SEMANTIC IMAGE SEGMENTATION WITH DEEP CONVOLUTIONAL NETS AND FULLY CONNECTED CRFS

Liang-Chieh Chen
Univ. of California, Los Angeles
lcchen@cs.ucla.edu

George Papandreou *
Google Inc.
gpapan@google.com

Iasonas Kokkinos
École Centrale Paris and INRIA
iasonas.kokkinos@ecp.fr

Kevin Murphy
Google Inc.
kpmurphy@google.com

Alan L. Yuille
Univ. of California, Los Angeles
yuille@stat.ucla.edu

ABSTRACT

Deep Convolutional Neural Networks (DCNNs) have recently shown state of the art performance in high level vision tasks, such as image classification and object detection. This work brings together methods from DCNNs and probabilistic graphical models for addressing the task of pixel-level classification (also called "semantic image segmentation"). We show that responses at the final layer of DCNNs are not sufficiently localized for accurate object segmentation. This is due to the very invariance properties that make DCNNs good for high level tasks. We overcome this poor localization property of deep networks by combining the responses at the final DCNN layer with a fully connected Conditional Random Field (CRF). Qualitatively, our "DeepLab" system is able to localize segment boundaries at a level of accuracy which is beyond previous methods. Quantitatively, our method sets the new state-of-art at the PASCAL VOC-2012 semantic image segmentation task, reaching 71.6% IOU accuracy in the test set. We show how these results can be obtained efficiently: Careful network re-purposing and a novel application of the 'hole' algorithm from the wavelet community allow dense computation of neural net responses at 8 frames per second on a modern GPU.

1 INTRODUCTION

Deep Convolutional Neural Networks (DCNNs) had been the method of choice for document recognition since LeCun et al. (1998), but have only recently become the mainstream of high-level vision research. Over the past two years DCNNs have pushed the performance of computer vision systems to soaring heights on a broad array of high-level problems, including image classification (Krizhevsky et al., 2013; Sermanet et al., 2013; Simonyan & Zisserman, 2014; Szegedy et al., 2014;

*Work initiated when G.P. was with the Toyota Technological Institute at Chicago. The first two authors contributed equally to this work.

Papandreou et al., 2014), object detection (Girshick et al., 2014), fine-grained categorization (Zhang et al., 2014), among others. A common theme in these works is that DCNNs trained in an end-to-end manner deliver strikingly better results than systems relying on carefully engineered representations, such as SIFT or HOG features. This success can be partially attributed to the built-in invariance of DCNNs to local image transformations, which underpins their ability to learn hierarchical abstractions of data (Zeiler & Fergus, 2014). While this invariance is clearly desirable for high-level vision tasks, it can hamper low-level tasks, such as pose estimation (Chen & Yuille, 2014; Tompson et al., 2014) and semantic segmentation - where we want precise localization, rather than abstraction of spatial details.

There are two technical hurdles in the application of DCNNs to image labeling tasks: signal downsampling, and spatial 'insensitivity' (invariance). The first problem relates to the reduction of signal resolution incurred by the repeated combination of max-pooling and downsampling ('striding') performed at every layer of standard DCNNs (Krizhevsky et al., 2013; Simonyan & Zisserman, 2014; Szegedy et al., 2014). Instead, as in Papandreou et al. (2014), we employ the 'atrous' (with holes) algorithm originally developed for efficiently computing the undecimated discrete wavelet transform (Mallat, 1999). This allows efficient dense computation of DCNN responses in a scheme substantially simpler than earlier solutions to this problem (Giusti et al., 2013; Sermanet et al., 2013).

The second problem relates to the fact that obtaining object-centric decisions from a classifier requires invariance to spatial transformations, inherently limiting the spatial accuracy of the DCNN model. We boost our model's ability to capture fine details by employing a fully-connected Conditional Random Field (CRF). Conditional Random Fields have been broadly used in semantic segmentation to combine class scores computed by multi-way classifiers with the low-level information captured by the local interactions of pixels and edges (Rother et al., 2004; Shotton et al., 2009) or superpixels (Lucchi et al., 2011). Even though works of increased sophistication have been proposed to model the hierarchical dependency (He et al., 2004; Ladicky et al., 2009; Lempitsky et al., 2011) and/or high-order dependencies of segments (Delong et al., 2012; Gonfaus et al., 2010; Kohli et al., 2009; Chen et al., 2013; Wang et al., 2015), we use the fully connected pairwise CRF proposed by Krähenbühl & Koltun (2011) for its efficient computation, and ability to capture fine edge details while also catering for long range dependencies. That model was shown in Krähenbühl & Koltun (2011) to largely improve the performance of a boosting-based pixel-level classifier, and in our work we demonstrate that it leads to state-of-the-art results when coupled with a DCNN-based pixel-level classifier.

The three main advantages of our "DeepLab" system are (i) speed: by virtue of the 'atrous' algorithm, our dense DCNN operates at 8 fps, while Mean Field Inference for the fully-connected CRF requires 0.5 second, (ii) accuracy: we obtain state-of-the-art results on the PASCAL semantic segmentation challenge, outperforming the second-best approach of Mostajabi et al. (2014) by a margin of 7.2% and (iii) simplicity: our system is composed of a cascade of two fairly well-established modules, DCNNs and CRFs.

