Attention to Scale: Scale-aware Semantic Image Segmentation

Liang-Chieh Chen* lcchen@cs.ucla.edu
Yi Yang, Jiang Wang, Wei Xu {yangyi05, wangjiang03, wei.xu}@baidu.com
Alan L. Yuille yuille@stat.ucla.edu

Abstract

Incorporating multi-scale features to deep convolutional neural networks (DCNNs) has been a key element to achieve state-of-art performance on semantic image segmentation benchmarks. One way to extract multi-scale features is by feeding several resized input images to a shared deep network and then merge the resulting multi-scale features for pixel-wise classification. In this work, we adapt a state-of-art semantic image segmentation model with multi-scale input images. We jointly train the network and an attention model which learns to softly weight the multi-scale features, and show that it outperforms average- or max-pooling over scales. The proposed attention model allows us to diagnostically visualize the importance of features at different positions and scales. Moreover, we show that adding extra supervision to the output of DCNN for each scale is essential to achieve excellent performance when merging multi-scale features. We demonstrate the effectiveness of our model with exhaustive experiments on three challenging datasets, including PASCAL-Person-Part, PASCAL VOC 2012 and a subset of MS-COCO 2014.

1. Introduction

Semantic image segmentation, also known as image labeling or scene parsing, relates to the problem of assigning semantic labels (*e.g.*, "person" or "dog") to every pixel in the image. It is one of the challenging tasks for scene understanding in computer vision. Recently, many methods [5, 8, 24, 27, 32, 43] based on Fully Convolutional Networks (FCNs) [28] demonstrate astonishing results on several semantic segmentation benchmarks.

Looking into the literature, we found that one of the key elements of successful semantic segmentation models is the employment of multi-scale features [11, 17, 24, 28, 31, 35]. For semantic segmentation, there are two main successful types of networks to exploit multi-scale features [42]. The first type, which we refer to as *skip-net*, combines features from the intermediate layers of Deep Convolutional Neural Networks (DCNNs) [5, 17, 28, 31]. The features within the DCNN are multi-scale in nature due to the increasingly larger receptive field sizes. The second type, which we refer to as *share-net*, feeds multi-scale inputs (*i.e.*, resize the input image to several scales) to a shared network [11, 24].

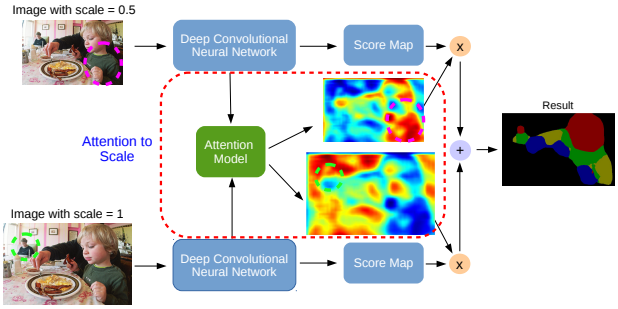


Figure 1. Model illustration. Attention model learns to put different weights on objects of different scales. For example, our model learns to put large weight on the small-scale person (green dashed circle) for features from scale = 1, and large weight on the big-scale child (magenta dashed circle) for features from scale = 0.5. We jointly train the DCNN component and the attention model.

For skip-net, a two-step training process is usually employed [5, 17, 28, 31]. That is, the deep network backbone is firstly trained and then fixed or slightly fine-tuned during multi-scale feature extraction. The problem with this strategy is that the training process is not ideal (*i.e.*, classifier training and feature-extraction are separate) and the training time is usually long (*e.g.*, three to five days [28]).

For share-net, the input image is resized to several scales and each is passed through a shared deep network. The final prediction is then based on the fusion of the resulting multi-scale features [11, 24]. Share-net does not need the two-step training process mentioned above. Average- or max-pooling over scales are usually employed [7, 8, 12, 34]. In this work, we propose to generalize average- and max-pooling. Our method not only yields better performance over baselines but also allows us to visualize which feature at which scale contributes to the classification most.

In particular, we employ an attention model [3] to generalize average- or max-pooling over scales, as shown in Fig. 1. The proposed attention model learns to weight the multi-scale features according to the object scales presented in the image (*e.g.*, the model learns to put large weights

*Work done in part during an internship at Baidu USA.

on features at coarse scale for large objects). In the experiments, we explore a state-of-art semantic segmentation model [5]. We adapt it to be a type of share-net and incorporate an attention model to it. The attention model as well as the DCNN component is jointly trained. For each scale, the attention model outputs a *weight map* which weights the accordingly features pixel-by-pixel, and by which we are able to visualize the importance of features at different positions and different scales. The weighted sum of DCNN-produced score maps from each scale is then used for classification.

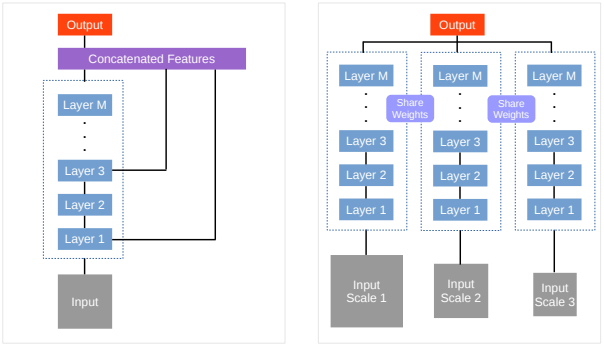
Motivated by [4, 23, 39, 42], we further introduce extra supervision to the output of DCNN for each scale. We find that introducing extra supervision is essential for our model to attain better performance. We demonstrate the effectiveness of our model on challenging datasets, including PASCAL-Person-Part [6], PASCAL VOC 2012 [10], and a subset of MS-COCO 2014 [25]. The experimental results show that our proposed methods consistently improve over strong baselines. Moreover, we demonstrate that our model generalizes well to other dataset by applying our model trained on PASCAL-Person-Part to some videos from MPII Human Pose dataset [1].

2. Related Works

Our model draws on the success of several areas, including deep networks, multi-scale features for semantic segmentation, and attention models.

Deep networks: Deep Convolutional Neural Networks (DCNNs) [22] have demonstrated state-of-art performance on several computer vision tasks, including image classification [21, 37, 38, 39, 34] and object detection [14, 18]. For the semantic image segmentation task, state-of-art methods are basically variants of the fully convolutional neural networks [28], including [5, 8, 24, 32, 43]. In particular, our method builds upon the current state-of-the-art DeepLab model [5].

Multi-scale features: It is known that multi-scale features are useful for computer vision tasks, *e.g.*, [2, 13]. In the context of deep networks for semantic segmentation, we mainly discuss two types of networks that exploit multi-scale features. The first type, *skip-net*, exploits features from different levels of the network. For example, FCN-8s [28] gradually learns finer-scale prediction from lower layers (initialized with coarser-scale prediction). Hariharan *et al*. [17] classified a pixel with hypercolumn representation (*i.e*., concatenation of features from intermediate layers). Mostajabi *et al*. [31] classified a superpixel with features extracted at zoom-out spatial levels from a small proximal neighborhood to whole image region. DeepLab-MSc (DeepLab with Multi-Scale features) [5] applied Multi-Layer Perceptrons (MLPs) to the input image and to the outputs of pooling layers, in order to extract multi-scale features. ParseNet [26] aggregated features over the whole image to provide global contextual information.



