Weakly- and Semi-Supervised Learning of a Deep Convolutional Network for Semantic Image Segmentation

George Papandreou*
Google, Inc.
gpapan@google.com

Liang-Chieh Chen*
UCLA
lcchen@cs.ucla.edu

Kevin Murphy
Google, Inc.
kpmurphy@google.com

Alan L. Yuille
UCLA
yuille@stat.ucla.edu

Abstract

Deep convolutional neural networks (DCNNs) trained on a large number of images with strong pixel-level annotations have recently significantly pushed the state-of-art in semantic image segmentation. We study the more challenging problem of learning DCNNs for semantic image segmentation from either (1) weakly annotated training data such as bounding boxes or image-level labels or (2) a combination of few strongly labeled and many weakly labeled images, sourced from one or multiple datasets. We develop Expectation-Maximization (EM) methods for semantic image segmentation model training under these weakly supervised and semi-supervised settings. Extensive experimental evaluation shows that the proposed techniques can learn models delivering competitive results on the challenging PASCAL VOC 2012 image segmentation benchmark, while requiring significantly less annotation effort. We share source code implementing the proposed system at https://bitbucket.org/deeplab/deeplab-public.

1. Introduction

Semantic image segmentation refers to the problem of assigning a semantic label (such as “person”, “car” or “dog”) to every pixel in the image. Various approaches have been tried over the years, but according to the results on the challenging Pascal VOC 2012 segmentation benchmark, the best performing methods all use some kind of Deep Convolutional Neural Network (DCNN) [2, 5, 8, 14, 25, 28, 42].

In this paper, we work with the DeepLab-CRF approach of [5, 42]. This combines a DCNN with a fully connected Conditional Random Field (CRF) [19], in order to get high resolution segmentations. This model achieves state-of-art results on the challenging PASCAL VOC segmentation benchmark [13], delivering a mean intersection-over-union (IOU) score exceeding 70%.

A key bottleneck in building this class of DCNN-based segmentation models is that they typically require pixel-level annotated images during training. Acquiring such data is an expensive, time-consuming annotation effort. Weak annotations, in the form of bounding boxes (*i.e.*, coarse object locations) or image-level labels (*i.e.*, information about which object classes are present) are far easier to collect than detailed pixel-level annotations. We develop new methods for training DCNN image segmentation models from weak annotations, either alone or in combination with a small number of strong annotations. Extensive experiments, in which we achieve performance up to 69.0%, demonstrate the effectiveness of the proposed techniques.

According to [24], collecting bounding boxes around each class instance in the image is about 15 times faster/cheaper than labeling images at the pixel level. We demonstrate that it is possible to learn a DeepLab-CRF model delivering 62.2% IOU on the PASCAL VOC 2012 test set by training it on a simple foreground/background segmentation of the bounding box annotations.

An even cheaper form of data to collect is image-level labels, which specify the presence or absence of semantic classes, but not the object locations. Most existing approaches for training semantic segmentation models from this kind of very weak labels use multiple instance learning (MIL) techniques. However, even recent weakly-supervised methods such as [25] deliver significantly inferior results compared to their fully-supervised counterparts, only achieving 25.7%. Including additional trainable objectness [7] or segmentation [1] modules that largely increase the system complexity, [32] has improved performance to 40.6%, which still significantly lags performance of fully-supervised systems.

We develop novel online Expectation-Maximization (EM) methods for training DCNN semantic segmentation models from weakly annotated data. The proposed algorithms alternate between estimating the latent pixel labels (subject to the weak annotation constraints), and optimizing the DCNN parameters using stochastic gradient descent (SGD). When we only have access to image-level annotated training data, we achieve 39.6%, close to [32] but

[*] The first two authors contributed equally to this work.

without relying on any external objectness or segmentation module. More importantly, our EM approach also excels in the semi-supervised scenario which is very important in practice. Having access to a small number of strongly (pixel-level) annotated images and a large number of weakly (bounding box or image-level) annotated images, the proposed algorithm can almost match the performance of the fully-supervised system. For example, having access to 2.9k pixel-level images and 9k image-level annotated images yields 68.5%, only 2% inferior the performance of the system trained with all 12k images strongly annotated at the pixel level. Finally, we show that using additional weak or strong annotations from the MS-COCO dataset can further improve results, yielding 73.9% on the PASCAL VOC 2012 benchmark.