2 Related Work

Our system works directly on the pixel representation, similarly to Long et al. (2014). This is in contrast to the two-stage approaches that are now most common in semantic segmentation with DCNNs: such techniques typically use a cascade of bottom-up image segmentation and DCNN-based region classification, which makes the system commit to potential errors of the front-end segmentation system. For instance, the bounding box proposals and masked regions delivered by (Arbeláez et al., 2014; Uijlings et al., 2013) are used in Girshick et al. (2014) and (Hariharan et al., 2014b) as inputs to a DCNN to introduce shape information into the classification process. Similarly, the authors of Mostajabi et al. (2014) rely on a superpixel representation. A celebrated non-DCNN precursor to these works is the second order pooling method of (Carreira et al., 2012) which also assigns labels to the regions proposals delivered by (Carreira & Sminchisescu, 2012). Understanding the perils of committing to a single segmentation, the authors of Cogswell et al. (2014) build on (Yadollahpour et al., 2013) to explore a diverse set of CRF-based segmentation proposals, computed also by (Carreira & Sminchisescu, 2012). These segmentation proposals are then re-ranked according to a DCNN trained in particular for this reranking task. Even though this approach explicitly tries to handle the temperamental nature of a front-end segmentation algorithm, there is still no explicit ex-

ploitation of the DCNN scores in the CRF-based segmentation algorithm: the DCNN is only applied post-hoc, while it would make sense to directly try to use its results *during* segmentation.

Moving towards works that lie closer to our approach, several other researchers have considered the use of convolutionally computed DCNN features for dense image labeling. Among the first have been Farabet et al. (2013) who apply DCNNs at multiple image resolutions and then employ a segmentation tree to smooth the prediction results; more recently, Hariharan et al. (2014a) propose to concatenate the computed inter-mediate feature maps within the DCNNs for pixel classification, and Dai et al. (2014) propose to pool the inter-mediate feature maps by region proposals. Even though these works still employ segmentation algorithms that are decoupled from the DCNN classifier's results, we believe it is advantageous that segmentation is only used at a later stage, avoiding the commitment to premature decisions.

More recently, the segmentation-free techniques of (Long et al., 2014; Eigen & Fergus, 2014) directly apply DCNNs to the whole image in a sliding window fashion, replacing the last fully connected layers of a DCNN by convolutional layers. In order to deal with the spatial localization issues outlined in the beginning of the introduction, Long et al. (2014) upsample and concatenate the scores from inter-mediate feature maps, while Eigen & Fergus (2014) refine the prediction result from coarse to fine by propagating the coarse results to another DCNN.

The main difference between our model and other state-of-the-art models is the combination of pixel-level CRFs and DCNN-based 'unary terms'. Focusing on the closest works in this direction, Cogswell et al. (2014) use CRFs as a proposal mechanism for a DCNN-based reranking system, while Farabet et al. (2013) treat superpixels as nodes for a local pairwise CRF and use graph-cuts for discrete inference; as such their results can be limited by errors in superpixel computations, while ignoring long-range superpixel dependencies. Our approach instead treats every pixel as a CRF node, exploits long-range dependencies, and uses CRF inference to directly optimize a DCNN-driven cost function.

After the first version of our manuscript was made publicly available, it came to our attention that two other groups have independently and concurrently pursued a very similar direction, combining DCNNs and densely connected CRFs (Bell et al., 2014; Zheng et al., 2015). There are several differences in technical aspects of the respective models. In terms of applications, Bell et al. (2014) focus on the problem of material classification. Similarly to us, Zheng et al. (2015) evaluate their system on the problem of semantic image segmentation but their results on the PASCAL VOC 2012 benchmark are somewhat inferior to ours. We refer the interested reader to these papers for different perspectives on the interplay of DCNNs and CRFs.

3 Convolutional Neural Networks for Dense Image Labeling

Herein we describe how we have re-purposed and finetuned the publicly available Imagenet-pretrained state-of-art 16-layer classification network of (Simonyan & Zisserman, 2014) (VGG-16) into an efficient and effective dense feature extractor for our dense semantic image segmentation system.

3.1 Efficient Dense Sliding Window Feature Extraction with the Hole Algorithm

Dense spatial score evaluation is instrumental in the success of our dense CNN feature extractor. As a first step to implement this, we convert the fully-connected layers of VGG-16 into convolutional ones and run the network in a convolutional fashion on the image at its original resolution. However this is not enough as it yields very sparsely computed detection scores (with a stride of 32 pixels). To compute scores more densely at our target stride of 8 pixels, we develop a variation of the method previously employed by Giusti et al. (2013); Sermanet et al. (2013). We skip subsampling after the last two max-pooling layers in the network of Simonyan & Zisserman (2014) and modify the convolutional filters in the layers that follow them by introducing zeros to increase their length ($2\times$ in the last three convolutional layers and $4\times$ in the first fully connected layer). We can implement this more efficiently by keeping the filters intact and instead sparsely sample the feature maps on which they are applied on using an input stride of 2 or 4 pixels, respectively. This approach, illustrated in Fig. 1 is known as the 'hole algorithm' ('atrous algorithm') and has been developed before for

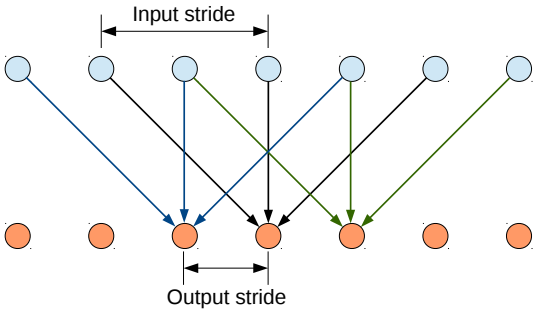


Figure 1: Illustration of the hole algorithm in 1-D, when *kernel_size = 3*, *input_stride = 2*, and *output_stride = 1*.

efficient computation of the undecimated wavelet transform (Mallat, 1999). We have implemented this within the Caffe framework (Jia et al., 2014) by adding to the *im2col* function (it converts multi-channel feature maps to vectorized patches) the option to sparsely sample the underlying feature map. This approach is generally applicable and allows us to efficiently compute dense CNN feature maps at any target subsampling rate without introducing any approximations.

