downsampling path"*, hence upsampling the low-dimensional information in the *"upsampling path"* creates coarse bounding areas around the objects.
- The proposed architecture is leveraging the available structural information of the convolutional feature maps by forward passing information from their max-pooling layers to the up-sampling layers. Merging feature maps from various resolution levels helps combining context information with spatial information.
- The proposed architecture is leveraging transfer learning by initializing the convolutional layers from an accurate pre-trained classification architecture(VGGnet).
- The convolutional layers can be frozen or trainable. Performance of the architecture has been evaluated for both cases.
- The architecture enables end-to-end training of the network. We define different models by adding forward passing feature maps from various resolution levels.
  - ▷ FCN32s : The original FCN without max-pooling forward passes.
  - ▷ FCN16s : Forward passing max pooled information from *conv4* to the output of *upsample6*.
  - ▷ FCN8s : Extends FCN16s by upsampling the FCN16s skip connection, then fusing it with the max-pooled output of *conv3* and applying *upsample8*.
  - ▷ FCNs : Extends FCN8s by summing the pooled output of *conv2* with the FCN8s skip connection(not illustrated in Figure 1).

  Figure 1 illustrates the FCN8s architecture, the FCN16s is derived by cutting the skip2, and FCN32s by cutting both skip1 and skip2.

Forward passing architecture

Structural information is being extracted from the pooling layers of the down-sampling path. Then this information is being fused with the up-sampled information by element-wise addition before moving upstream. Batch normalization and dropout layers are being applied to the outputs of every convolution and up-sampling layer. ReLu was used as the activation function.

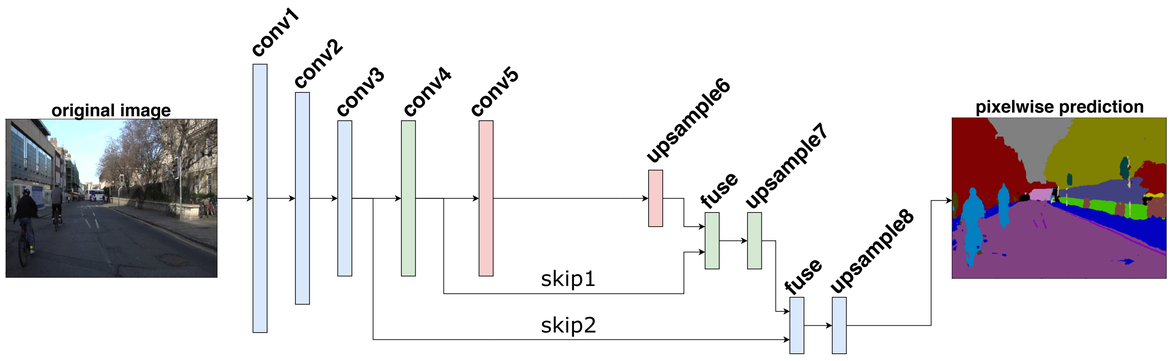

Figure 1: FCN8s Architecture

Dataset Information

The Cambridge driving Labeled Video Database[2] (CamVid) is used to train and validate the FCN network. CamVid dataset is comprised of video frames with pixelwise class semantic labels. The database provides ground truth labels that associate each pixel with one of 32 labels. The dataset has 701 images in total. The traning and test sets are chosen randomly. 631 images are used for training and 70 images are used for validation.

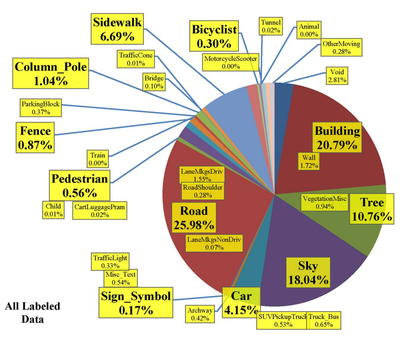

Figure 2: Camvid dataset class frequency[1]

Loss and Performance criteria

For the task of image segmentation pixel wise binary cross-entropy loss is used. It examines each pixels individually. Since the target vectors are one out of K encoded (total 32 class labels), the binary cross entropy is calculated using each encoded class and then averaged over all the 32 classes. This makes sure that each pixel is learning equally while training. However, we ran into problems because of non-weighted averaging. The CamVid dataset has significantly large number of big objects than small objects. Hence, even after the training, the network cannot segment small objects with same accuracy as bigger objects.