(a) skip-net (b) share-net

Figure 2. Differnet network structures for extracting multi-scale features: (a) Skip-net: features from intermediate layers are fused to produce the final output. (b) Share-net: multi-scale inputs are applied to a shared network for prediction. In this work, we demonstrate the effectiveness of the share-net when combined with attention mechanisms over scales.

The second type, *share-net*, applies multi-scale input images to a shared network. For example, Farabet *et al*. [11] employed a Laplacian pyramid, passed each scale through a shared network, and fused the features from all the scales. Lin *et al*. [24] resized the input image for three scales and concatenated the resulting three-scale features to generate the unary or pairwise potential of a Conditional Random Field (CRF). Pinheiro *et al*. [35], instead of applying multi-scale input images at once, fed multi-scale images at different stages of recurrent convolutional neural network. This share-net strategy has also been employed at the test stage for better performance by Dai *et al*. [8]. In this work, we extend DeepLab [5] to be a type of *share-net* and demonstrate its effectiveness on three challenging datasets. Note that Eigen and Fergus [9] applied input image to DCNNs at three scales from coarse to fine squentially. The DCNNs at different scales have different structures, and a two-step training process is required for their model.

Attention models for deep networks: Mnih *et al*. [30] learn an attention model, which adaptively selects image locations for processing. However, their attention model is not differentiable, which is necessary for standard back-propagation during training. On the other hand, Gregor *et al*. [15] employ a differentiable attention model to specify where to read/write image regions for image generation. For machine translation, Bahdanau *et al*. [3] propose an attention model that softly weights the importance of words in a source sentence when predicting a target word.

Attention to scale: To merge the predictions from multi-scale features, there are two common ways: average-pooling [7, 8] or max-pooling [12, 34] over scales. Motivated by [3], 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 the performance over average- and max-pooling, but also allows us to diagnostically *visualize* the importance of features at different positions and scales, which separates us from existing works on exploiting multi-scale features for semantic segmentation.

3. Model

In this section, we first review the publicly available model, DeepLab, which we build upon with proposed methods. After that, we introduce the attention model, which weights features at different scales, and then how we further improve the performance by adding extra supervision.

3.1. Review of DeepLab

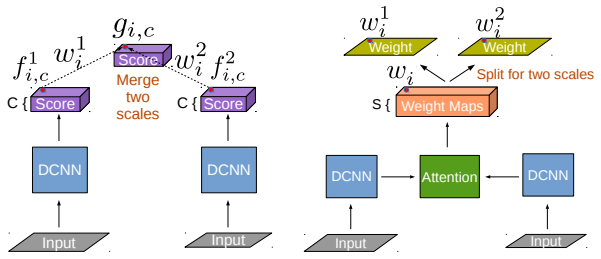
DCNNs have proven successful in semantic image segmentation [8, 27, 43]. In this subsection, we briefly review the DeepLab model [5], which is a variant of FCNs [28].

DeepLab adopts the 16-layer architecture of state-of-art classification network of [38] (*i.e.*, VGG-16 net). The network is modified to be fully convolutional [28], producing dense feature maps. In particular, the last fully-connected layers of 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 and each with stride 2. DeepLab reduces it to 8 by using the à trous (with holes) algorithm [29], 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 [5]. 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 convolutionalized 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.

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 [7, 8] or max pooling [12, 34] 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 DCNN weights are shared for 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



(a) (b)

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

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.*,

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

The weight w^s_i is computed by

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

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 the VGG-16 [38], and it consists of two layers (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.

The weight w^s_i reflects the importance of feature at position i and scale s. As a result, the attention model decides how much attention to pay to for features at different positions and different scales. It further enables us to visualize the attention for each scale by visualizing w^s_i. Note in our formulation, average-pooling or max-pooling over scales are two special cases. In particular, the weights w^s_i in Eq. (1) will be replaced by $1/S$ for average-pooling, while the summation in Eq. (1) becomes the $\max$ operation and $w^s_i = 1\ \forall s$ 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 [3]. Therefore, we are able to jointly train the attention model as well as the DCNN (*i.e.*, DeepLab) part end-to-end.

3.3. Extra supervision

We learn the network parameters using training images annotated at the pixel-level. The final output is produced by performing 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 [38].

In addition to the supervision introduced to the final output, we also add extra supervision to the DCNN for each scale, similar to [4, 23, 39, 42]. 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 [23], discriminative classifiers trained with discriminative features demonstrate better performance for classification tasks. Instead of adding extra supervision to the intermediate layers [4, 23, 39, 42], 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. In the experimental results, we show that adding extra supervision is essential for merging multi-scale inputs for our proposed methods.

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 [6], PASCAL VOC 2012 [10], and a subset of MS-COCO 2014 [25].

Network architectures: Our network is based on the publicly available model, DeepLab-LargeFOV [5], which modifies VGG-16 net [38] to be FCN [28]. We employ the same settings for DeepLab-LargeFOV as [5].

Training: SGD with mini-batch is used for training. We set 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 momentum of 0.9 and weight decay of 0.0005. Fine-tuning our network on all the reported experiments takes about 21 hours on a NVIDIA Tesla K40 GPU. During training our model takes all scaled inputs and performs training jointly. Thus, the total training time is twice more than training vanilla DeepLab-LargeFOV. The average inference time for one PASCAL image is 350 ms/image.

Evaluation metric: The performance is measured in terms of pixel intersection-over-union (IOU) averaged across classes [10].

Reproducibility: The proposed methods are implemented by extending the Caffe framework [19]. Upon acceptance, we plan to release our source code and trained models, to allow reproducing all results in the paper.

Experiments: To demonstrate the effectiveness of our proposed model, we mainly experiment along with 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) adding extra supervision or not.

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.

4.1. PASCAL-Person-Part

Dataset: We perform experiments on semantic part segmentation, annotated by [6] from PASCAL VOC 2010 dataset. Few works [40, 41] 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, the dataset contains detailed part annotations for every person, including eyes, nose, *etc*. We merge the annotations to be Head, Torso, Upper/Lower Arms and Upper/Lower Legs, resulting in six person part classes and one background class. We only use those images containing persons for training (1716 images) and validation (1817 images).