Contributions In summary, our main contributions are:

1. We present EM algorithms for training with image-level or bounding box annotation, applicable to both the weakly-supervised and semi-supervised settings.
2. We show that our approach achieves excellent performance when combining a small number of pixel-level annotated images with a large number of image-level or bounding box annotated images, nearly matching the results achieved when all training images have pixel-level annotations.
3. We show that combining weak or strong annotations across datasets yields further improvements. In particular, we reach 73.9% IOU performance on PASCAL VOC 2012 by combining annotations from the PASCAL and MS-COCO datasets.

2. Related work

Training segmentation models with only image-level labels has been a challenging problem in the literature [12, 37, 38, 40]. Our work is most related to other recent DCNN models such as [31, 32], who also study the weakly supervised setting. They both develop MIL-based algorithms for the problem. In contrast, our model employs an EM algorithm, which similarly to [26] takes into account the weak labels when inferring the latent image segmentations. Moreover, [32] proposed to smooth the prediction results by region proposal algorithms, *e.g.*, CPMC [3] and MCG [1], learned on pixel-segmented images. Neither [31, 32] cover the semi-supervised setting.

Bounding box annotations have been utilized for semantic segmentation by [39, 43], while [15, 21, 41] describe schemes exploiting both image-level labels and bounding box annotations. [4] attained human-level accuracy for car segmentation by using 3D bounding boxes. Bounding box annotations are also commonly used in interactive segmentation [22, 34]; we show that such foreground/background segmentation methods can effectively estimate object segments accurate enough for training a DCNN semantic segmentation system. Working in a setting very similar to ours, [9] employed MCG [1] (which requires training from pixel-level annotations) to infer object masks from bounding box labels during DCNN training.

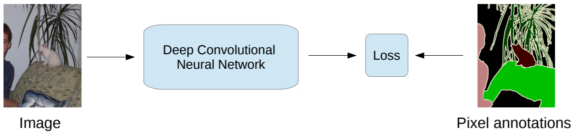

Figure 1. DeepLab model training from fully annotated images.

3. Proposed Methods

We build on the DeepLab model for semantic image segmentation proposed in [5]. This uses a DCNN to predict the label distribution per pixel, followed by a fully-connected (dense) CRF [19] to smooth the predictions while preserving image edges. In this paper, we focus for simplicity on methods for training the DCNN parameters from weak labels, only using the CRF at test time. Additional gains can be obtained by integrated end-to-end training of the DCNN and CRF parameters [42, 6].

Notation We denote by $\boldsymbol{x}$ the image values and $\boldsymbol{y}$ the segmentation map. In particular, $y_m \in \{0, \dots, L\}$ is the pixel label at position $m \in \{1, \dots, M\}$, assuming that we have the background as well as L possible foreground labels and M is the number of pixels. Note that these pixel-level labels may not be visible in the training set. We encode the set of image-level labels by $\boldsymbol{z}$, with $z_l = 1$, if the l-th label is present anywhere in the image, *i.e.*, if $\sum_m [y_m = l] > 0$.

3.1. Pixel-level annotations

In the fully supervised case illustrated in Fig. 1, the objective function is

$$J(\boldsymbol{\theta}) = \log P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}) = \sum_{m=1}^{M} \log P(y_m|\boldsymbol{x};\boldsymbol{\theta}) \,, \tag{1}$$

where $\boldsymbol{\theta}$ is the vector of DCNN parameters. The per-pixel label distributions are computed by

$$P(y_m|\boldsymbol{x};\boldsymbol{\theta}) \propto \exp(f_m(y_m|\boldsymbol{x};\boldsymbol{\theta})) \,, \tag{2}$$

where $f_m(y_m|\boldsymbol{x};\boldsymbol{\theta})$ is the output of the DCNN at pixel m. We optimize $J(\boldsymbol{\theta})$ by mini-batch SGD.

