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 alan.yuille@jhu.edu

Abstract

Incorporating multi-scale features in fully convolutional neural networks (FCNs) has been a key element to achieving state-of-the-art performance on semantic image segmentation. One common way to extract multi-scale features is to feed multiple resized input images to a shared deep network and then merge the resulting features for pixel-wise classification. In this work, we propose an attention mechanism that learns to softly weight the multi-scale features at each pixel location. We adapt a state-of-the-art semantic image segmentation model, which we jointly train with multi-scale input images and the attention model. The proposed attention model not only outperforms average- and max-pooling, but allows us to diagnostically visualize the importance of features at different positions and scales. Moreover, we show that adding extra supervision to the output at each scale is essential to achieving excellent performance when merging multi-scale features. We demonstrate the effectiveness of our model with extensive experiments on three challenging datasets, including PASCAL-Person-Part, PASCAL VOC 2012 and a subset of MS-COCO 2014.

一种提取多尺度特征的方式是多尺度的输入然后融合输出到分类器中，

我们提出利用注意力机制学习softly weight多尺度特征对于每个像素点

提出的注意力模型不仅表现出最大池化，也允许我们诊断可视化重要特征在不同的位置和尺度下。

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 a very challenging task in computer vision and one of the most crucial steps towards scene understanding [18]. Successful image segmentation techniques could facilitate a large group of applications such as image editing [17], augmented reality [3] and self-driving vehicles [22].

Recently, various methods [11, 15, 37, 42, 58, 34] based on *Fully Convolutional Networks* (FCNs) [38] demonstrate astonishing results on several semantic segmentation benchmarks. Among these models, one of the key elements to successful semantic segmentation is the use of multi-scale features [19, 45, 27, 38, 41, 34]. In the FCNs setting, there are mainly two types of network structures that exploit multi-scale features [54].

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

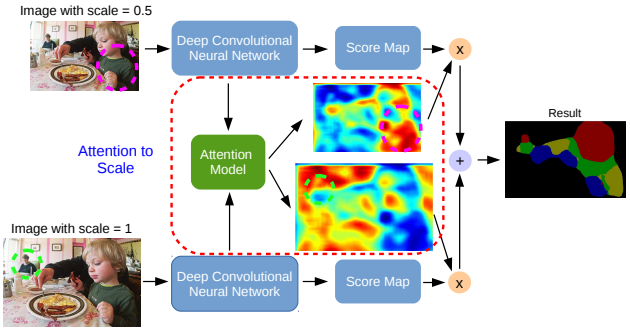


Figure 1. Model illustration. The attention model learns to put different weights on objects of different scales. For example, our model learns to put large weights on the small-scale person (green dashed circle) for features from scale = 1, and large weights on the large-scale child (magenta dashed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

The first type, which we refer to as *skip-net*, combines features from the intermediate layers of FCNs [27, 38, 41, 11]. Features within a skip-net are multi-scale in nature due to the increasingly large receptive field sizes. During training, a skip-net usually employs a two-step process [27, 38, 41, 11], where it first trains the deep network backbone and then fixes or slightly fine-tunes 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 [38]).

skip-net：组合中间层特征。这里多尺度是因为增加的感受野。如FCN最后几层的融合。

The second type, which we refer to as *share-net*, resizes the input image to several scales and passes each through a shared deep network. It then computes the final prediction based on the fusion of the resulting multi-scale features [19, 34]. A share-net does not need the two-step training process mentioned above. It usually employs average- or max-pooling over scales [20, 14, 44, 15]. Features at each scale are either equally important or sparsely selected.

share-net：输入多个尺寸，然后输出的结果融合。

Recently, attention models have shown great success in several computer vision and natural language processing

tasks [5, 40, 55, 9]. Rather than compressing an entire image or sequence into a static representation, attention allows the model to focus on the most relevant features as needed. In this work, we incorporate an attention model for semantic image segmentation. Unlike previous work that employs attention models in the 2D spatial and/or temporal dimension [48, 56], we explore its effect in the scale dimension.

In particular, we adapt a state-of-the-art semantic segmentation model [11] to a share-net and employ a soft attention model [5] to generalize average- and 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 on features at coarse scale for large objects). For each scale, the attention model outputs a *weight map* which weights features pixel by pixel, and the weighted sum of FCN-produced score maps across all scales is then used for classification.