Improve DeepLab: We report the results in Tab. 1 when employing DeepLab-LargeFOV as baseline. We find that using two input scales improves over using only one input scale, and it is also slightly better than using three input scales combined with average-pooling or attention model. We hypothesize that when merging three scale inputs, the features to be merged must be sufficiently discrimintive and direct fusion of them degrades the performance. On the other hand, max-pooling seems robust to this effect. No matter how many scales are used, our proposed attention model leads to a better strategy to merge the multi-scale features than average-pooling or max-pooling. We further visualize the weight maps produced by max-pooling and attention model in Fig. 4, which clearly shows that our attention model learns better interpretable weight maps for different scales. Moreover, we find that by introducing extra supervision to the DCNN for each scale significantly improves the performance (see the column *w/ E-Supv*) over the case where extra supervision is not added, regardless of what merging scheme is employed. The results show

that adding extra supervision is essential for merging multi-scale features, which experimentally proves our hypothesis. Finally, we compare our proposed method with DeepLab-MSc-LargeFOV, which exploits the features from the intermediate layers for classification (MSc denotes Multi-Scale features). Note DeepLab-MSc-LargeFOV is a type of *skip-net*. Our best model (56.39%) attains 2.67% better performance than DeepLab-MSc-LargeFOV (53.72%).

Design choices: For all the experiments reported in this work, our proposed attention model takes as input the convolutionalized fc_7 features [38], and employs a FCN consisting of two layers (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 have experimented with different settings, including using only one layer for attention model, changing the kernel of first layer to be 1×1, and varying the number of filters for the first layer. The performance does not vary too much; the degradation ranges from 0.1% to 0.4%. Furthermore, we find that using fc_8 as features for attention model results in worse performance (drops $\sim 0.5\%$), while using fc_6 and fc_7 yield similar performance. We also try to add one more scale (four scales totally: $s \in \{1, 0.75, 0.5, 0.25\}$), however, the performance drops by 0.5%. We think the scale $\{0.25\}$ produces too small score maps after VGG-16 net.

Qualitative results: We visualize the part segmentation results as well as the weight maps produced by the attention model in Fig. 5. Merging the multi-scale features with attention model yields not only better performance quantitatively but also better interpretable weight maps. Specifically, scale-1 attention (*i.e.*, the weight map learend by attention model for scale $s = 1$) usually focuses on small-scale objects, scale-0.75 attention concentrates on middle-scale objects, and scale-0.5 attention usually puts large weight on large-scale objects or background, since it is easier to capture the largest scale objects or background contextual information when the image is shrinked to be half of the original resolution.

Failure modes: We show two failure examples in the bottom of Fig. 5. The failure examples are due to the extremely difficult human poses or the confusion between cloth and person parts. The first problem may be resolved by acquiring more data, while the second one is challenging because person parts are usually covered by clothes.

Test on unseen dataset: We further apply our trained model to some videos from MPII Human Pose dataset [1] (the video can be downloaded from the first author's website *). The model is not fine-tuned on the dataset, and the result is run frame-by-frame. As shown in the video, even for images from other dataset, our model is able to produce reasonably and visually good part segmentation results and it infers meaningful attention for different scales.

*http://web.cs.ucla.edu/~lcchen/

Baseline: DeepLab-LargeFOV		62.28
Merging Method		w/ E-Supv
Scales = {*1, 0.5*}		
Max-Pooling	64.81	67.43
Average-Pooling	64.86	67.79
Attention	65.27	68.24
Scales = {*1, 0.75, 0.5*}		
Max-Pooling	65.15	67.79
Average-Pooling	63.92	67.98
Attention	64.37	**69.08**

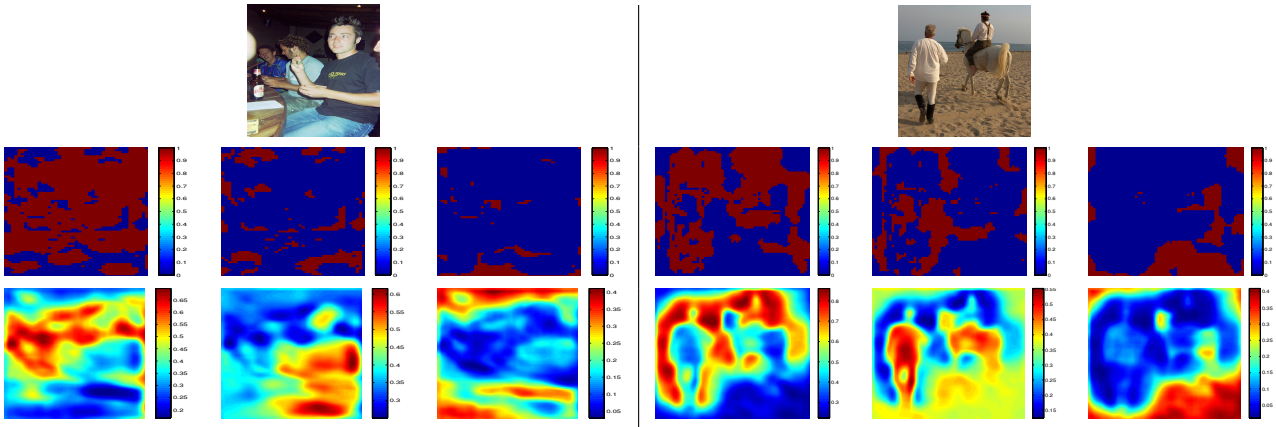
Table 2. Results on PASCAL VOC 2012 *validation* set, pretrained with ImageNet. E-Supv: extra supervision.

4.2. PASCAL VOC 2012

Dataset: PASCAL VOC 2012 segmentation benchmark [10] consists of 20 foreground object classes and one background class. Following the same experimental protocol [5, 8, 43], we augment the original training set from the annotations by [16]. We report the results on the original PASCAL VOC 2012 validation set and test set.

Pretrained with ImageNet: First, we experiment with the scenario where the underlying DeepLab-LargeFOV is only pretrained on ImageNet [36]. Our reproduction of DeepLab-LargeFOV and DeepLab-MSc-LargeFOV yields performance of 62.28% and 64.39% on validation set, respectively. They are similar to those (62.25% and 64.21%) reported in [5]. We report the results of proposed methods on validation set in Tab. 2. We observe similar experimental results as PASCAL-Person-Part dataset: (1) Using two input scales is better than single input scale. (2) Adding extra supervision is necessary to achieve better performance for merging three input scales, especially for average-pooling and proposed attention model. (3) The best performance (6.8% improvement over the DeepLab-LargeFOV baseline) is obtained with three input scales, attention model, and extra supervision, and its performance is 4.69% better than DeepLab-MSc-LargeFOV (64.39%).

We also report the test set result with our best model on validation set in the top of Tab. 3. We first observe that employing proposed attention model yields 1% performance better than employing average-pooling, consistent to our results on validation set. We then compare our models with DeepLab-LargeFOV and DeepLab-MSc-LargeFOV [5]. We find that our proposed model improves 6.4% over DeepLab-LargeFOV, and yields 4.5% better performance than DeepLab-MSc-LargeFOV. Finally, we compare our models with two other methods: ParseNet [26] and TTI_zoomout_v2 [31]. ParseNet incorporates the image-level feature as global contextual information. We consider ParseNet as a special case to exploit multi-scale features,



(a) Scale-1 Attention (b) Scale-0.75 Attention (c) Scale-0.5 Attention (a) Scale-1 Attention (b) Scale-0.75 Attention (c) Scale-0.5 Attention

Figure 4. Weight maps produced by max-pooling (row 2) and by attention model (row 3). Notice that our attention model learns better interpretable weight maps for different scales. (a) Scale-1 attention (*i.e.*, weight map for scale $s = 1$) captures small-scale objects, (b) Scale-0.75 attention usually focuses on middle-scale objects, and (c) Scale-0.5 attention emphasizes on background contextual information.