To compare the performance of the FCN, we will consider two measures : pixel accuracy and IoU (Intersection over Union)

$$\text{Pixel accuracy} = \frac{\text{No of pixels predicted accurately}}{\text{No of pixels}} \qquad \text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

The adaptive optimizer RMSprop was used. Rprop is trying to avoid saddle points in the minimization problem, by tuning the step size of each individual weight based on the gradient sign. RMSprop is extending the central idea of Rprop by including temporal information about the gradient in the sense that it is trying to keep the moving average of the squared gradient stable.

Best model performance

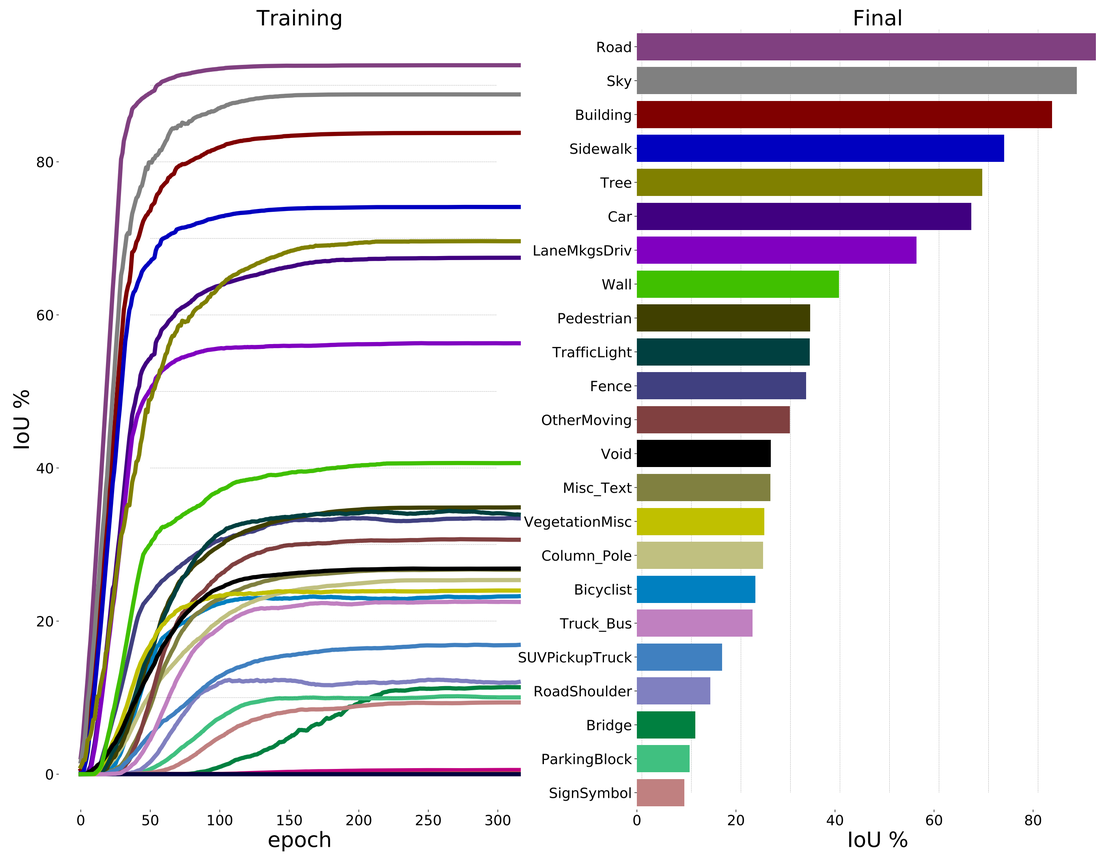

Figure 3: Training and Final performance on IoU for the best model, "FCN8s with trainable VGGnet as a backbone"

