EE-559 – Deep learning

7.4. Networks for semantic segmentation

François Fleuret

https://fleuret.org/ee559/

Mon Apr 1 13:50:32 UTC 2019

The historical approach to image segmentation was to define a measure of similarity between pixels, and to cluster groups of similar pixels.

The historical approach to image segmentation was to define a measure of similarity between pixels, and to cluster groups of similar pixels. Such approaches account poorly for semantic content.

The historical approach to image segmentation was to define a measure of similarity between pixels, and to cluster groups of similar pixels. Such approaches account poorly for semantic content.

The deep-learning approach re-casts semantic segmentation as pixel classification, and re-uses networks trained for image classification by making them fully convolutional.

Shelhamer et al. (2016) use a pre-trained classification network (*e.g.* VGG 16 layers) from which the final fully connected layer is removed, and the other ones are converted to 1×1 convolutional filters.

They add a final 1×1 convolutional layers with 21 output channels (VOC 20 classes + "background").

Since VGG16 has 5 max-pooling with 2×2 kernels, with proper padding, the output is $1/2^5 = 1/32$ the size of the input.

Shelhamer et al. (2016) use a pre-trained classification network (*e.g.* VGG 16 layers) from which the final fully connected layer is removed, and the other ones are converted to 1×1 convolutional filters.

They add a final 1×1 convolutional layers with 21 output channels (VOC 20 classes + "background").

Since VGG16 has 5 max-pooling with 2×2 kernels, with proper padding, the output is $1/2^5 = 1/32$ the size of the input.

This map is then up-scaled with a de-convolution layer with kernel 64×64 and stride 32×32 to get a final map of same size as the input image.

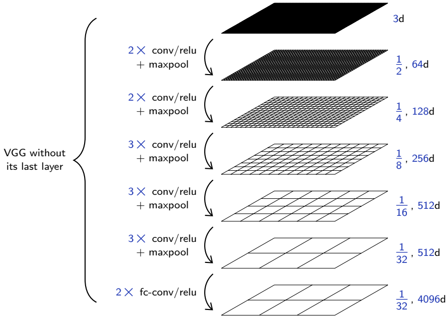
Shelhamer et al. (2016) use a pre-trained classification network (*e.g.* VGG 16 layers) from which the final fully connected layer is removed, and the other ones are converted to 1×1 convolutional filters.

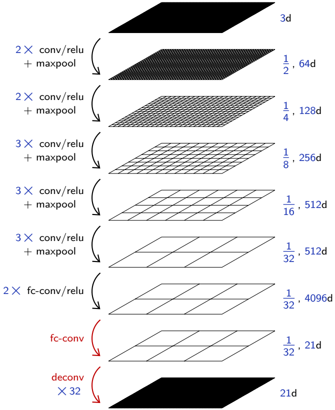
They add a final 1×1 convolutional layers with 21 output channels (VOC 20 classes + "background").

Since VGG16 has 5 max-pooling with 2×2 kernels, with proper padding, the output is $1/2^5 = 1/32$ the size of the input.

This map is then up-scaled with a de-convolution layer with kernel 64×64 and stride 32×32 to get a final map of same size as the input image.

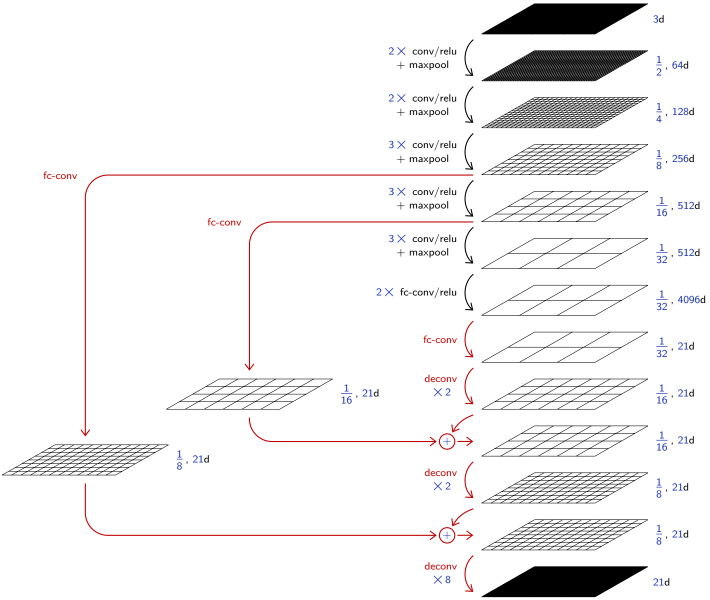
Training is achieved with full images and pixel-wise cross-entropy, starting with a pre-trained VGG16. All layers are fine-tuned, although fixing the up-scaling de-convolution to bilinear does as well.