Method	mean	bkg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Pretrained with ImageNet																						
DeepLab-LargeFOV [5]	65.1	90.7	74.7	34.0	74.3	57.1	62.0	82.6	75.5	79.1	26.2	65.7	55.8	73.0	68.0	78.6	76.2	50.6	73.9	45.5	66.6	57.1
DeepLab-MSc-LargeFOV [5]	67.0	91.6	78.7	51.4	75.7	59.5	61.7	82.4	76.7	79.4	26.8	67.7	54.7	74.3	70.0	79.9	77.3	52.5	75.5	46.5	67.0	57.1
TTI_zoomout_v2 [31]	69.6	91.9	85.6	37.3	83.2	62.5	66.0	85.1	80.7	84.9	27.2	73.3	57.5	78.1	79.2	81.1	77.1	53.6	74.0	49.2	71.7	63.3
ParseNet [26]	69.8	92.4	84.1	37.0	77.0	62.8	64.0	85.8	79.7	83.7	27.7	74.8	57.6	77.1	78.3	81.0	78.2	52.6	80.4	49.9	75.7	65.0
DeepLab-LargeFOV-**AveragePooling**	70.5	92.7	83.5	37.2	75.4	60.9	69.3	89.0	83.4	83.5	28.2	73.4	58.7	78.4	79.0	83.0	79.7	54.4	79.6	50.2	78.0	63.5
DeepLab-LargeFOV-**Attention**	71.5	92.9	86.0	38.8	78.2	63.1	70.2	89.6	84.1	82.9	29.4	75.2	58.7	79.3	78.4	83.9	80.3	53.5	82.6	51.5	79.2	64.2
Pretrained with MS-COCO																						
DeepLab-CRF-COCO-LargeFOV [33]	72.7	93.4	89.1	38.3	88.1	63.3	69.7	87.1	83.1	85.0	29.3	76.5	56.5	79.8	77.9	85.8	82.4	57.4	84.3	54.9	80.5	64.1
DeepLab-MSc-CRF-COCO-LargeFOV [33]	73.6	93.8	88.7	53.1	87.7	64.4	69.5	85.9	81.6	85.3	31.0	76.4	62.0	79.8	77.3	84.6	83.2	59.1	85.5	55.9	76.5	64.3
DeepLab-CRF-COCO-LargeFOV-**Attention**	75.1	94.0	92.0	41.2	87.8	57.2	72.7	92.8	85.9	90.5	30.5	78.0	62.8	85.8	85.3	87.2	85.6	57.7	85.1	56.5	83.0	65.0

Table 3. Labeling IOU on the PASCAL VOC 2012 test set, using the trainval set for training.

Baseline: DeepLab-LargeFOV	67.58	
Merging Method		w/ E-Supv
Scales = {*1, 0.5*}		
Max-Pooling	69.15	70.01
Average-Pooling	69.22	70.44
Attention	69.90	70.76
Scales = {*1, 0.75, 0.5*}		
Max-Pooling	69.70	70.06
Average-Pooling	68.82	70.55
Attention	69.47	**71.42**

Table 4. Results on PASCAL VOC 2012 *validation* set, pretrained with MS-COCO. E-Supv: extra supervision.

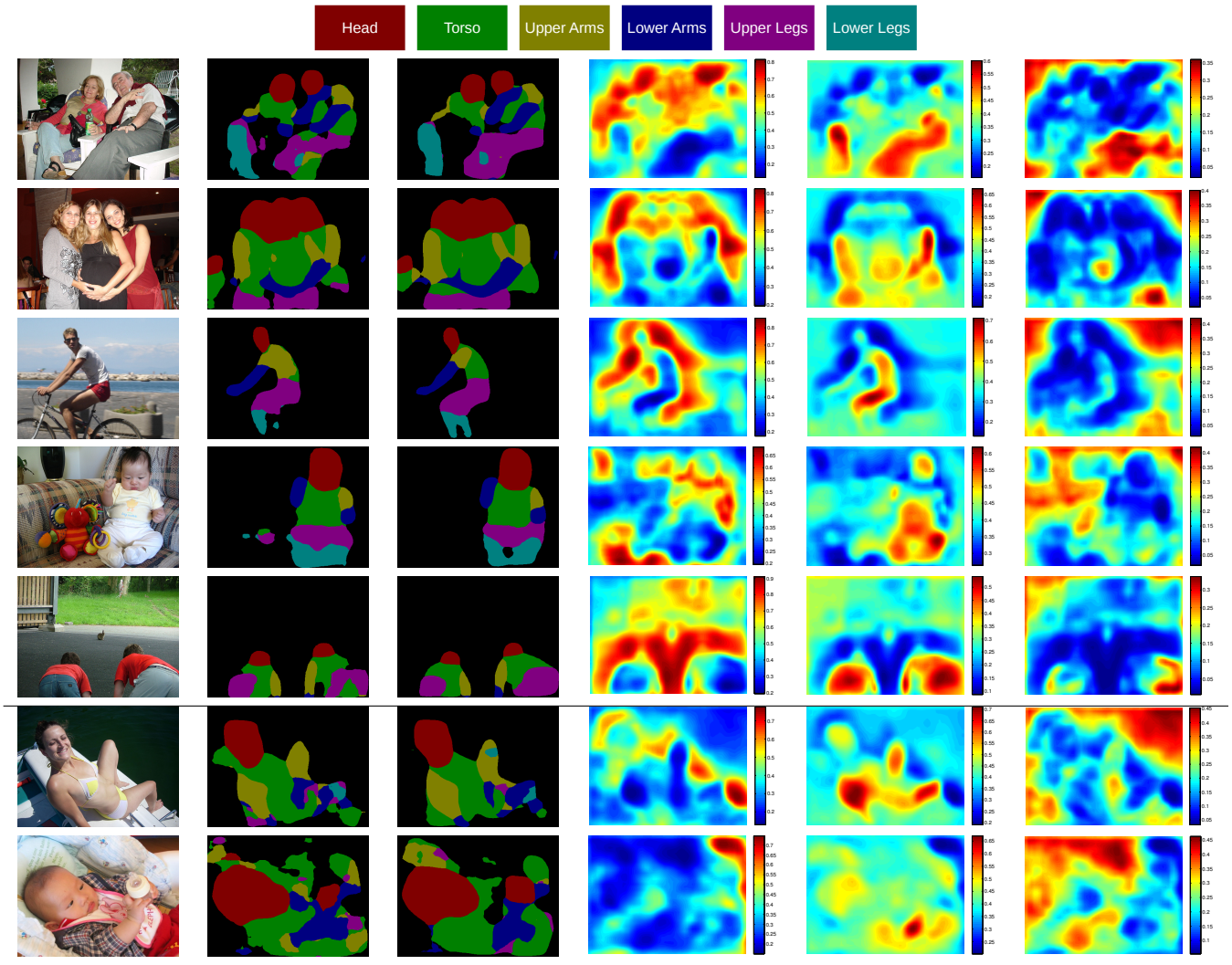
where the whole image is summarized by the image-level feature. TTI_zoomout_v2 also exploits features at different spatial scales. As shown in the table, our proposed model outperforms both of them. Note none of the methods discussed here employs a fully connected CRF [20].