We finetune the model weights of the Imagenet-pretrained VGG-16 network to adapt it to the image classification task in a straightforward fashion, following the procedure of Long et al. (2014). We replace the 1000-way Imagenet classifier in the last layer of VGG-16 with a 21-way one. Our loss function is the sum of cross-entropy terms for each spatial position in the CNN output map (subsampled by 8 compared to the original image). All positions and labels are equally weighted in the overall loss function. Our targets are the ground truth labels (subsampled by 8). We optimize the objective function with respect to the weights at all network layers by the standard SGD procedure of Krizhevsky et al. (2013).

During testing, we need class score maps at the original image resolution. As illustrated in Figure 2 and further elaborated in Section 4.1, the class score maps (corresponding to log-probabilities) are quite smooth, which allows us to use simple bilinear interpolation to increase their resolution by a factor of 8 at a negligible computational cost. Note that the method of Long et al. (2014) does not use the hole algorithm and produces very coarse scores (subsampled by a factor of 32) at the CNN output. This forced them to use learned upsampling layers, significantly increasing the complexity and training time of their system: Fine-tuning our network on PASCAL VOC 2012 takes about 10 hours, while they report a training time of several days (both timings on a modern GPU).

3.2 Controlling the Receptive Field Size and Accelerating Dense Computation with Convolutional Nets

Another key ingredient in re-purposing our network for dense score computation is explicitly controlling the network's receptive field size. Most recent DCNN-based image recognition methods rely on networks pre-trained on the Imagenet large-scale classification task. These networks typically have large receptive field size: in the case of the VGG-16 net we consider, its receptive field is 224×224 (with zero-padding) and 404×404 pixels if the net is applied convolutionally. After converting the network to a fully convolutional one, the first fully connected layer has 4,096 filters of large 7×7 spatial size and becomes the computational bottleneck in our dense score map computation.

We have addressed this practical problem by spatially subsampling (by simple decimation) the first FC layer to 4×4 (or 3×3) spatial size. This has reduced the receptive field of the network down to 128×128 (with zero-padding) or 308×308 (in convolutional mode) and has reduced computation time for the first FC layer by $2 - 3$ times. Using our Caffe-based implementation and a Titan GPU, the resulting VGG-derived network is very efficient: Given a 306×306 input image, it produces 39×39 dense raw feature scores at the top of the network at a rate of about 8 frames/sec during testing. The speed during training is 3 frames/sec. We have also successfully experimented with reducing the number of channels at the fully connected layers from 4,096 down to 1,024, considerably further decreasing computation time and memory footprint without sacrificing performance, as detailed in

Section 5. Using smaller networks such as Krizhevsky et al. (2013) could allow video-rate test-time dense feature computation even on light-weight GPUs.

4 Detailed Boundary Recovery: Fully-Connected Conditional Random Fields and Multi-scale Prediction

4.1 Deep Convolutional Networks and the Localization Challenge

As illustrated in Figure 2, DCNN score maps can reliably predict the presence and rough position of objects in an image but are less well suited for pin-pointing their exact outline. There is a natural trade-off between classification accuracy and localization accuracy with convolutional networks: Deeper models with multiple max-pooling layers have proven most successful in classification tasks, however their increased invariance and large receptive fields make the problem of inferring position from the scores at their top output levels more challenging.

Recent work has pursued two directions to address this localization challenge. The first approach is to harness information from multiple layers in the convolutional network in order to better estimate the object boundaries (Long et al., 2014; Eigen & Fergus, 2014). The second approach is to employ a super-pixel representation, essentially delegating the localization task to a low-level segmentation method. This route is followed by the very successful recent method of Mostajabi et al. (2014).

In Section 4.2, we pursue a novel alternative direction based on coupling the recognition capacity of DCNNs and the fine-grained localization accuracy of fully connected CRFs and show that it is remarkably successful in addressing the localization challenge, producing accurate semantic segmentation results and recovering object boundaries at a level of detail that is well beyond the reach of existing methods.

4.2 Fully-Connected Conditional Random Fields for Accurate Localization

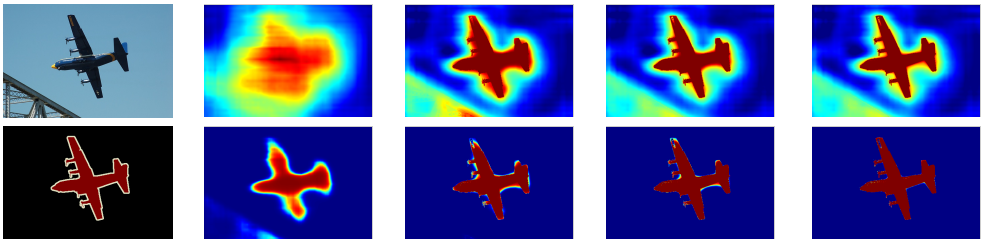


Figure 2: Score map (input before softmax function) and belief map (output of softmax function) for Aeroplane. We show the score (1st row) and belief (2nd row) maps after each mean field iteration. The output of last DCNN layer is used as input to the mean field inference. Best viewed in color.