Model comparison

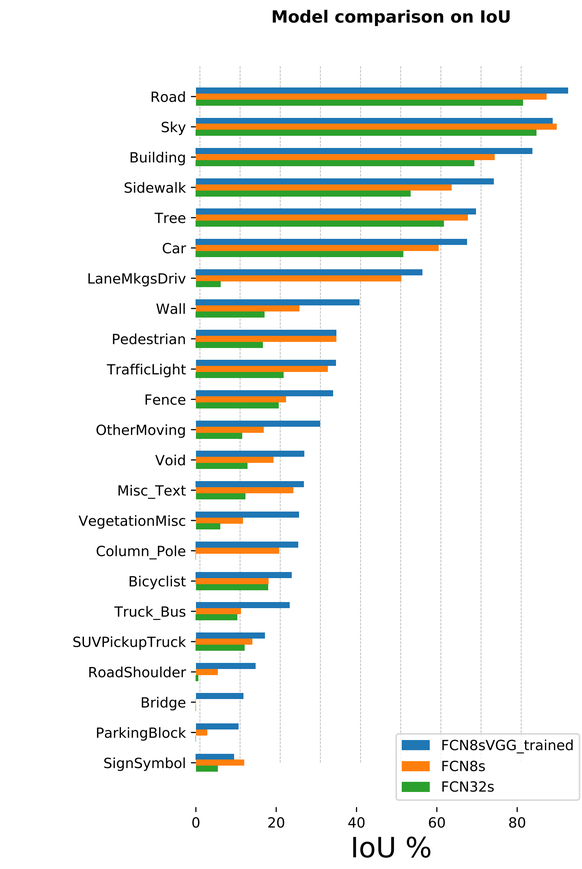

Figure 4: All models IoU classes comparison

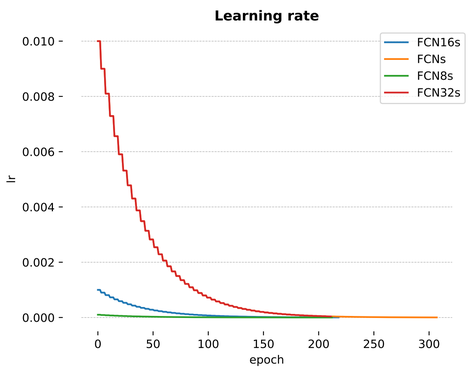

Figure 5: learning rate

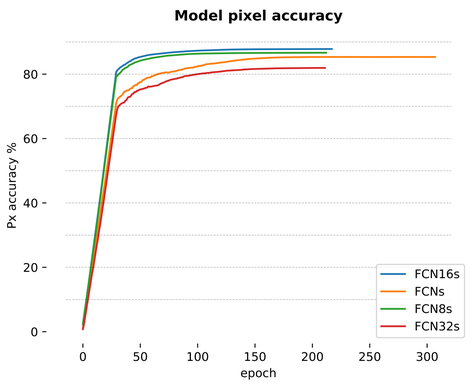

Figure 6: Models pixel accuracy

Segmentation output from different Networks

Figure 7: Input

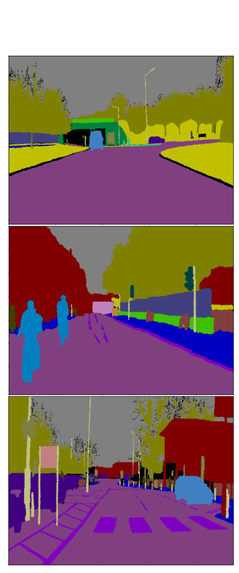

Figure 8: Target

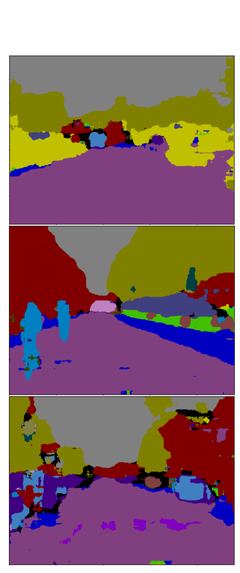

Figure 9: FCN32

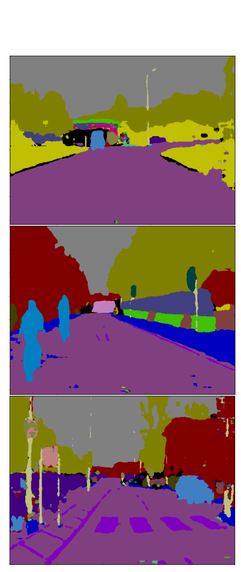

Figure 10: FCN16

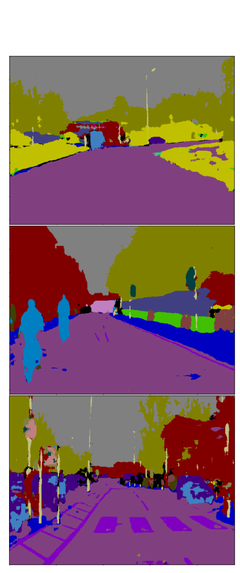

Figure 11: FCN8

References

[1] camvid class frequency. URL `http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/pr/DataPercents.jpg`.

[2] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV (1)*, pages 44–57, 2008.

[3] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, June 2015. doi: 10.1109/CVPR.2015.7298965.

s182589/s181344 @student.dtu.dk