Pretrained with MS-COCO: Second, we experiment with the scenario where the underlying baseline, DeepLab-LargeFOV, has been pretrained on MS-COCO 2014 dataset [25]. The goal is to test if we can still observe any improvement with such a strong baseline. As shown in Tab. 4, we again observe the similar experimental results as before, and our best model still outperforms the DeepLab-LargeFOV baseline by 3.84%. We also report the both best models on the *test* set in the bottom of Tab. 3. For fair comparison with the reported DeepLab variants on test set, we also employ a fully connected CRF [20] as post processing. As shown in the table, our model attains the performance of 75.1%, outperforming DeepLab-CRF-LargeFOV and DeepLab-MSc-CRF-LaregeFOV by 2.4%, and 1.5%, respectively.

Note our models do not outperform current best models [24, 27], which employ joint training of CRF (with *spatial* pairwise term) and DCNN. However, we think our proposed methods (*e.g.*, attention model for scales) could be complementary to theirs. We emphasize that our models are trained end-to-end with one pass to exploit multi-scale features, instead of multiple training steps.

4.3. Subset of MS-COCO

Dataset: The MS-COCO 2014 dataset [25] contains 80 foreground object classes and one background class. The training set has about 80K images, and 40K images for val-

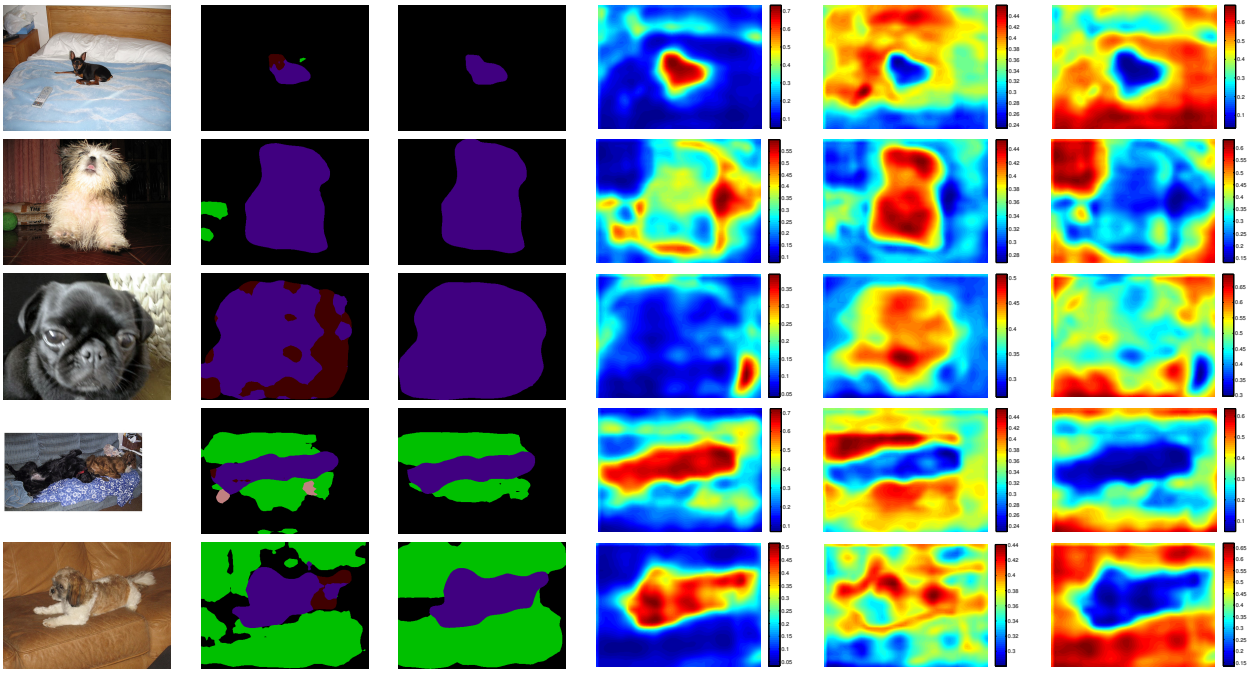


(a) Image (b) Baseline (c) Our model (d) Scale-1 Attention (e) Scale-0.75 Attention (f) Scale-0.5 Attention

Figure 5. Results on PASCAL-Person-Part *validation* set. DeepLab-LargeFOV with one scale input is used as baseline. Our model employs three scale inputs, attention model and extra supervision. Scale-1 attention captures small-scale parts, scale-0.75 attention catches middle-scale torsos and legs, while scale-0.5 attention focuses on large-scale legs and background. Bottom two rows show failure examples.