Traditionally, conditional random fields (CRFs) have been employed to smooth noisy segmentation maps (Rother et al., 2004; Kohli et al., 2009). Typically these models contain energy terms that couple neighboring nodes, favoring same-label assignments to spatially proximal pixels. Qualitatively, the primary function of these short-range CRFs has been to clean up the spurious predictions of weak classifiers built on top of local hand-engineered features.

Compared to these weaker classifiers, modern DCNN architectures such as the one we use in this work produce score maps and semantic label predictions which are qualitatively different. As illustrated in Figure 2, the score maps are typically quite smooth and produce homogeneous classification results. In this regime, using short-range CRFs can be detrimental, as our goal should be to recover detailed local structure rather than further smooth it. Using contrast-sensitive potentials (Rother et al., 2004) in conjunction to local-range CRFs can potentially improve localization but still miss thin-structures and typically requires solving an expensive discrete optimization problem.

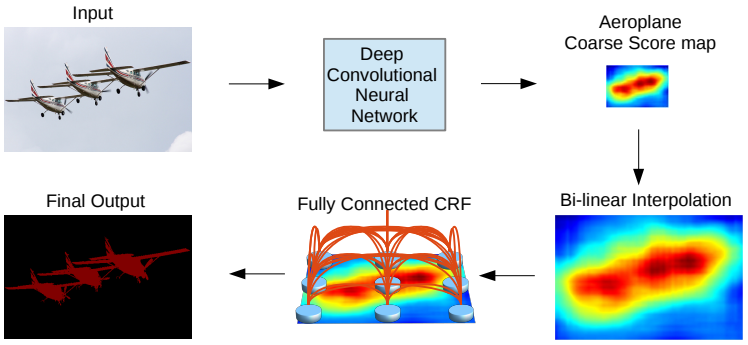


Figure 3: Model Illustration. The coarse score map from Deep Convolutional Neural Network (with fully convolutional layers) is upsampled by bi-linear interpolation. A fully connected CRF is applied to refine the segmentation result. Best viewed in color.

To overcome these limitations of short-range CRFs, we integrate into our system the fully connected CRF model of Krähenbühl & Koltun (2011). The model employs the energy function

$$E(\boldsymbol{x}) = \sum_i \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j) \tag{1}$$

where $\boldsymbol{x}$ is the label assignment for pixels. We use as unary potential $\theta_i(x_i) = -\log P(x_i)$, where $P(x_i)$ is the label assignment probability at pixel i as computed by DCNN. The pairwise potential is $\theta_{ij}(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{K} w_m \cdot k^m(\boldsymbol{f}_i, \boldsymbol{f}_j)$, where $\mu(x_i, x_j) = 1$ if $x_i \neq x_j$, and zero otherwise (*i.e.*, Potts Model). There is one pairwise term for each pair of pixels i and j in the image no matter how far from each other they lie, *i.e.* the model's factor graph is fully connected. Each k^m is the Gaussian kernel depends on features (denoted as $\boldsymbol{f}$) extracted for pixel i and j and is weighted by parameter w_m. We adopt bilateral position and color terms, specifically, the kernels are

$$w_1 \exp\Big(-\frac{||p_i - p_j||^2}{2\sigma_\alpha^2} - \frac{||I_i - I_j||^2}{2\sigma_\beta^2}\Big) + w_2 \exp\Big(-\frac{||p_i - p_j||^2}{2\sigma_\gamma^2}\Big) \tag{2}$$

where the first kernel depends on both pixel positions (denoted as p) and pixel color intensities (denoted as I), and the second kernel only depends on pixel positions. The hyper parameters σ_α, σ_β and σ_γ control the "scale" of the Gaussian kernels.

Crucially, this model is amenable to efficient approximate probabilistic inference (Krähenbühl & Koltun, 2011). The message passing updates under a fully decomposable mean field approximation $b(\boldsymbol{x}) = \prod_i b_i(x_i)$ can be expressed as convolutions with a Gaussian kernel in feature space. High-dimensional filtering algorithms (Adams et al., 2010) significantly speed-up this computation resulting in an algorithm that is very fast in practice, less that 0.5 sec on average for Pascal VOC images using the publicly available implementation of (Krähenbühl & Koltun, 2011).

4.3 Multi-Scale Prediction

Following the promising recent results of (Hariharan et al., 2014a; Long et al., 2014) we have also explored a multi-scale prediction method to increase the boundary localization accuracy. Specifically, we attach to the input image and the output of each of the first four max pooling layers a two-layer MLP (first layer: 128 3x3 convolutional filters, second layer: 128 1x1 convolutional filters) whose feature map is concatenated to the main network's last layer feature map. The aggregate feature map fed into the softmax layer is thus enhanced by 5 * 128 = 640 channels. We only adjust the newly added weights, keeping the other network parameters to the values learned by the method of Section 3. As discussed in the experimental section, introducing these extra direct connections from fine-resolution layers improves localization performance, yet the effect is not as dramatic as the one obtained with the fully-connected CRF.

