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 $1 \times 1$ convolutional filters.

They add a final $1 \times 1$ convolutional layers with 21 output channels (VOC 20 classes + “background”).

Since VGG16 has 5 max-pooling with $2 \times 2$ 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 $64 \times 64$ and stride $32 \times 32$ 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.
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.
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.

References