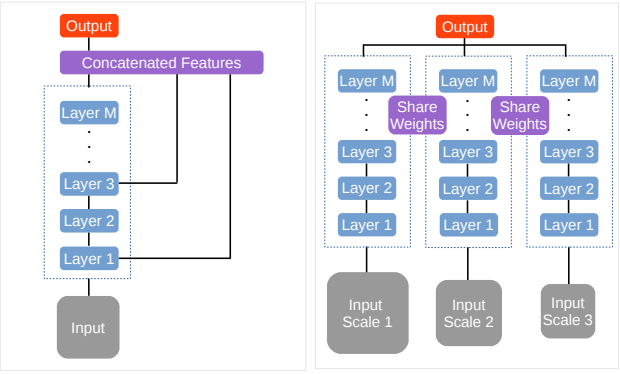
Motivated by [6, 33, 50, 54], we further introduce extra supervision to the output of FCNs at each scale, which we find essential for a better performance. We jointly train the attention model and the multi-scale networks. We demonstrate the effectiveness of our model on several challenging datasets, including PASCAL-Person-Part [13], PASCAL VOC 2012 [18], and a subset of MS-COCO 2014 [35]. Experimental results show that our proposed method consistently improves over strong baselines. The attention component also gives a non-trivial improvement over average- and max-pooling methods. More importantly, the proposed attention model provides diagnostic visualization, unveiling the black box network operation by visualizing the importance of features at each scale for every image position.

2. Related Work

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

Deep networks: Deep Convolutional Neural Networks (DCNNs) [32] have demonstrated state-of-the-art performance on several computer vision tasks, including image classification [31, 47, 50, 49, 44] and object detection [24, 28]. For the semantic image segmentation task, state-of-the-art methods are variants of the fully convolutional neural networks (FCNs) [38], including [11, 15, 34, 42, 58]. In particular, our method builds upon the current state-of-the-art DeepLab model [11].

Multi-scale features: It is known that multi-scale features are useful for computer vision tasks, *e.g.*, [21, 2]. 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 [38] gradually learns finer-scale prediction from lower layers (initialized with coarser-scale prediction). Hariharan *et al.* [27] classified a pixel with hypercolumn representation (*i.e.*, concatenation of features from intermediate layers). Mostajabi *et al.* [41] classified a superpixel with features extracted at zoom-out spatial levels from a small proximal neighborhood to the whole image region. DeepLab-MSc (DeepLab with Multi-Scale features) [11] applied Multi-Layer Perceptrons (MLPs) to the input image and to the outputs of pooling layers, in order to extract multi-scale features. ParseNet [36] aggregated features over the whole image to provide global contextual information.



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

Figure 2. Different 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.* [19] employed a Laplacian pyramid, passed each scale through a shared network, and fused the features from all the scales. Lin *et al.* [34] resized the input image for three scales and concatenated the resulting three-scale features to generate the unary and pairwise potentials of a Conditional Random Field (CRF). Pinheiro *et al.* [45], instead of applying multi-scale input images at once, fed multi-scale images at different stages in a recurrent convolutional neural network. This share-net strategy has also been employed during the test stage for a better performance by Dai *et al.* [15]. In this work, we extend DeepLab [11] to be a type of *share-net* and demonstrate its effectiveness on three challenging datasets. Note that Eigen and Fergus [16] fed input images to DCNNs at three scales from coarse to fine sequentially. The DCNNs at different scales have different structures, and a two-step training process is required for their model.

Attention models for deep networks: In computer vision, attention models have been used widely used for image classification [8, 25, 53] and object detection [4, 7, 57]. Mnih *et al.* [40] learn an attention model that adaptively se-

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.

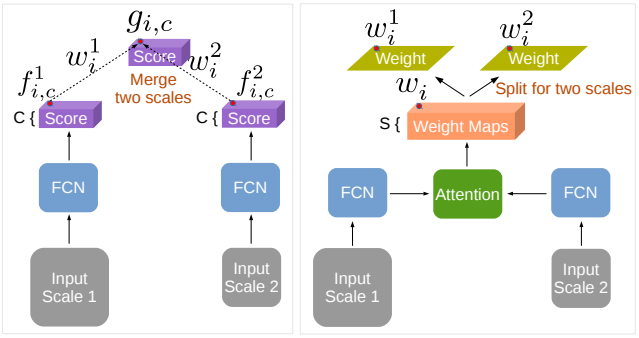
Attention to scale: To merge the predictions from multi-scale features, there are two common approachs: average-pooling [14, 15] or max-pooling [20, 44] over scales. Motivated by [5], we propose to jointly learn an attention model that softly weights the features from different input scales when predicting 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 $f_{i,c}^s$ 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 $f_{i,c}^s$ 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*.,

$$g_{i,c} = \sum_{s=1}^{S} w_i^s \cdot f_{i,c}^s \tag{1}$$

The weight w_i^s is computed by

$$w_i^s = \frac{\exp(h_i^s)}{\sum_{t=1}^{S} \exp(h_i^t)} \tag{2}$$

where h_i^s is the score map (*i.e*., last layer output before SoftMax) produced by the attention model at position i for scale s. Note w_i^s 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.