Method	mean IOU (%)
DeepLab	59.80
DeepLab-CRF	63.74
DeepLab-MSc	61.30
DeepLab-MSc-CRF	65.21
DeepLab-7x7	64.38
DeepLab-CRF-7x7	67.64
DeepLab-LargeFOV	62.25
DeepLab-CRF-LargeFOV	67.64
DeepLab-MSc-LargeFOV	64.21
DeepLab-MSc-CRF-LargeFOV	68.70

(a)

Method	mean IOU (%)
MSRA-CFM	61.8
FCN-8s	62.2
TTI-Zoomout-16	64.4
DeepLab-CRF	66.4
DeepLab-MSc-CRF	67.1
DeepLab-CRF-7x7	70.3
DeepLab-CRF-LargeFOV	70.3
DeepLab-MSc-CRF-LargeFOV	71.6

(b)

Table 1: (a) Performance of our proposed models on the PASCAL VOC 2012 'val' set (with training in the augmented 'train' set). The best performance is achieved by exploiting both multi-scale features and large field-of-view. (b) Performance of our proposed models (with training in the augmented 'trainval' set) compared to other state-of-art methods on the PASCAL VOC 2012 'test' set.

5 Experimental Evaluation

Dataset We test our DeepLab model on the PASCAL VOC 2012 segmentation benchmark (Everingham et al., 2014), consisting of 20 foreground object classes and one background class. The original dataset contains $1,464$, $1,449$, and $1,456$ images for training, validation, and testing, respectively. The dataset is augmented by the extra annotations provided by Hariharan et al. (2011), resulting in $10,582$ training images. The performance is measured in terms of pixel intersection-over-union (IOU) averaged across the 21 classes.

Training We adopt the simplest form of piecewise training, decoupling the DCNN and CRF training stages, assuming the unary terms provided by the DCNN are fixed during CRF training.

For DCNN training we employ the VGG-16 network which has been pre-trained on ImageNet. We fine-tuned the VGG-16 network on the VOC 21-way pixel-classification task by stochastic gradient descent on the cross-entropy loss function, as described in Section 3.1. We use a mini-batch of 20 images and initial learning rate of 0.001 (0.01 for the final classifier layer), multiplying the learning rate by 0.1 at every 2000 iterations. We use momentum of 0.9 and a weight decay of 0.0005.

After the DCNN has been fine-tuned, we cross-validate the parameters of the fully connected CRF model in Eq. (2) along the lines of Krähenbühl & Koltun (2011). We use the default values of $w_2 = 3$ and $\sigma_\gamma = 3$ and we search for the best values of w_1, σ_α, and σ_β by cross-validation on a small subset of the validation set (we use 200 images). We employ coarse-to-fine search scheme. Specifically, the initial search range of the parameters are $w_1 \in [5, 10]$, $\sigma_\alpha \in [50 : 10 : 100]$ and $\sigma_\beta \in [3 : 1 : 10]$ (MATLAB notation), and then we refine the search step sizes around the first round's best values. We fix the number of mean field iterations to 10 for all reported experiments.

Evaluation on Validation set We conduct the majority of our evaluations on the PASCAL 'val' set, training our model on the augmented PASCAL 'train' set. As shown in Tab. 1 (a), incorporating the fully connected CRF to our model (denoted by DeepLab-CRF) yields a substantial performance boost, about 4% improvement over DeepLab. We note that the work of Krähenbühl & Koltun (2011) improved the 27.6% result of TextonBoost (Shotton et al., 2009) to 29.1%, which makes the improvement we report here (from 59.8% to 63.7%) all the more impressive.

Turning to qualitative results, we provide visual comparisons between DeepLab and DeepLab-CRF in Fig. 7. Employing a fully connected CRF significantly improves the results, allowing the model to accurately capture intricate object boundaries.

Multi-Scale features We also exploit the features from the intermediate layers, similar to Hariharan et al. (2014a); Long et al. (2014). As shown in Tab. 1 (a), adding the multi-scale features to our

Method	kernel size	input stride	receptive field	# parameters	mean IOU (%)	Training speed (img/sec)
DeepLab-CRF-7x7	7×7	4	224	134.3M	67.64	1.44
DeepLab-CRF	4×4	4	128	65.1M	63.74	2.90
DeepLab-CRF-4x4	4×4	8	224	65.1M	67.14	2.90
DeepLab-CRF-LargeFOV	3×3	12	224	20.5M	67.64	4.84

Table 2: Effect of Field-Of-View. We show the performance (after CRF) and training speed on the PASCAL VOC 2012 'val' set as the function of (1) the kernel size of first fully connected layer, (2) the input stride value employed in the atrous algorithm.

DeepLab model (denoted as DeepLab-MSc) improves about 1.5% performance, and further incorporating the fully connected CRF (denoted as DeepLab-MSc-CRF) yields about 4% improvement. The qualitative comparisons between DeepLab and DeepLab-MSc are shown in Fig. 4. Leveraging the multi-scale features can slightly refine the object boundaries.

Field of View The 'atrous algorithm' we employed allows us to arbitrarily control the Field-of-View (FOV) of the models by adjusting the input stride, as illustrated in Fig. 1. In Tab. 2, we experiment with several kernel sizes and input strides at the first fully connected layer. The method, DeepLab-CRF-7x7, is the direct modification from VGG-16 net, where the kernel size = 7×7 and input stride = 4. This model yields performance of 67.64% on the 'val' set, but it is relatively slow (1.44 images per second during training). We have improved model speed to 2.9 images per second by reducing the kernel size to 4×4. We have experimented with two such network variants with different FOV sizes, DeepLab-CRF and DeepLab-CRF-4x4; the latter has large FOV (*i.e.*, large input stride) and attains better performance. Finally, we employ kernel size 3×3 and input stride = 12, and further change the filter sizes from 4096 to 1024 for the last two layers. Interestingly, the resulting model, DeepLab-CRF-LargeFOV, matches the performance of the expensive DeepLab-CRF-7x7. At the same time, it is 3.36 times faster to run and has significantly fewer parameters (20.5M instead of 134.3M).