3.2. Image-level annotations

When only image-level annotation is available, we can observe the image values $\boldsymbol{x}$ and the image-level labels $\boldsymbol{z}$, but the pixel-level segmentations $\boldsymbol{y}$ are latent variables. We

Algorithm 1 Weakly-Supervised EM (fixed bias version)

```
Input: Initial CNN parameters $\boldsymbol{\theta}'$, potential parameters $b_l$, $l \in \{0, \dots, L\}$, image $\boldsymbol{x}$, image-level label set $\boldsymbol{z}$.
E-Step: For each image position $m$
    $\hat{f}_m(l) = f_m(l|\boldsymbol{x};\boldsymbol{\theta}') + b_l$, if $z_l = 1$
    $\hat{f}_m(l) = f_m(l|\boldsymbol{x};\boldsymbol{\theta}')$, if $z_l = 0$
    $\hat{y}_m = \text{argmax}_l \, \hat{f}_m(l)$
M-Step:
    $Q(\boldsymbol{\theta};\boldsymbol{\theta}') = \log P(\hat{\boldsymbol{y}}|\boldsymbol{x},\boldsymbol{\theta}) = \sum_{m=1}^{M} \log P(\hat{y}_m|\boldsymbol{x},\boldsymbol{\theta})$
    Compute $\nabla_{\boldsymbol{\theta}} Q(\boldsymbol{\theta};\boldsymbol{\theta}')$ and use SGD to update $\boldsymbol{\theta}'$.
```

have the following probabilistic graphical model:

$$P(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z};\boldsymbol{\theta}) = P(\boldsymbol{x}) \left(\prod_{m=1}^{M} P(y_m|\boldsymbol{x};\boldsymbol{\theta}) \right) P(\boldsymbol{z}|\boldsymbol{y}) \, . \tag{3}$$

We pursue an EM-approach in order to learn the model parameters $\boldsymbol{\theta}$ from training data. If we ignore terms that do not depend on $\boldsymbol{\theta}$, the expected complete-data log-likelihood given the previous parameter estimate $\boldsymbol{\theta}'$ is

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}') = \sum_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z};\boldsymbol{\theta}') \log P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}) \approx \log P(\hat{\boldsymbol{y}}|\boldsymbol{x};\boldsymbol{\theta}) \, , \tag{4}$$

where we adopt a hard-EM approximation, estimating in the E-step of the algorithm the latent segmentation by

$$\begin{aligned} \hat{\boldsymbol{y}} &= \underset{\boldsymbol{y}}{\text{argmax}} \, P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}')P(\boldsymbol{z}|\boldsymbol{y}) && (5) \\ &= \underset{\boldsymbol{y}}{\text{argmax}} \log P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}') + \log P(\boldsymbol{z}|\boldsymbol{y}) && (6) \\ &= \underset{\boldsymbol{y}}{\text{argmax}} \left(\sum_{m=1}^{M} f_m(y_m|\boldsymbol{x};\boldsymbol{\theta}') + \log P(\boldsymbol{z}|\boldsymbol{y}) \right) && (7) \end{aligned}$$

In the M-step of the algorithm, we optimize $Q(\boldsymbol{\theta};\boldsymbol{\theta}') \approx \log P(\hat{\boldsymbol{y}}|\boldsymbol{x};\boldsymbol{\theta})$ by mini-batch SGD similarly to (1), treating $\hat{\boldsymbol{y}}$ as ground truth segmentation.

To completely identify the E-step (7), we need to specify the observation model $P(\boldsymbol{z}|\boldsymbol{y})$. We have experimented with two variants, *EM-Fixed* and *EM-Adapt*.

EM-Fixed In this variant, we assume that $\log P(\boldsymbol{z}|\boldsymbol{y})$ factorizes over pixel positions as

$$\log P(\boldsymbol{z}|\boldsymbol{y}) = \sum_{m=1}^{M} \phi(y_m, \boldsymbol{z}) + (\text{const}) \, , \tag{8}$$

allowing us to estimate the E-step segmentation at each pixel separately

$$\hat{y}_m = \underset{y_m}{\text{argmax}} \, \hat{f}_m(y_m) \doteq f_m(y_m|\boldsymbol{x};\boldsymbol{\theta}') + \phi(y_m, \boldsymbol{z}) \, . \tag{9}$$