注意力模型决定了多少注意力要pay给不同位置和尺度的特征。

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.

我们强调注意力模型计算软加权给每个尺度和位置

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

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.

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].

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,

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).

Improvement over DeepLab: We report the results in Tab. 1 when employing DeepLab-LargeFOV as the 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 discriminative or direct fusion degrades performance. On the other hand, max-pooling seems robust to this effect. No matter how many scales are used, our attention model yields better results than average-pooling and max-pooling. We further visualize the weight maps produced by max-pooling and our 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 FCNs for each scale significantly improves the performance (see the column *w/ E-Supv*), regardless of what merging scheme is employed. The results show that adding extra supervision is essential for merging multi-scale features. 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 [49], and employs a FCN consisting of two layers (the first layer has 512 filters with kernel size 3×3 and the 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 the attention model, changing the kernel of the 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 the attention model results in worse performance (drops $\sim 0.5\%$) with similar results for fc_6 and fc_7. We also tried adding one more scale (four scales in total: $s \in \{1, 0.75, 0.5, 0.25\}$), however, the performance drops by 0.5%. We believe the score maps produced from scale $s = 0.25$ were simply too small to be useful.

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 the attention model yields not only better performance but also more interpretable weight maps. Specifically, scale-1 attention (*i.e*., the weight map learned 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 shrunk to be half of the original resolution.

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

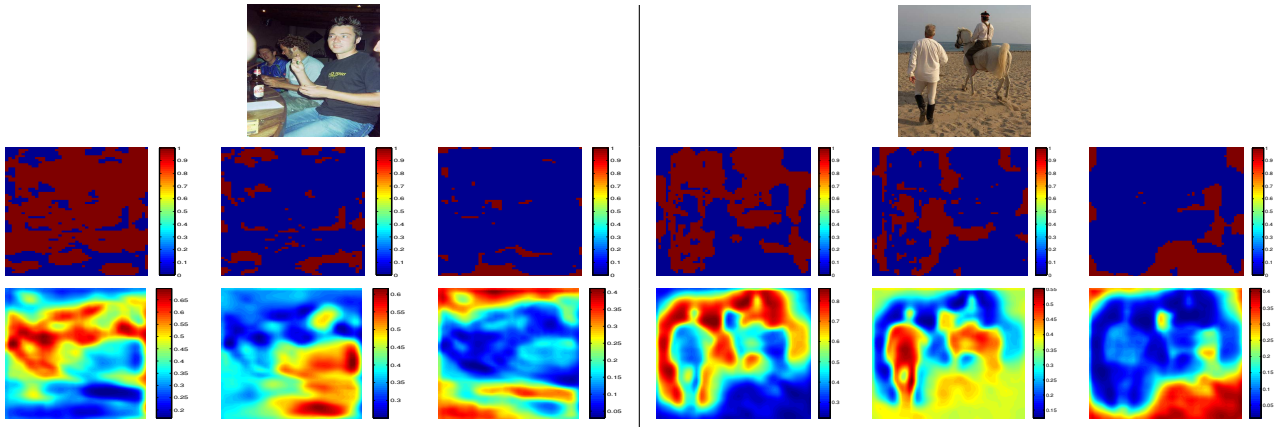
Failure modes: We show two failure examples in the bottom of Fig. 5. The failure examples are due to the extremely difficult human poses or the confusion between cloth and person parts. The first problem may be resolved by acquiring more data, while the second one is challenging because person parts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

Supplementary materials: In the supplementary materials, we apply our trained model to some videos from MPII Human Pose dataset [1]. 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 another dataset, our model is able to produce reasonably and visually good part segmentation results and it infers meaningful attention for different scales. Additionally, we provide more qualitative results for all datasets in the supplementary materials.