The performance of several model variants is summarized in Tab. 1, showing the benefit of exploiting multi-scale features and large FOV.

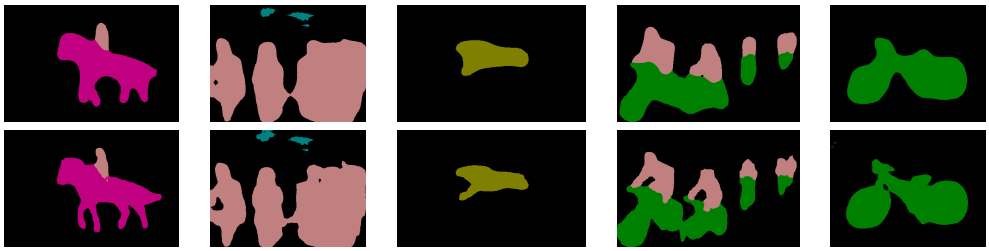
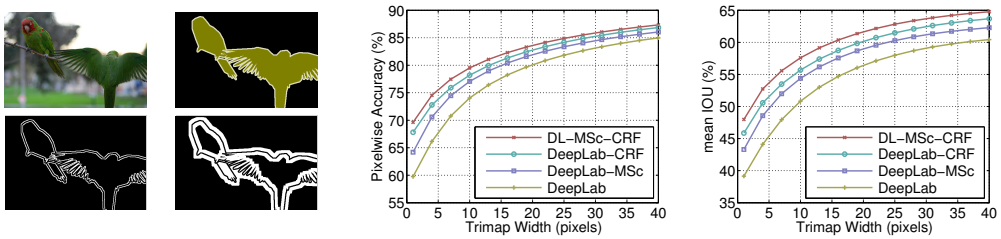


Figure 4: Incorporating multi-scale features improves the boundary segmentation. We show the results obtained by DeepLab and DeepLab-MSc in the first and second row, respectively. Best viewed in color.

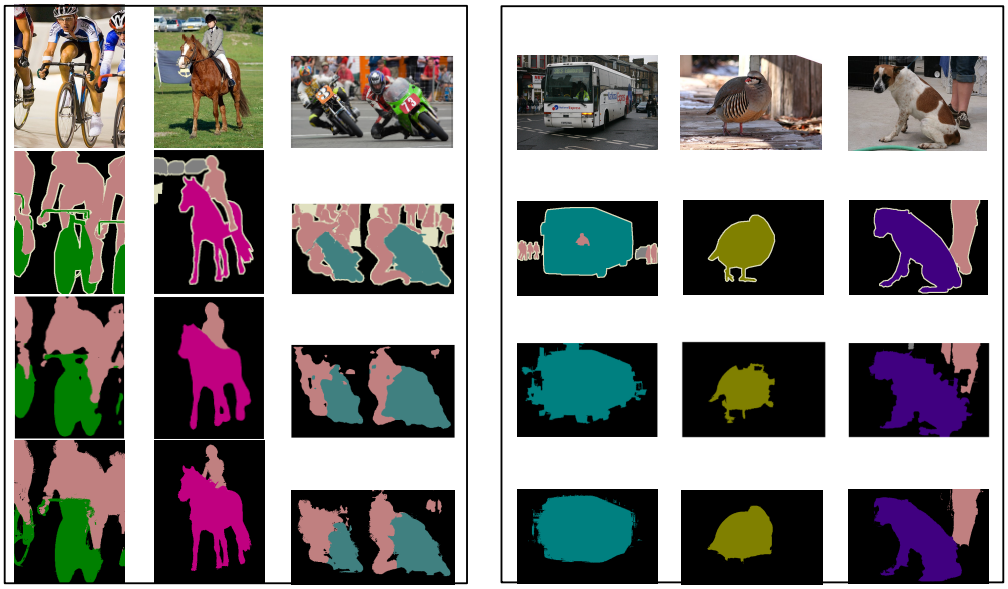
Mean Pixel IOU along Object Boundaries To quantify the accuracy of the proposed model near object boundaries, we evaluate the segmentation accuracy with an experiment similar to Kohli et al. (2009); Krähenbühl & Koltun (2011). Specifically, we use the 'void' label annotated in val set, which usually occurs around object boundaries. We compute the mean IOU for those pixels that are located within a narrow band (called trimap) of 'void' labels. As shown in Fig. 5, exploiting the multi-scale features from the intermediate layers and refining the segmentation results by a fully connected CRF significantly improve the results around object boundaries.

Comparison with State-of-art In Fig. 6, we qualitatively compare our proposed model, DeepLab-CRF, with two state-of-art models: FCN-8s (Long et al., 2014) and TTI-Zoomout-16 (Mostajabi et al., 2014) on the 'val' set (the results are extracted from their papers). Our model is able to capture the intricate object boundaries.



(a) (b) (c)

Figure 5: (a) Some trimap examples (top-left: image. top-right: ground-truth. bottom-left: trimap of 2 pixels. bottom-right: trimap of 10 pixels). Quality of segmentation result within a band around the object boundaries for the proposed methods. (b) Pixelwise accuracy. (c) Pixel mean IOU.