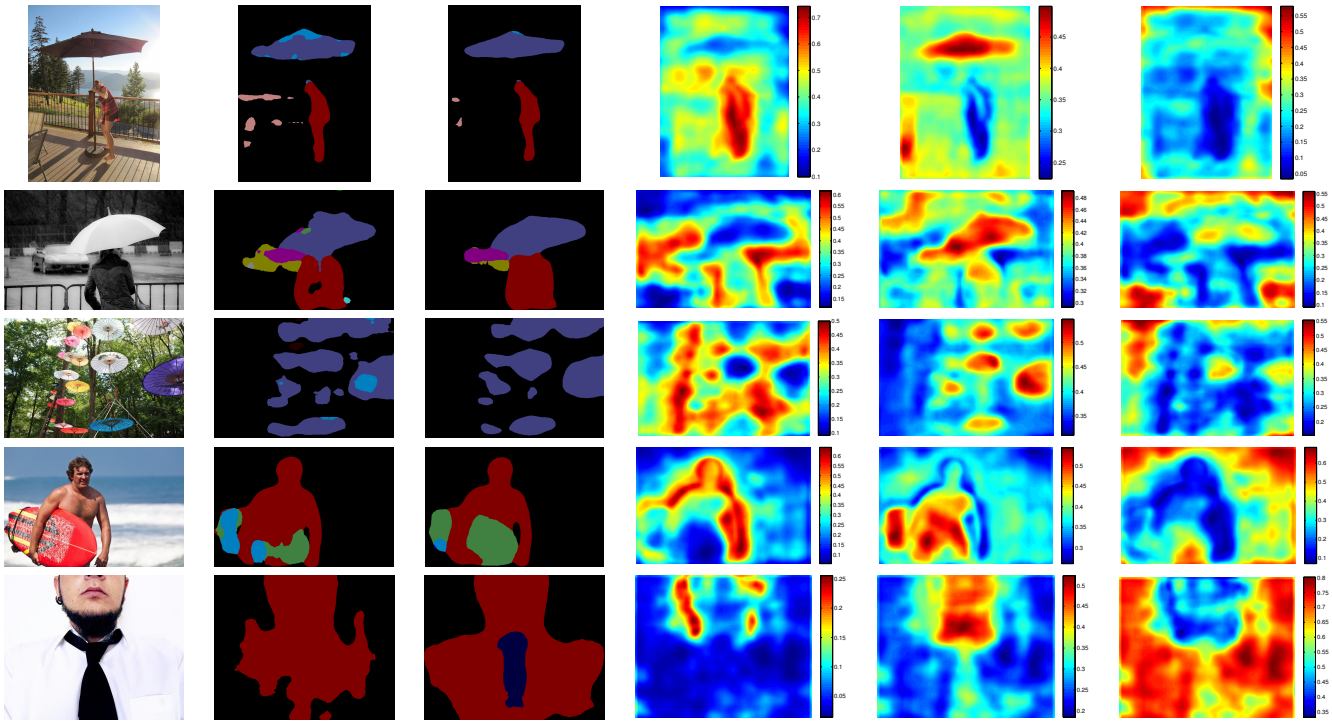
idation. We randomly select 10K images from the training set and 1,500 images from the validation set (the resulting training and validation sets have same sizes as those we used for PASCAL VOC 2012). The goal is to demonstrate our model on another challenging dataset.

Improve DeepLab: In addition to observing similar results as before, we find that the DeepLab-LargeFOV baseline achieves a low mean IOU 31.22% in Tab. 5 due to the difficulty of MS-COCO dataset (*e.g.*, large object scale variance and more object classes). However, employing multi-scale inputs, attention model, and extra supervision can still brings 4.6% improvement over the DeepLab-LargeFOV baseline, and 4.17% over DeepLab-MSc-LargeFOV (31.61%). We find that the results of employing average-pooling and attention model as merging methods are very similar. We think it is because that there are many small object classes (*e.g.*, fork, mouse, and toothbrush) with extremely low prediciton accuracy, which reduces the improvment. This challenging problem (*i.e.*, segment small obects and handle imbalanced classes) is considered as future work. On the other hand, we looked into the performance for some class. In particular, we show the performance for the *person* class in Tab. 6 because the *person* class occurs most frequently and appears with different scales (see Fig. 5(a), and Fig. 13(b) in [25]) in this dataset. As shown in the table, the improvement from the proposed methods becomes more noticeable in this case, and we observe the same results as before. The qualitative results are shown in Fig. 7.



(a) Image (b) Baseline (c) Our model (d) Scale-1 Attention (e) Scale-0.75 Attention (f) Scale-0.5 Attention

Figure 6. Results on PASCAL VOC 2012 *validation* set. DeepLab-LargeFOV with one scale input is used as baseline. Our model employs three scale inputs, attention model and extra supervision. Scale-1 attention captures small-scale dogs (dark blue label), scale-0.75 attention concentrates on middle-scale dogs and part of sofa (light green label), while scale-0.5 attention catches largest-scale dogs and sofa.



(a) Image (b) Baseline (c) Our model (d) Scale-1 Attention (e) Scale-0.75 Attention (f) Scale-0.5 Attention

Figure 7. Results on subset of MS-COCO 2014 *validation* set. DeepLab-LargeFOV with one scale input is used as baseline. Our model employs three scale inputs, attention model and extra supervision. Scale-1 attention captures small-scale person (dark red label) and umbrella (violet label). Scale-0.75 attention concentrates on middle-scale umbrella and head, while scale-0.5 attention catches large-scale person torso.

Baseline: DeepLab-LargeFOV		31.22
Merging Method		w/ E-Supv
Scales = {*1, 0.5*}		
Max-Pooling	32.95	34.70
Average-Pooling	33.69	35.14
Attention	34.03	35.41
Scales = {*1, 0.75, 0.5*}		
Max-Pooling	33.58	35.08
Average-Pooling	33.74	35.72
Attention	33.42	**35.78**

Table 5. Results on subset of MS-COCO *validation* set with DeepLab-LargeFOV as baseline. E-Supv: extra supervision.

Baseline: DeepLab-LargeFOV		68.76
Merging Method		w/ E-Supv
Scales = {*1, 0.5*}		
Max-Pooling	70.07	71.06
Average-Pooling	70.38	71.60
Attention	70.66	72.20
Scales = {*1, 0.75, 0.5*}		
Max-Pooling	69.97	71.43
Average-Pooling	69.69	71.70
Attention	70.14	**72.72**

Table 6. **Person** class IOU on subset of MS-COCO *validation* set with DeepLab-LargeFOV as baseline. E-Supv: extra supervision.

5. Conclusion

For semantic segmentation, this paper has adapted a state-of-art model (*i.e.*, DeepLab-LargeFOV) to exploit multi-scale inputs. Experiments on three datasets have shown that: (1) Using multi-scale inputs yields better performance than single scale input. (2) Merging the multi-scale features with the proposed attention model not only improves the performance over average- or max-pooling baselines, but also allows us to diagnostically visualize the importance of features at different positions and scales. (3) Excellent performance can be obtained by adding extra supervision to the final output of networks for each scale.

Supplementary Material

We include as appendix: (1) more qualitative results on PASCAL-Person-Part, PASCAL VOC 2012, and subset of MS-COCO 2014 datasets.

A. More qualitative results

We show more qualitative results on PASCAL-Person-Part [6] in Fig. 8, on PASCAL VOC 2012 [10] in Fig. 9, and on subset of MS-COCO 2014 [25] in Fig. 10.