4.2. PASCAL VOC 2012

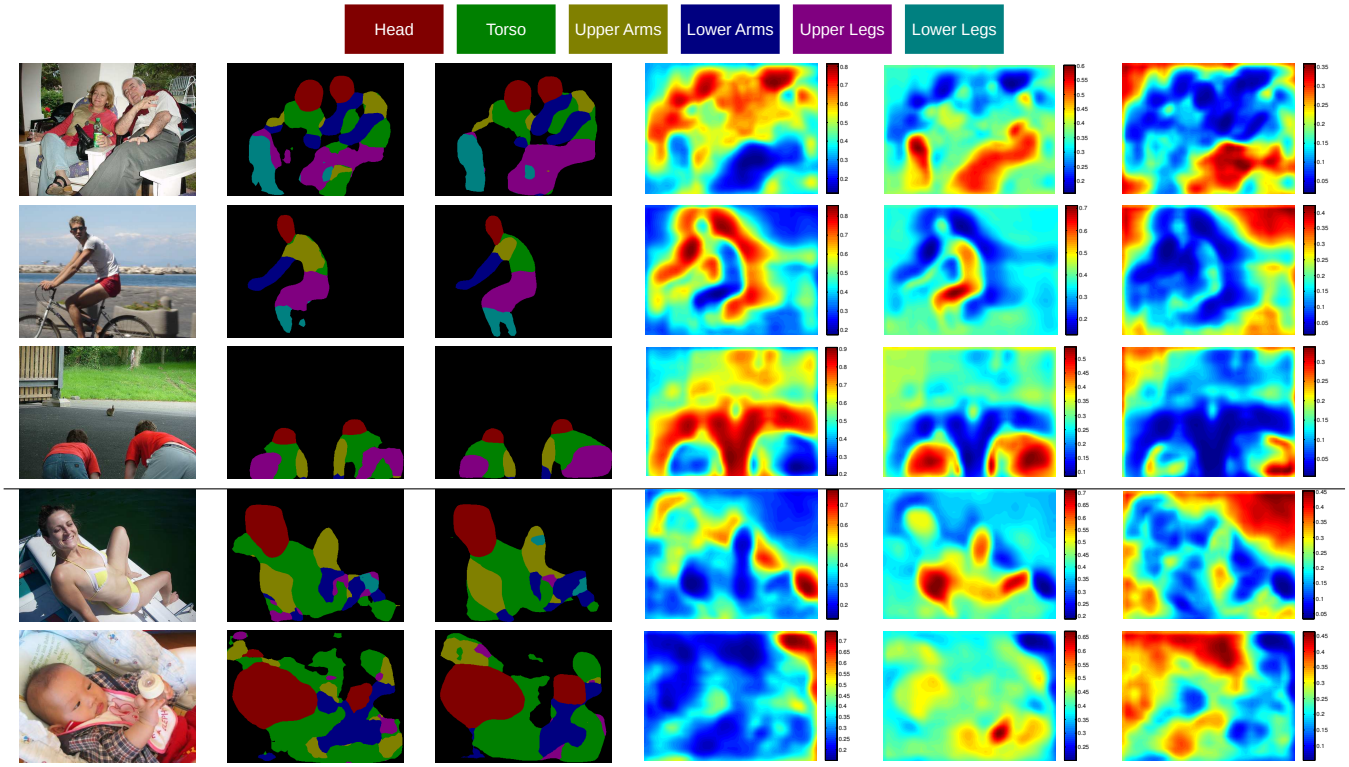
Dataset: The PASCAL VOC 2012 segmentation benchmark [18] consists of 20 foreground object classes and one background class. Following the same experimental protocol [11, 15, 58], we augment the original training set from the annotations by [26]. We report the results on the original PASCAL VOC 2012 validation set and test set.

Pretrained with ImageNet: First, we experiment with the scenario where the underlying DeepLab-LargeFOV is only pretrained on ImageNet [46]. Our reproduction of DeepLab-LargeFOV and DeepLab-MSc-LargeFOV yields performance of 62.28% and 64.39% on the validation



(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.