(a) FCN-8s vs. DeepLab-CRF (b) TTI-Zoomout-16 vs. DeepLab-CRF

Figure 6: Comparisons with state-of-the-art models on the val set. First row: images. Second row: ground truths. Third row: other recent models (Left: FCN-8s, Right: TTI-Zoomout-16). Fourth row: our DeepLab-CRF. Best viewed in color.

Reproducibility We have implemented the proposed methods by extending the excellent Caffe framework (Jia et al., 2014). We share our source code, configuration files, and trained models that allow reproducing the results in this paper at a companion web site `https://bitbucket.org/deeplab/deeplab-public`.

Test set results Having set our model choices on the validation set, we evaluate our model variants on the PASCAL VOC 2012 official 'test' set. As shown in Tab. 3, our DeepLab-CRF and DeepLab-MSc-CRF models achieve performance of 66.4% and 67.1% mean IOU[1], respectively. Our models outperform all the other state-of-the-art models (specifically, TTI-Zoomout-16 (Mostajabi et al., 2014), FCN-8s (Long et al., 2014), and MSRA-CFM (Dai et al., 2014)). When we increase the FOV of the models, DeepLab-CRF-LargeFOV yields performance of 70.3%, the same as DeepLab-CRF-7x7, while its training speed is faster. Furthermore, our best model, DeepLab-MSc-CRF-LargeFOV, attains the best performance of 71.6% by employing both multi-scale features and large FOV.

[1] `http://host.robots.ox.ac.uk:8080/leaderboard/displaylb.php?challengeid=11&compid=6`

Figure 7: Visualization results on VOC 2012-val. For each row, we show the input image, the segmentation result delivered by the DCNN (DeepLab), and the refined segmentation result of the Fully Connected CRF (DeepLab-CRF). We show our failure modes in the last three rows. Best viewed in color.

Method	bkg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
MSRA-CFM	-	75.7	26.7	69.5	48.8	65.6	81.0	69.2	73.3	30.0	68.7	51.5	69.1	68.1	71.7	67.5	50.4	66.5	44.4	58.9	53.5	61.8
FCN-8s	-	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
TTI-Zoomout-16	89.8	81.9	35.1	78.2	57.4	56.5	80.5	74.0	79.8	22.4	69.6	53.7	74.0	76.0	76.6	68.8	44.3	70.2	40.2	68.9	55.3	64.4
DeepLab-CRF	92.1	78.4	33.1	78.2	55.6	65.3	81.3	75.5	78.6	25.3	69.2	52.7	75.2	69.0	79.1	77.6	54.7	78.3	45.1	73.3	56.2	66.4
DeepLab-MSc-CRF	92.6	80.4	36.8	77.4	55.2	66.4	81.5	77.5	78.9	27.1	68.2	52.7	74.3	69.6	79.4	79.0	56.9	78.8	45.2	72.7	59.3	67.1
DeepLab-CRF-7x7	92.8	83.9	36.6	77.5	58.4	**68.0**	84.6	**79.7**	83.1	29.5	**74.6**	59.3	78.9	76.0	82.1	80.6	**60.3**	81.7	49.2	**78.0**	60.7	70.3
DeepLab-CRF-LargeFOV	92.6	83.5	36.6	**82.5**	62.3	66.5	**85.4**	78.5	**83.7**	30.4	72.9	**60.4**	78.5	75.5	82.1	79.7	58.2	82.0	48.8	73.7	63.3	70.3
DeepLab-MSc-CRF-LargeFOV	**93.1**	**84.4**	**54.5**	81.5	**63.6**	65.9	85.1	79.1	83.4	**30.7**	74.1	59.8	**79.0**	**76.1**	**83.2**	**80.8**	59.7	**82.2**	**50.4**	73.1	**63.7**	**71.6**

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

6 DISCUSSION

Our work combines ideas from deep convolutional neural networks and fully-connected conditional random fields, yielding a novel method able to produce semantically accurate predictions and detailed segmentation maps, while being computationally efficient. Our experimental results show that the proposed method significantly advances the state-of-art in the challenging PASCAL VOC 2012 semantic image segmentation task.

There are multiple aspects in our model that we intend to refine, such as fully integrating its two main components (CNN and CRF) and train the whole system in an end-to-end fashion, similar to Krähenbühl & Koltun (2013); Chen et al. (2014); Zheng et al. (2015). We also plan to experiment with more datasets and apply our method to other sources of data such as depth maps or videos. Recently, we have pursued model training with weakly supervised annotations, in the form of bounding boxes or image-level labels (Papandreou et al., 2015).

At a higher level, our work lies in the intersection of convolutional neural networks and probabilistic graphical models. We plan to further investigate the interplay of these two powerful classes of methods and explore their synergistic potential for solving challenging computer vision tasks.

ACKNOWLEDGMENTS

This work was partly supported by NIH Grant 5R01EY022247-03. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of GPUs used for this research. We would like to thank the anonymous reviewers for their detailed comments and constructive feedback.

PAPER REVISIONS

Here we present the list of major paper revisions for the convenience of the readers.

v1 Submission to ICLR 2015. Introduces the model DeepLab-CRF, which attains the performance of 66.4% on PASCAL VOC 2012 test set.

v2 Rebuttal for ICLR 2015. Adds the model DeepLab-MSc-CRF, which incorporates multi-scale features from the intermediate layers. DeepLab-MSc-CRF yields the performance of 67.1% on PASCAL VOC 2012 test set.

v3 Camera-ready for ICLR 2015. Experiments with large Field-Of-View. On PASCAL VOC 2012 test set, DeepLab-CRF-LargeFOV achieves the performance of 70.3%. When exploiting both mutliscale features and large FOV, DeepLab-MSc-CRF-LargeFOV attains the performance of 71.6%.