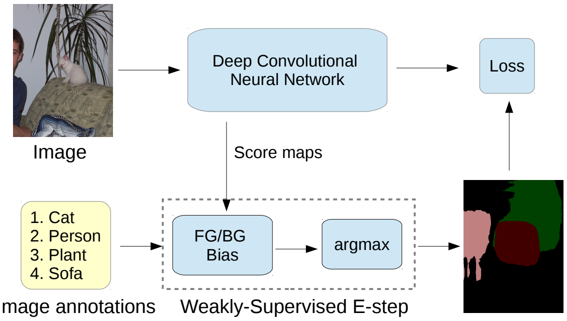

Figure 2. DeepLab model training using image-level labels.

We assume that

$$\phi(y_m = l, \boldsymbol{z}) = \begin{cases} b_l & \text{if } z_l = 1 \\ 0 & \text{if } z_l = 0 \end{cases} \tag{10}$$

We set the parameters $b_l = b_{\text{fg}}$, if $l > 0$ and $b_0 = b_{\text{bg}}$, with $b_{\text{fg}} > b_{\text{bg}} > 0$. Intuitively, this potential encourages a pixel to be assigned to one of the image-level labels $\boldsymbol{z}$. We choose $b_{\text{fg}} > b_{\text{bg}}$, boosting present foreground classes more than the background, to encourage full object coverage and avoid a degenerate solution of all pixels being assigned to background. The procedure is summarized in Algorithm 1 and illustrated in Fig. 2.

EM-Adapt In this method, we assume that $\log P(\boldsymbol{z}|\boldsymbol{y}) = \phi(\boldsymbol{y}, \boldsymbol{z}) + (\text{const})$, where $\phi(\boldsymbol{y}, \boldsymbol{z})$ takes the form of a cardinality potential [23, 33, 36]. In particular, we encourage *at least* a ρ_l portion of the image area to be assigned to class l, if $z_l = 1$, and enforce that no pixel is assigned to class l, if $z_l = 0$. We set the parameters $\rho_l = \rho_{\text{fg}}$, if $l > 0$ and $\rho_0 = \rho_{\text{bg}}$. Similar constraints appear in [10, 20].

In practice, we employ a variant of Algorithm 1. We adaptively set the image- and class-dependent biases b_l so as the prescribed proportion of the image area is assigned to the background or foreground object classes. This acts as a powerful constraint that explicitly prevents the background score from prevailing in the whole image, also promoting higher foreground object coverage. The detailed algorithm is described in the supplementary material.

EM vs. MIL It is instructive to compare our EM-based approach with two recent Multiple Instance Learning (MIL) methods for learning semantic image segmentation models [31, 32]. The method in [31] defines an MIL classification objective based on the per-class spatial maximum of the local label distributions of (2), $\hat{P}(l|\boldsymbol{x};\boldsymbol{\theta}) \doteq \max_m P(y_m = l|\boldsymbol{x};\boldsymbol{\theta})$, and [32] adopts a softmax function. While this approach has worked well for image classification tasks [29, 30], it is less suited for segmentation as it does not promote full object coverage: The DCNN becomes tuned to focus on the most distinctive object parts (*e.g.*, human face) instead of capturing the whole object (*e.g.*, human body).

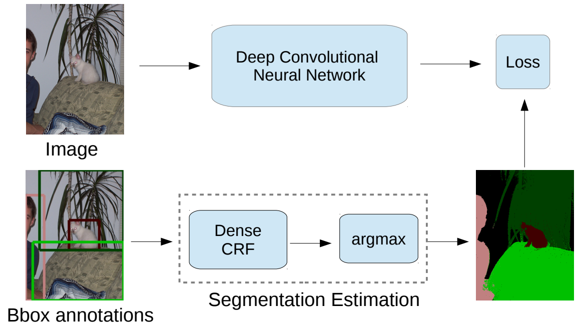

Figure 3. DeepLab model training from bounding boxes.

3.3. Bounding Box Annotations

We explore three alternative methods for training our segmentation model from labeled bounding boxes.

The first *Bbox-Rect* method amounts to simply considering each pixel within the bounding box as positive example for the respective object class. Ambiguities are resolved by assigning pixels that belong to multiple bounding boxes to the one that has the smallest area.