References

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 2, 5

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*. 2

[3] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. 1, 2, 3

[4] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al. Greedy layer-wise training of deep networks. In *NIPS*, 2007. 2, 4

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 1, 2, 3, 4, 5, 6

[6] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, 2014. 2, 4, 9

[7] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*, 2012. 1, 2, 3

[8] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015. 1, 2, 3, 5

[9] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 2

[10] M. Everingham, S. A. Eslami, L. V. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2014. 2, 4, 5, 9

[11] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *PAMI*, 35(8):1915–1929, 2013. 1, 2

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010. 1, 2, 3

[13] L. Florack, B. T. H. Romeny, M. Viergever, and J. Koenderink. The gaussian scale-space paradigm and the multi-scale local jet. *IJCV*, 18(1):61–75, 1996. 2

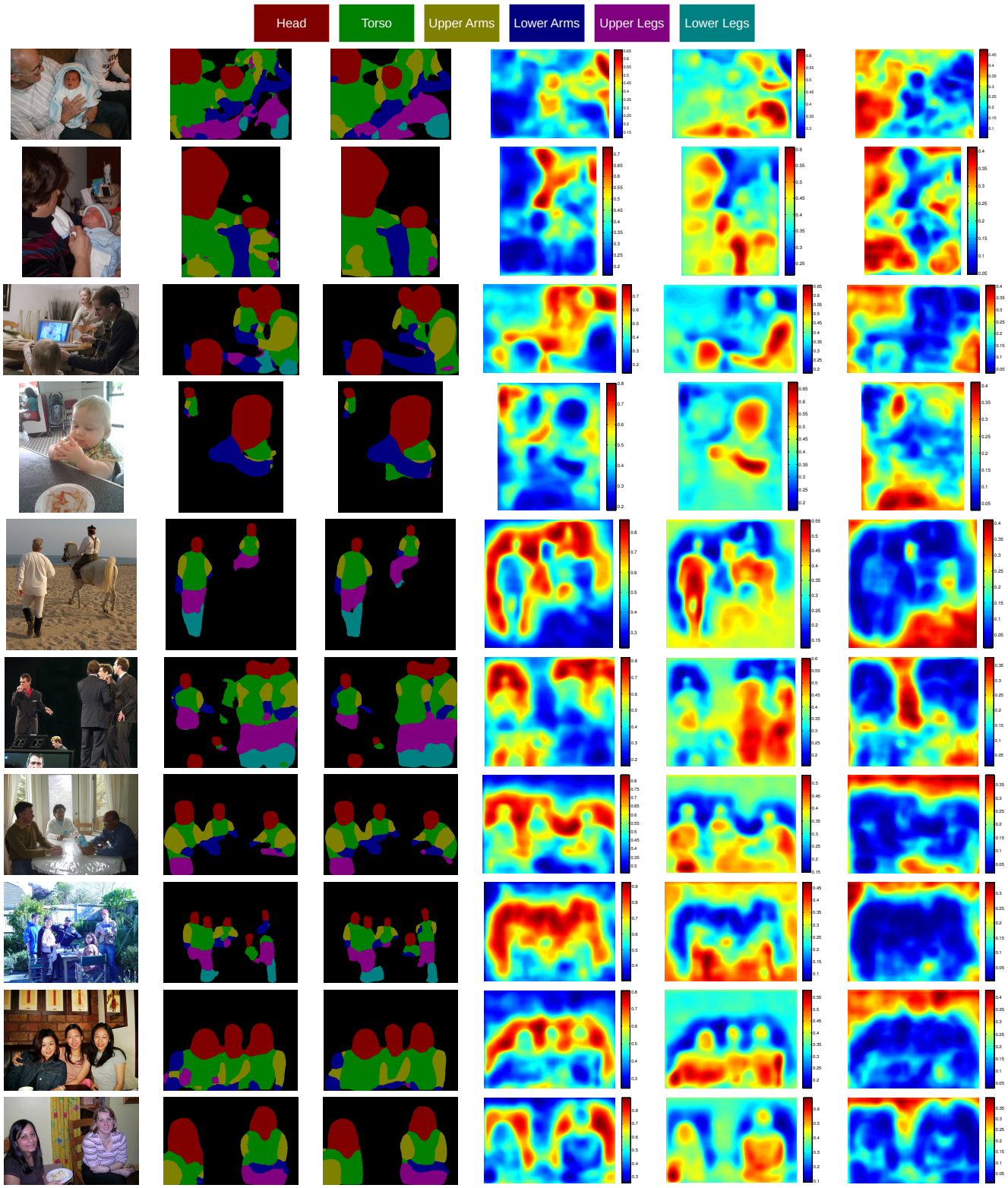
[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[15] K. Gregor, I. Danihelka, A. Graves, and D. Wierstra. Draw: A recurrent neural network for image generation. In *ICML*, 2015. 2

[16] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 5

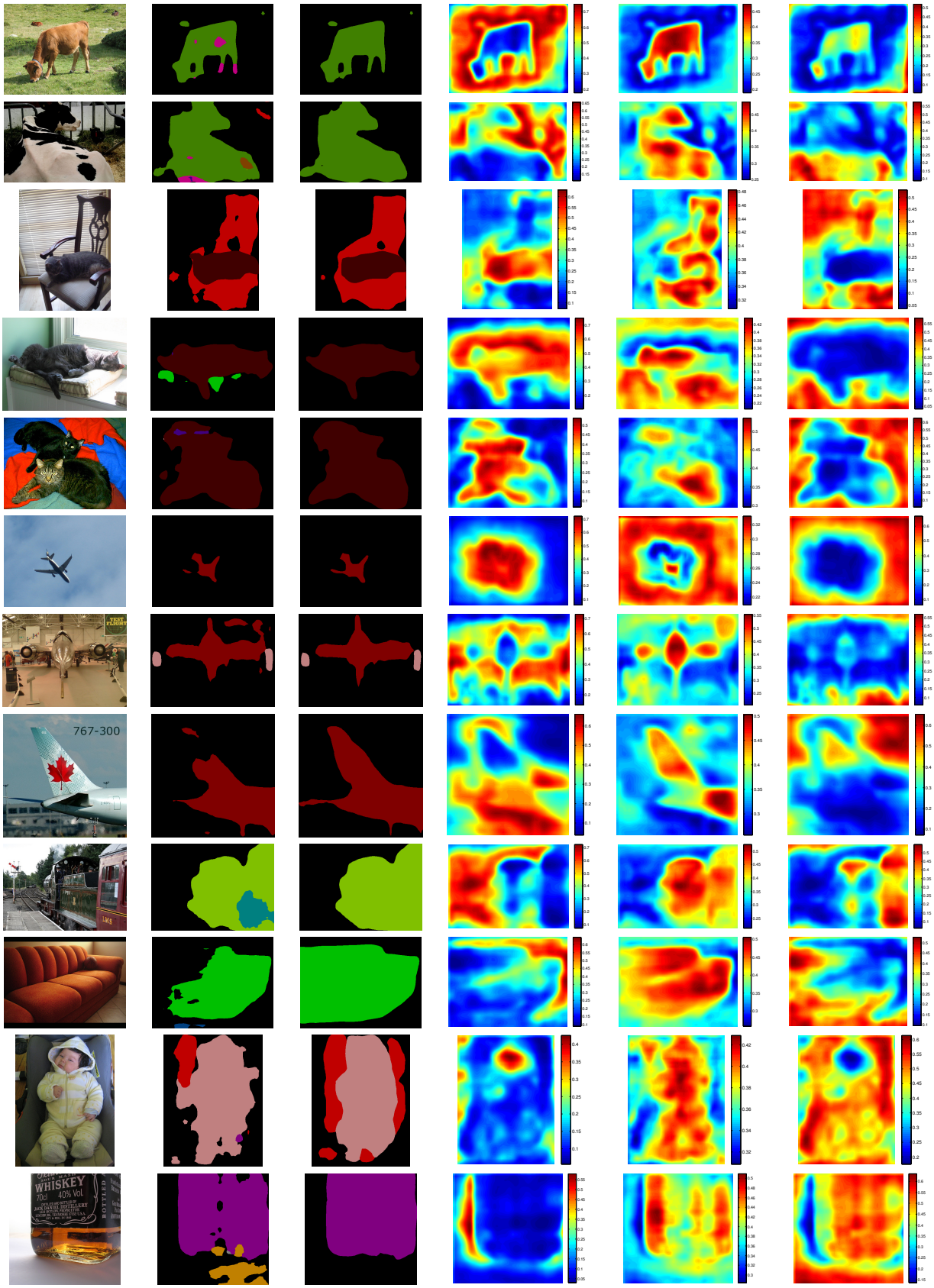
[17] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 1, 2