REFERENCES

Adams, A., Baek, J., and Davis, M. A. Fast high-dimensional filtering using the permutohedral lattice. In *Computer Graphics Forum*, 2010.

Arbeláez, P., Pont-Tuset, J., Barron, J. T., Marques, F., and Malik, J. Multiscale combinatorial grouping. In *CVPR*, 2014.

Bell, S., Upchurch, P., Snavely, N., and Bala, K. Material recognition in the wild with the materials in context database. *arXiv:1412.0623*, 2014.

Carreira, J. and Sminchisescu, C. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *PAMI*, 2012.

Carreira, J., Caseiro, R., Batista, J., and Sminchisescu, C. Semantic segmentation with second-order pooling. In *ECCV*, 2012.

Chen, L.-C., Papandreou, G., and Yuille, A. Learning a dictionary of shape epitomes with applications to image labeling. In *ICCV*, 2013.

Chen, L.-C., Schwing, A., Yuille, A., and Urtasun, R. Learning deep structured models. *arXiv:1407.2538*, 2014.

Chen, X. and Yuille, A. L. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*, 2014.

Cogswell, M., Lin, X., Purushwalkam, S., and Batra, D. Combining the best of graphical models and convnets for semantic segmentation. *arXiv:1412.4313*, 2014.

Dai, J., He, K., and Sun, J. Convolutional feature masking for joint object and stuff segmentation. *arXiv:1412.1283*, 2014.

Delong, A., Osokin, A., Isack, H. N., and Boykov, Y. Fast approximate energy minimization with label costs. *IJCV*, 2012.

Eigen, D. and Fergus, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *arXiv:1411.4734*, 2014.

Everingham, M., Eslami, S. M. A., Gool, L. V., Williams, C. K. I., Winn, J., and Zisserma, A. The pascal visual object classes challenge a retrospective. *IJCV*, 2014.

Farabet, C., Couprie, C., Najman, L., and LeCun, Y. Learning hierarchical features for scene labeling. *PAMI*, 2013.

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

Giusti, A., Ciresan, D., Masci, J., Gambardella, L., and Schmidhuber, J. Fast image scanning with deep max-pooling convolutional neural networks. In *ICIP*, 2013.

Gonfaus, J. M., Boix, X., Van de Weijer, J., Bagdanov, A. D., Serrat, J., and Gonzalez, J. Harmony potentials for joint classification and segmentation. In *CVPR*, 2010.

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

Hariharan, B., Arbeláez, P., Girshick, R., and Malik, J. Hypercolumns for object segmentation and fine-grained localization. *arXiv:1411.5752*, 2014a.

Hariharan, B., Arbeláez, P., Girshick, R., and Malik, J. Simultaneous detection and segmentation. In *ECCV*, 2014b.

He, X., Zemel, R. S., and Carreira-Perpindn, M. Multiscale conditional random fields for image labeling. In *CVPR*, 2004.

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

Kohli, P., Ladicky, L., and Torr, P. H. Robust higher order potentials for enforcing label consistency. *IJCV*, 2009.

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

Krähenbühl, P. and Koltun, V. Parameter learning and convergent inference for dense random fields. In *ICML*, 2013.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2013.

Ladicky, L., Russell, C., Kohli, P., and Torr, P. H. Associative hierarchical crfs for object class image segmentation. In *ICCV*, 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. In *Proc. IEEE*, 1998.

Lempitsky, V., Vedaldi, A., and Zisserman, A. Pylon model for semantic segmentation. In *NIPS*, 2011.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. *arXiv:1411.4038*, 2014.

Lucchi, A., Li, Y., Boix, X., Smith, K., and Fua, P. Are spatial and global constraints really necessary for segmentation? In *ICCV*, 2011.

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

Mostajabi, M., Yadollahpour, P., and Shakhnarovich, G. Feedforward semantic segmentation with zoom-out features. *arXiv:1412.0774*, 2014.

Papandreou, G., Kokkinos, I., and Savalle, P.-A. Untangling local and global deformations in deep convolutional networks for image classification and sliding window detection. *arXiv:1412.0296*, 2014.

Papandreou, G., Chen, L.-C., Murphy, K., and Yuille, A. L. Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. 2015. URL `http://arxiv.org/abs/1502.02734`.

Rother, C., Kolmogorov, V., and Blake, A. Grabcut: Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv:1312.6229*, 2013.

Shotton, J., Winn, J., Rother, C., and Criminisi, A. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 2009.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

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

Tompson, J., Jain, A., LeCun, Y., and Bregler, C. Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. In *NIPS*, 2014.

Uijlings, J., van de Sande, K., Gevers, T., and Smeulders, A. Selective search for object recognition. *IJCV*, 2013.

Wang, P., Shen, X., Lin, Z., Cohen, S., Price, B., and Yuille, A. Towards unified depth and semantic prediction from a single image. In *CVPR*, 2015.

Yadollahpour, P., Batra, D., and Shakhnarovich, G. Discriminative re-ranking of diverse segmentations. In *CVPR*, 2013.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

Zhang, N., Donahue, J., Girshick, R., and Darrell, T. Part-based r-cnns for fine-grained category detection. In *ECCV*, 2014.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. Conditional random fields as recurrent neural networks. *arXiv:1502.03240*, 2015.