The weight w_i^s reflects the importance of feature at position i and scale s. As a result, the attention model decides how much attention to pay to features at different positions and scales. It further enables us to visualize the attention for each scale by visualizing w_i^s. Note in our formulation, average-pooling or max-pooling over scales are two special cases. In particular, the weights w_i^s in Eq. (1) will be replaced by $1/S$ for average-pooling, while the summation in Eq. (1) becomes the $\max$ operation and $w_i^s = 1 \; \forall s$ and i in the case of max-pooling.

We emphasize that the attention model computes a soft weight for each scale and position, and it allows the gradient of the loss function to be backpropagated through, similar to [5]. Therefore, we are able to jointly train the attention model as well as the FCN (*i.e.*, DeepLab) part end-to-end. One advantage of the proposed joint training is that tedious annotations of the "ground truth scale" for each pixel is avoided, letting the model adaptively find the best weights on scales.

3.3. Extra supervision

+ auxilary Loss

We learn the network parameters using training images annotated at the pixel-level. The final output is produced by performing a softmax operation on the merged score maps across all the scales. We minimize the cross-entropy loss averaged over all image positions with Stochastic Gradient Descent (SGD). The network parameters are initialized from the ImageNet-pretrained VGG-16 model of [49].

In addition to the supervision introduced to the final output, we add extra supervision to the FCN for each scale [33]. The motivation behind this is that we would like to merge *discriminative* features (after pooling or attention model) for the final classifier output. As pointed out by [33], discriminative classifiers trained with discriminative features demonstrate better performance for classification tasks. Instead of adding extra supervision to the intermediate layers [6, 33, 50, 54], we inject extra supervision to the final output of DeepLab for each scale so that the features to be merged are trained to be more discriminative. Specifically, the total loss function contains $1 + S$ cross entropy loss functions (one for final output and one for each scale) with weight one for each. The ground truths are downsampled properly w.r.t. the output resolutions during training.

discriminator
↓
scale 1
Scale 0.75
Scale 0.5
↓
3개의
classification
↓
각 Scale의
feature가
더 두드러지게
만든다.

4. Experimental Evaluations

In this section, after presenting the common setting for all the experiments, we evaluate our method on three datasets, including PASCAL-Person-Part [13], PASCAL VOC 2012 [18], and a subset of MS-COCO 2014 [35].

Network architectures: Our network is based on the publicly available model, DeepLab-LargeFOV [11], which modifies VGG-16 net [49] to be FCN [38]. We employ the same settings for DeepLab-LargeFOV as [11].

⊕ 1 = Final output
(ex) $\mathcal{L}_{ic}$)
↓
4개 class
classification

Baseline: DeepLab-LargeFOV		51.91
Merging Method		w/ E-Supv
Scales = {*1, 0.5*}		
Max-Pooling	52.90	55.26
Average-Pooling	52.71	55.17
Attention	53.49	55.85
Scales = {*1, 0.75, 0.5*}		
Max-Pooling	53.02	55.78
Average-Pooling	52.56	55.72
Attention	53.12	**56.39**

Table 1. Results on PASCAL-Person-Part *validation* set. E-Supv: extra supervision.

Head	Torso	U-arms	L-arms	U-legs	L-legs	Bkg	Avg
81.47	59.06	44.15	42.50	38.28	35.62	93.65	56.39

Table 2. Per-part results on PASCAL-Person-Part *validation* set with our attention model.

Training: SGD with mini-batch is used for training. We set the mini-batch size of 30 images and initial learning rate of 0.001 (0.01 for the final classifier layer). The learning rate is multiplied by 0.1 after 2000 iterations. We use the momentum of 0.9 and weight decay of 0.0005. Fine-tuning our network on all the reported experiments takes about 21 hours on an NVIDIA Tesla K40 GPU. During training, our model takes all scaled inputs and performs training jointly. Thus, the total training time is twice that of a vanilla DeepLab-LargeFOV. The average inference time for one PASCAL image is 350 ms.

Evaluation metric: The performance is measured in terms of pixel intersection-over-union (IOU) averaged across classes [18].

Reproducibility: The proposed methods are implemented by extending Caffe framework [29]. The code and models are available at `http://liangchiehchen.com/projects/DeepLab.html`.

Experiments: To demonstrate the effectiveness of our model, we mainly experiment along three axes: (1) multi-scale inputs (from one scale to three scales with $s \in \{1, 0.75, 0.5\}$), (2) different methods (average-pooling, max-pooling, or attention model) to merge multi-scale features, and (3) training with or without extra supervision.

4.1. PASCAL-Person-Part

Dataset: We perform experiments on semantic part segmentation, annotated by [13] from the PASCAL VOC 2010 dataset. Few works [51, 52] have worked on the animal part segmentation for the dataset. On the other hand, we focus on the *person* part for the dataset, which contains more training data and large scale variation. Specifically,