The bounding boxes fully surround objects but also contain background pixels that contaminate the training set with false positive examples for the respective object classes. To filter out these background pixels, we have also explored a second *Bbox-Seg* method in which we perform automatic foreground/background segmentation. To perform this segmentation, we use the same CRF as in DeepLab. More specifically, we constrain the center area of the bounding box ($\alpha\%$ of pixels within the box) to be foreground, while we constrain pixels outside the bounding box to be background. We implement this by appropriately setting the unary terms of the CRF. We then infer the labels for pixels in between. We cross-validate the CRF parameters to maximize segmentation accuracy in a small held-out set of fully-annotated images. This approach is similar to the grabcut method of [34]. Examples of estimated segmentations with the two methods are shown in Fig. 4.

The two methods above, illustrated in Fig. 3, estimate segmentation maps from the bounding box annotation as a pre-processing step, then employ the training procedure of Sec. 3.1, treating these estimated labels as ground-truth.

Our third *Bbox-EM-Fixed* method is an EM algorithm that allows us to refine the estimated segmentation maps throughout training. The method is a variant of the *EM-Fixed* algorithm in Sec. 3.2, in which we boost the present foreground object scores only within the bounding box area.

3.4. Mixed strong and weak annotations

In practice, we often have access to a large number of weakly image-level annotated images and can only afford to procure detailed pixel-level annotations for a small fraction of these images. We handle this hybrid training scenario by combining the methods presented in the previous sections, as illustrated in Figure 5. In SGD training of our deep CNN models, we bundle to each mini-batch a fixed proportion of strongly/weakly annotated images, and employ our EM algorithm in estimating at each iteration the latent semantic segmentations for the weakly annotated images.

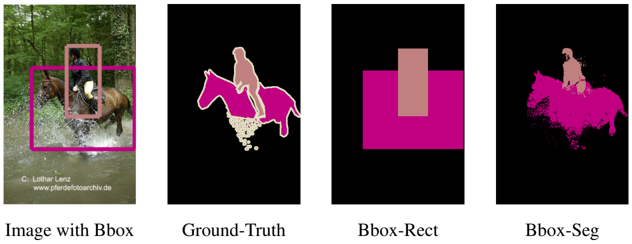

Figure 4. Estimated segmentation from bounding box annotation.

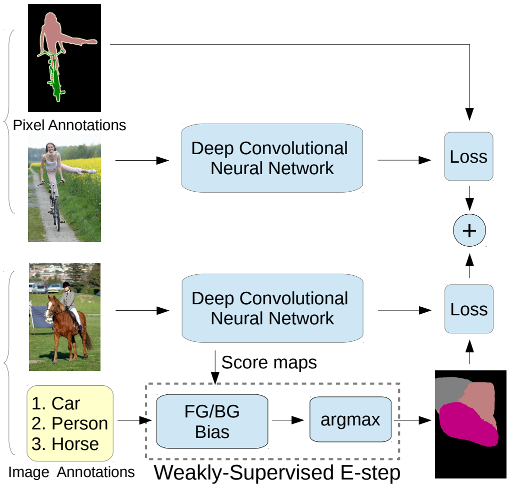

Figure 5. DeepLab model training on a union of full (strong labels) and image-level (weak labels) annotations.

4. Experimental Evaluation

4.1. Experimental Protocol

Datasets The proposed training methods are evaluated on the PASCAL VOC 2012 segmentation benchmark [13], consisting of 20 foreground object classes and one background class. The segmentation part of the original PASCAL VOC 2012 dataset contains 1464 (*train*), 1449 (*val*), and 1456 (*test*) images for training, validation, and test, respectively. We also use the extra annotations provided by [16], resulting in augmented sets of $10,582$ (*train_aug*) and $12,031$ (*trainval_aug*) images. We have also experimented with the large MS-COCO 2014 dataset [24], which contains $123,287$ images in its *trainval* set. The MS-COCO 2014 dataset has 80 foreground object classes and one background class and is also annotated at the pixel level.

The performance is measured in terms of pixel intersection-over-union (IOU) averaged across the 21 classes. We first evaluate our proposed methods on the PASCAL VOC 2012 *val* set. We then report our results on the official PASCAL VOC 2012 benchmark *test* set (whose annotations are not released). We also compare our *test* set results with other competing methods.