VGG without
its last layer
2 × conv/relu
+ maxpool
2 × conv/relu
+ maxpool
3 × conv/relu
+ maxpool
3 × conv/relu
+ maxpool
3 × conv/relu
+ maxpool
2 × fc-conv/relu
3d
1/2, 64d
1/4, 128d
1/8, 256d
1/16, 512d
1/32, 512d
1/32, 4096d

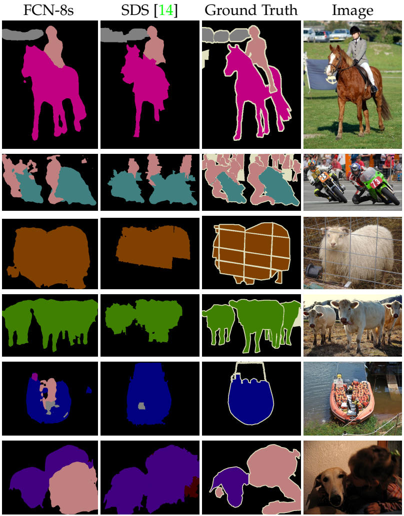

3d
2 × conv/relu
+ maxpool
1/2, 64d
2 × conv/relu
+ maxpool
1/4, 128d
3 × conv/relu
+ maxpool
1/8, 256d
3 × conv/relu
+ maxpool
1/16, 512d
3 × conv/relu
+ maxpool
1/32, 512d
2 × fc-conv/relu
1/32, 4096d
fc-conv
1/32, 21d
deconv
× 32
21d

Although this Fully Connected Network (FCN) achieved almost state-of-the-art results when published, its main weakness is the coarseness of the signal from which the final output is produced ($1/32$ of the original resolution).

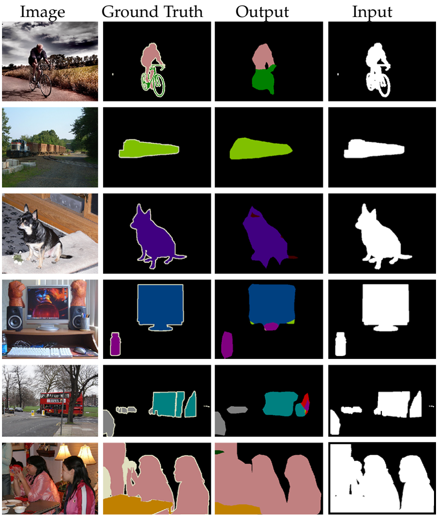
Shelhamer et al. proposed an additional element, that consists of using the same prediction/up-scaling from intermediate layers of the VGG network.



3d
2 × conv/relu
+ maxpool
$\frac{1}{2}$, 64d
2 × conv/relu
+ maxpool
$\frac{1}{4}$, 128d
3 × conv/relu
+ maxpool
$\frac{1}{8}$, 256d
3 × conv/relu
+ maxpool
$\frac{1}{16}$, 512d
3 × conv/relu
+ maxpool
$\frac{1}{32}$, 512d
2 × fc-conv/relu
$\frac{1}{32}$, 4096d
fc-conv
$\frac{1}{32}$, 21d
deconv
× 2
$\frac{1}{16}$, 21d
$\frac{1}{16}$, 21d
deconv
× 2
$\frac{1}{8}$, 21d
$\frac{1}{8}$, 21d
deconv
× 8
21d
fc-conv
fc-conv
$\frac{1}{16}$, 21d
$\frac{1}{8}$, 21d



Left column is the best network from Shelhamer et al. (2016).



Results with a network trained from mask only (Shelhamer et al., 2016).

It is noteworthy that for detection and semantic segmentation, there is an heavy re-use of large networks trained for classification.

The models themselves, as much as the source code of the algorithm that produced them, or the training data, are generic and re-usable assets.

The end

References

E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. CoRR, abs/1605.06211, 2016.