(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 the 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.

set, respectively. They are similar to those (62.25% and 64.21%) reported in [11]. We report results of the proposed methods on the validation set in Tab. 3. 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 the proposed attention model. (3) The best performance (6.8% improvement over the DeepLab-LargeFOV baseline) is obtained with three input scales, at-

Method	mIOU
Pretrained with ImageNet	
DeepLab-LargeFOV [11]	65.1
DeepLab-MSc-LargeFOV [11]	67.0
TTI_zoomout_v2 [41]	69.6
ParseNet [36]	69.8
DeepLab-LargeFOV-**AveragePooling**	70.5
DeepLab-LargeFOV-**MaxPooling**	70.6
DeepLab-LargeFOV-**Attention**	71.5
Pretrained with MS-COCO	
DeepLab-CRF-COCO-LargeFOV [43]	72.7
DeepLab-MSc-CRF-COCO-LargeFOV [43]	73.6
CRF-RNN [58]	74.7
BoxSup [15]	75.2
DPN [37]	77.5
Adelaide [34]	77.8
DeepLab-CRF-COCO-LargeFOV-**Attention**	75.1
DeepLab-CRF-COCO-LargeFOV-**Attention+**	75.7
DeepLab-CRF-Attention-DT [10]	76.3

Table 4. Labeling IOU on the PASCAL VOC 2012 test set.

tention model, and extra supervision, and its performance is 4.69% better than DeepLab-MSc-LargeFOV (64.39%).

We also report results on the test set for our best model in Tab. 4. First, we observe that the attention model yields a 1% improvement over average pooling, consistent with our results on the validation set. We then compare our models with DeepLab-LargeFOV and DeepLab-MSc-LargeFOV [11] *. We find that our proposed model improves 6.4% over DeepLab-LargeFOV, and gives a 4.5% boost over DeepLab-MSc_LargeFOV. Finally, we compare our models with two other methods: ParseNet [36] and TTI_zoomout_v2 [41]. ParseNet incorporates the image-level feature as global contextual information. We consider ParseNet as a special case to exploit multi-scale features, 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 employ a fully connected CRF [30].

Pretrained with MS-COCO: Second, we experiment with the scenario where the underlying baseline, DeepLab-LargeFOV, has been pretrained on the MS-COCO 2014 dataset [35]. The goal is to test if we can still observe any improvement with such a strong baseline. As shown in Tab. 5, we again observe similar experimental results, and our best model still outperforms the DeepLab-LargeFOV baseline by 3.84%. We also report the best model on the *test* set in the bottom of Tab. 4. For a fair comparison with

*test results are obtained by personal communication with authors [11]

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

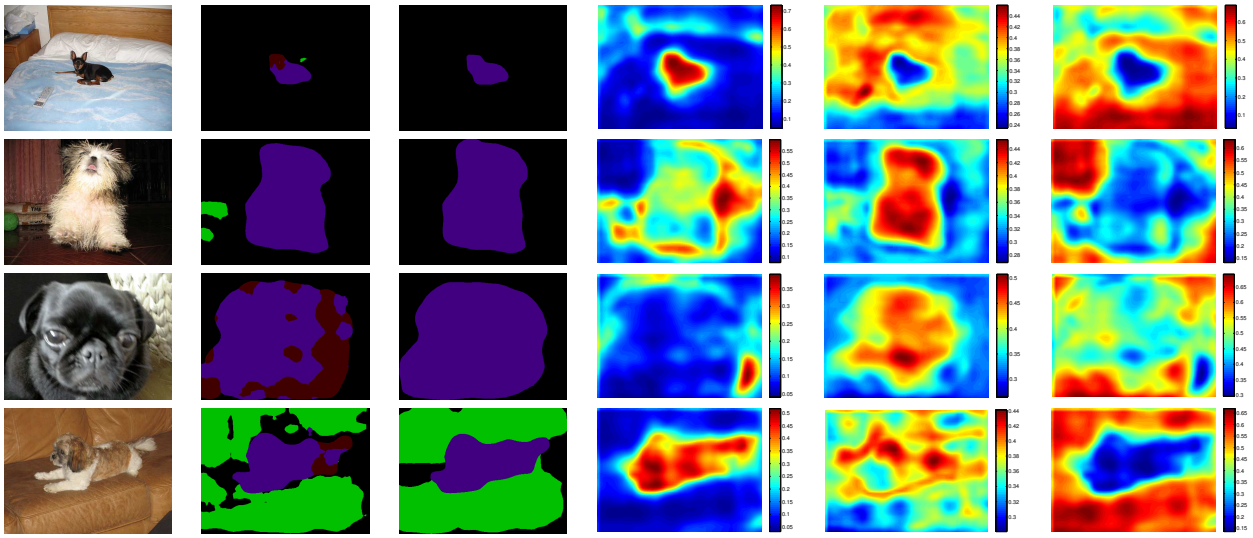
the reported DeepLab variants on the test set, we employ a fully connected CRF [30] as post processing. As shown in the table, our model attains the performance of 75.1%, outperforming DeepLab-CRF-LargeFOV and DeepLab-MSc-CRF-LaregeFOV by 2.4%, and 1.5%, respectively. Motivated by [34], incorporating data augmentation by randomly scaling input images (from 0.6 to 1.4) during training brings extra 0.6% improvement in our model.