Reproducibility We have implemented the proposed methods by extending the excellent Caffe framework [18]. 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`.

Weak annotations In order to simulate the situations where only weak annotations are available and to have fair comparisons (e.g., use the same images for all settings), we generate the weak annotations from the pixel-level annotations. The image-level labels are easily generated by summarizing the pixel-level annotations, while the bounding box annotations are produced by drawing rectangles tightly containing each object instance (PASCAL VOC 2012 also provides instance-level annotations) in the dataset.

Network architectures We have experimented with the two DCNN architectures of [5], with parameters initialized from the VGG-16 ImageNet [11] pretrained model of [35]. They differ in the receptive field of view (FOV) size. We have found that large FOV (224×224) performs best when at least some training images are annotated at the pixel level, whereas small FOV (128×128) performs better when only image-level annotations are available. In the main paper we report the results of the best architecture for each setup and defer the full comparison between the two FOVs to the supplementary material.

Training We employ our proposed training methods to learn the DCNN component of the DeepLab-CRF model of [5]. For SGD, we use a mini-batch of 20-30 images and initial learning rate of 0.001 (0.01 for the final classifier layer), multiplying the learning rate by 0.1 after a fixed number of iterations. We use momentum of 0.9 and a weight decay of 0.0005. Fine-tuning our network on PASCAL VOC 2012 takes about 12 hours on a NVIDIA Tesla K40 GPU.

Similarly to [5], we decouple the DCNN and Dense CRF training stages and learn the CRF parameters by cross validation to maximize IOU segmentation accuracy in a held-out set of 100 Pascal *val* fully-annotated images. We use 10 mean-field iterations for Dense CRF inference [19]. Note that the IOU scores are typically 3-5% worse if we don't use the CRF for post-processing of the results.

4.2. Pixel-level annotations

We have first reproduced the results of [5]. Training the DeepLab-CRF model with strong pixel-level annotations on PASCAL VOC 2012, we achieve a mean IOU score of 67.6% on *val* and 70.3% on *test*; see method *DeepLab-CRF-LargeFOV* in [5, Table 1].

Method	#Strong	#Weak	val IOU
EM-Fixed (Weak)	-	10,582	20.8
EM-Adapt (Weak)	-	10,582	38.2
EM-Fixed (Semi)	200	10,382	47.6
	500	10,082	56.9
	750	9,832	59.8
	1,000	9,582	62.0
	1,464	5,000	63.2
	1,464	9,118	64.6
Strong	1,464	-	62.5
	10,582	-	67.6

Table 1. VOC 2012 *val* performance for varying number of pixel-level (strong) and image-level (weak) annotations (Sec. 4.3).

Method	#Strong	#Weak	test IOU
MIL-FCN [31]	-	10k	25.7
MIL-sppxl [32]	-	760k	35.8
MIL-obj [32]	BING	760k	37.0
MIL-seg [32]	MCG	760k	40.6
EM-Adapt (Weak)	-	12k	39.6
EM-Fixed (Semi)	1.4k	10k	66.2
	2.9k	9k	68.5
Strong [5]	12k	-	70.3

Table 2. VOC 2012 *test* performance for varying number of pixel-level (strong) and image-level (weak) annotations (Sec. 4.3).

4.3. Image-level annotations

Validation results We evaluate our proposed methods in training the DeepLab-CRF model using image-level weak annotations from the 10,582 PASCAL VOC 2012 *train_aug* set, generated as described in Sec. 4.1 above. We report the *val* performance of our two weakly-supervised EM variants described in Sec. 3.2. In the *EM-Fixed* variant we use $b_{\text{fg}} = 5$ and $b_{\text{bg}} = 3$ as fixed foreground and background biases. We found the results to be quite sensitive to the difference $b_{\text{fg}} - b_{\text{bg}}$ but not very sensitive to their absolute values. In the adaptive *EM-Adapt* variant we constrain at least $\rho_{\text{bg}} = 40\%$ of the image area to be assigned to background and at least $\rho_{\text{fg}} = 20\%$ of the image area to be assigned to foreground (as specified by the weak label set).