[18] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*. 2014. 2



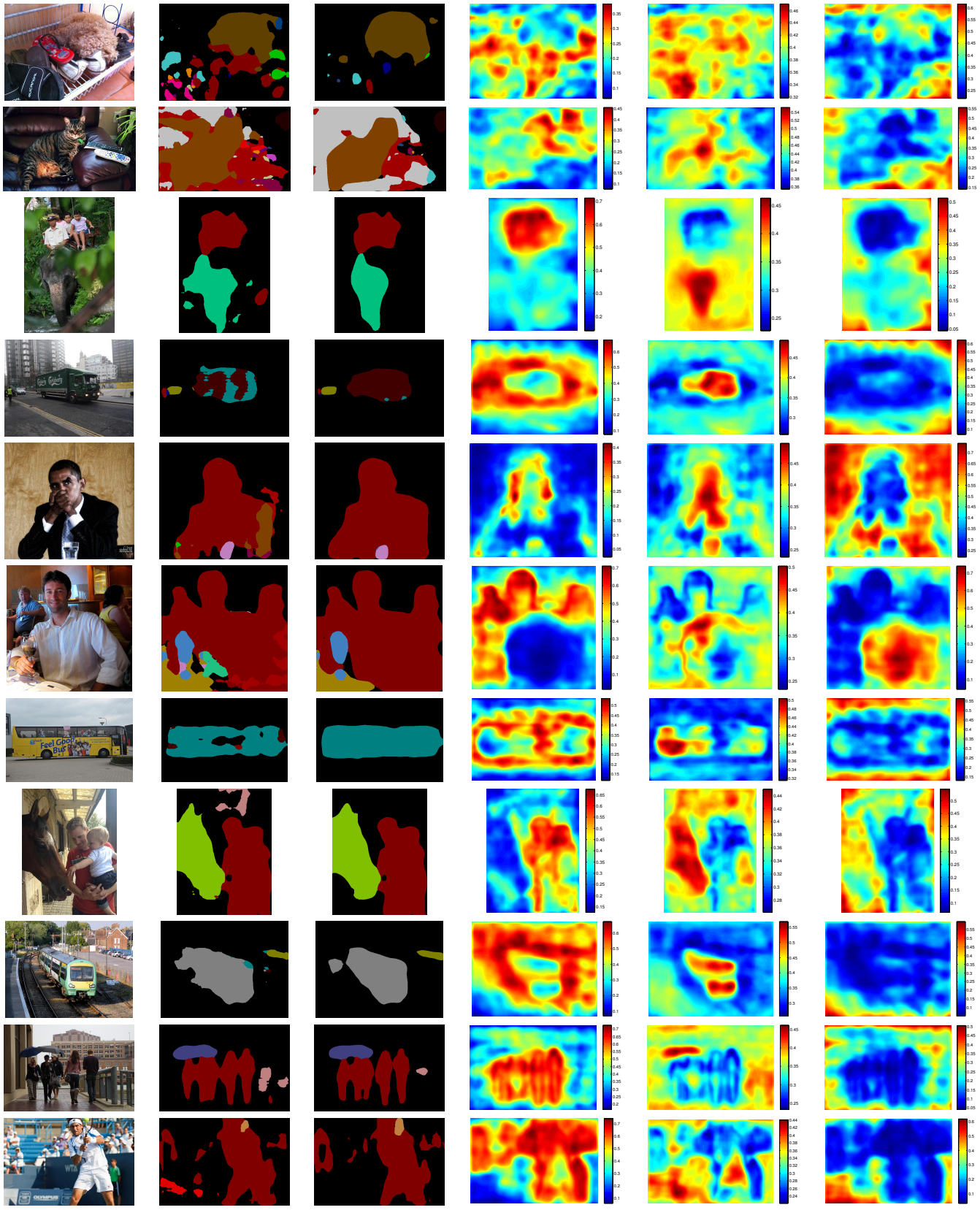
(a) Image (b) Baseline (c) Our model (d) Scale-1 Attention (e) Scale-0.75 Attention (f) Scale-0.5 Attention

Figure 8. Qualitative segmentation results on PASCAL-Person-Part validation set.



(a) Image (b) Baseline (c) Our model (d) Scale-1 Attention (e) Scale-0.75 Attention (f) Scale-0.5 Attention

Figure 9. Qualitative segmentation results on PASCAL VOC 2012 validation set.



(a) Image (b) Baseline (c) Our model (d) Scale-1 Attention (e) Scale-0.75 Attention (f) Scale-0.5 Attention

Figure 10. Qualitative segmentation results on subset of MS-COCO 2014 validation set.

[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014. 4

[20] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011. 6

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[22] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[23] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *AISTATS*, 2015. 2, 4

[24] G. Lin, C. Shen, I. Reid, et al. Efficient piecewise training of deep structured models for semantic segmentation. *arXiv:1504.01013*, 2015. 1, 2, 6

[25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 4, 6, 7, 9

[26] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. In *ICCV*, 2015. 2, 5, 6

[27] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *ICCV*, 2015. 1, 3, 6

[28] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 2, 3, 4

[29] S. Mallat. *A Wavelet Tour of Signal Processing*. Acad. Press, 2 edition, 1999. 3

[30] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In *NIPS*, 2014. 2

[31] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *CVPR*, 2015. 1, 2, 5, 6

[32] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. *arXiv:1505.04366*, 2015. 1, 2

[33] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. In *ICCV*, 2015. 6

[34] G. Papandreou, I. Kokkinos, and P.-A. Savalle. Untangling local and global deformations in deep convolutional networks for image classification and sliding window detection. In *CVPR*, 2015. 1, 2, 3

[35] P. H. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene parsing. *arXiv:1306.2795*, 2013. 1, 2

[36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, pages 1–42, 2015. 5

[37] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014. 2

[38] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2, 3, 4, 5

[39] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv:1409.4842*, 2014. 2, 4

[40] J. Wang and A. Yuille. Semantic part segmentation using compositional model combining shape and appearance. In *CVPR*, 2015. 4

[41] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille. Joint object and part segmentation using deep learned potentials. In *ICCV*, 2015. 4

[42] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015. 1, 2, 4

[43] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 1, 2, 3, 5