Note our models do not outperform current best models [34, 37], which employ joint training of CRF (*e.g.*, with the *spatial* pairwise term) and FCNs [12]. However, we believe our proposed method (*e.g.*, attention model for scales) could be complementary to them. We emphasize that our models are trained end-to-end with one pass to exploit multi-scale features, instead of multiple training steps. Recently, [10] has been shown that further improvement can be attained by combining our proposed model and a discriminatively trained domain transform [23].

4.3. Subset of MS-COCO

Dataset: The MS-COCO 2014 dataset [35] contains 80 foreground object classes and one background class. The training set has about 80K images, and 40K images for validation. 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.

Improvement over 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. 6 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 bring 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 the attention model as merging



(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.

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 6. Results on the subset of MS-COCO *validation* set with DeepLab-LargeFOV as the 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 7. **Person** class IOU on subset of MS-COCO *validation* set with DeepLab-LargeFOV as baseline. E-Supv: extra supervision.

methods are very similar. We hypothesize that many small object classes (*e.g.*, fork, mouse, and toothbrush) with extremely low prediction accuracy reduce the improvement. This challenging problem (*i.e.*, segment small objects and handle imbalanced classes) is considered as future work. On the other hand, we show the performance for the *person* class in Tab. 7 because it occurs most frequently and appears with different scales (see Fig. 5(a), and Fig. 13(b) in [35]) 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. We leave the qualitative results in the supplementary material.

5. Conclusion

For semantic segmentation, this paper adapts a state-of-the-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 a 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.

Acknowledgments This work wast partly supported by ARO 62250-CS and NIH Grant 5R01EY022247-03. We thank Xiao-Chen Lian for valuable discussions. We also thank Sam Hallman and Haonan Yu for the proofreading.

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] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898–916, 2011.

[3] R. T. Azuma. A survey of augmented reality. *Presence: Teleoperators and virtual environments*, 6(4):355–385, 1997.

[4] J. Ba, V. Mnih, and K. Kavukcuoglu. Multiple object recognition with visual attention. *arXiv:1412.7755*, 2014.

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

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

[7] J. C. Caicedo and S. Lazebnik. Active object localization with deep reinforcement learning. In *ICCV*, 2015.

[8] C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *ICCV*, 2015.

[9] K. Chen, J. Wang, L.-C. Chen, H. Gao, W. Xu, and R. Nevatia. ABC-CNN: An attention based convolutional neural network for visual question answering. *arXiv:1511.05960*, 2015.

[10] L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. *arXiv:1511.03328*, 2015.

[11] 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.

[12] L.-C. Chen, A. Schwing, A. Yuille, and R. Urtasun. Learning deep structured models. In *ICML*, 2015.

[13] 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.

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

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

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

[17] M. Evening. *Adobe Photoshop CS2 for Photographers: A professional image editor's guide to the creative use of Photoshop for the Macintosh and PC*. Taylor & Francis, 2005.

[18] 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.

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

[20] 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.

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

[22] J. Fritsch, T. Kuhnl, and A. Geiger. A new performance measure and evaluation benchmark for road detection algorithms. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 1693–1700. IEEE, 2013.

[23] E. S. L. Gastal and M. M. Oliveira. Domain transform for edge-aware image and video processing. In *SIGGRAPH*, 2011.

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

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

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

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

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

[29] 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.

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

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

[32] 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.

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

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

[35] 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.

[36] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. *arXiv:1506.04579*, 2015.

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

[38] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

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

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

[41] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *CVPR*, 2015.

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

[43] 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.

[44] 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.

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

[46] 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.

[47] 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.

[48] S. Sharma, R. Kiros, and R. Salakhutdinov. Action recognition using visual attention. *arXiv:1511.04119*, 2015.

[49] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[50] 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.

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

[52] 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.

[53] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *CVPR*, 2015.

[54] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015.

[55] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv:1502.03044*, 2015.

[56] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *ICCV*, 2015.

[57] D. Yoo, S. Park, J.-Y. Lee, A. S. Paek, and I. So Kweon. Attentionnet: Aggregating weak directions for accurate object detection. In *ICCV*, 2015.

[58] 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.