lects image regions for processing. However, their attention model is not differentiable, which is necessary for standard backpropagation during training. On the other hand, Gregor *et al*. [25] employ a differentiable attention model to specify where to read/write image regions for image generation.

Bahdanau *et al*. [5] propose an attention model that softly weights the importance of input words in a source sentence when predicting a target word for machine translation. Following this, Xu *et al*. [55] and Yao *et al*. [56] use attention models for image captioning and video captioning respectively. These methods apply attention in the 2D spatial and/or temporal dimension while we use attention to identify the most relevant scales.

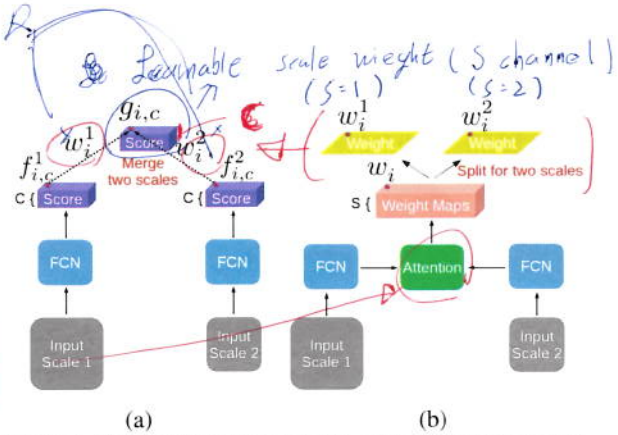
Attention to scale: To merge the predictions from multi-scale features, there are two common approachs: average-pooling [14, 15] or max-pooling [20, 44] over scales. Motivated by [5], we propose to jointly learn an attention model that softly weights the features from different input scales when predicting the semantic label of a pixel. The final output of our model is produced by the weighted sum of score maps across all the scales. We show that the proposed attention model not only improves performance over average- and max-pooling, but also allows us to diagnostically *visualize* the importance of features at different positions and scales, separating us from existing work that exploits multi-scale features for semantic segmentation.

3. Model

3.1. Review of DeepLab

FCNs have proven successful in semantic image segmentation [15, 37, 58]. In this subsection, we briefly review the DeepLab model [11], which is a variant of FCNs [38].

DeepLab adopts the 16-layer architecture of state-of-the-art classification network of [49] (*i.e*., VGG-16 net). The network is modified to be fully convolutional [38], producing dense feature maps. In particular, the last fully-connected layers of the original VGG-16 net are turned into convolutional layers (*e.g*., the last layer has a spatial convolutional kernel with size 1×1). The spatial decimation factor of the original VGG-16 net is 32 because of the employment of five max-pooling layers each with stride 2. DeepLab reduces it to 8 by using the à trous (with holes) algorithm [39], and employs linear interpolation to upsample by a factor of 8 the score maps of the final layer to original image resolution. There are several variants of DeepLab [11]. In this work, we mainly focus on DeepLab-LargeFOV. The suffix, LargeFOV, comes from the fact that the model adjusts the filter weights at the convolutional variant of fc_6 (fc_6 is the original first fully connected layer in VGG-16 net) with à trous algorithm so that its Field-Of-View is larger.



(a) (b)

Figure 3. (a) Merging score maps (*i.e*., last layer output before SoftMax) for two scales. (b) Our proposed attention model makes use of features from FCNs and produces weight maps, reflecting how to do a weighted merge of the FCN-produced score maps at different scales and at different positions.

3.2. Attention model for scales

Herein, we discuss how to merge the multi-scale features for our proposed model. We propose an attention model that learns to weight the multi-scale features. Average pooling [14, 15] or max pooling [20, 44] over scales to merge features can be considered as special cases of our method.

Based on share-net, suppose an input image is resized to several scales $s \in \{1, ..., S\}$. Each scale is passed through the DeepLab (the FCN weights are shared across all scales) and produces a score map for scale s, denoted as $f^s_{i,c}$ where i ranges over all the spatial positions (since it is fully convolutional) and $c \in \{1, ..., C\}$ where C is the number of classes of interest. The score maps $f^s_{i,c}$ are resized to have the same resolution (with respect to the finest scale) by bilinear interpolation. We denote $g_{i,c}$ to be the weighted sum of score maps at (i, c) for all scales, *i.e*.,

$$g_{i,c} = \sum_{s=1}^{S} w^s_i \cdot f^s_{i,c} \quad (1)$$

The weight w^s_i is computed by

$$w^s_i = \frac{\exp(h^s_i)}{\sum_{t=1}^{S} \exp(h^t_i)} \quad (2)$$

where h^s_i is the score map (*i.e*., last layer output before SoftMax) produced by the attention model at position i for scale s. Note w^s_i is shared across all the channels. The attention model is parameterized by another FCN so that dense maps are produced. The proposed attention model takes as input the convolutionalized fc_7 features from VGG-16 [49], and it consists of two layers (the first layer has 512 filters with kernel size 3×3 and second layer has S filters with kernel size 1×1 where S is the number of scales employed). We will discuss this design choice in the experimental results.