We also examine using weak image-level annotations in addition to a varying number of pixel-level annotations, within the semi-supervised learning scheme of Sec. 3.4. In this *Semi* setting we employ strong annotations of a subset of PASCAL VOC 2012 *train* set and use the weak image-level labels from another non-overlapping subset of the *train_aug* set. We perform segmentation inference for the images that only have image-level labels by means of *EM-Fixed*, which we have found to perform better than *EM-Adapt* in the semi-supervised training setting.

The results are summarized in Table 1. We see that the EM-Adapt algorithm works much better than the EM-Fixed algorithm when we only have access to image level annotations, 20.8% vs. 38.2% validation IOU. Using 1,464 pixel-level and 9,118 image-level annotations in the EM-Fixed semi-supervised setting significantly improves performance, yielding 64.6%. Note that image-level annotations are helpful, as training only with the 1,464 pixel-level annotations only yields 62.5%.

Test results In Table 2 we report our *test* results. We compare the proposed methods with the recent MIL-based approaches of [31, 32], which also report results obtained with image-level annotations on the VOC benchmark. Our EM-Adapt method yields 39.6%, which improves over MIL-FCN [31] by a large 13.9% margin. As [32] shows, MIL can become more competitive if additional segmentation information is introduced: Using low-level superpixels, MIL-sppxl [32] yields 35.8% and is still inferior to our EM algorithm. Only if augmented with BING [7] or MCG [1] can MIL obtain results comparable to ours (MIL-obj: 37.0%, MIL-seg: 40.6%) [32]. Note, however, that both BING and MCG have been trained with bounding box or pixel-annotated data on the PASCAL *train* set, and thus both MIL-obj and MIL-seg indirectly rely on bounding box or pixel-level PASCAL annotations.

The more interesting finding of this experiment is that including very few strongly annotated images in the semi-supervised setting significantly improves the performance compared to the pure weakly-supervised baseline. For example, using 2.9k pixel-level annotations along with 9k image-level annotations in the semi-supervised setting yields 68.5%. We would like to highlight that this result surpasses all techniques which are not based on the DCNN+CRF pipeline of [5] (see Table 6), even if trained with all available pixel-level annotations.

4.4. Bounding box annotations

Validation results In this experiment, we train the DeepLab-CRF model using bounding box annotations from the *train_aug* set. We estimate the training set segmentations in a pre-processing step using the *Bbox-Rect* and *Bbox-Seg* methods described in Sec. 3.3. We assume that we also have access to 100 fully-annotated PASCAL VOC 2012 *val* images which we have used to cross-validate the value of the single *Bbox-Seg* parameter α (percentage of the center bounding box area constrained to be foreground). We varied α from 20% to 80%, finding that $\alpha = 20\%$ maximizes accuracy in terms of IOU in recovering the ground truth foreground from the bounding box. We also examine the effect of combining these weak bounding box annotations with strong pixel-level annotations, using the semi-supervised learning methods of Sec. 3.4.

The results are summarized in Table 3. When using only bounding box annotations, we see that *Bbox-Seg* improves over *Bbox-Rect* by 8.1%, and gets within 7.0% of the strong pixel-level annotation result. We observe that combining 1,464 strong pixel-level annotations with weak bounding box annotations yields 65.1%, only 2.5% worse than the strong pixel-level annotation result. In the semi-supervised learning settings and 1,464 strong annotations, *Semi-Bbox-EM-Fixed* and *Semi-Bbox-Seg* perform similarly.

Method	#Strong	#Box	val IOU
Bbox-Rect (Weak)	-	10,582	52.5
Bbox-EM-Fixed (Weak)	-	10,582	54.1
Bbox-Seg (Weak)	-	10,582	60.6
Bbox-Rect (Semi)	1,464	9,118	62.1
Bbox-EM-Fixed (Semi)	1,464	9,118	64.8
Bbox-Seg (Semi)	1,464	9,118	65.1
Strong	1,464	-	62.5
	10,582	-	67.6

Table 3. VOC 2012 *val* performance for varying number of pixel-level (strong) and bounding box (weak) annotations (Sec. 4.4).

Method	#Strong	#Box	test IOU
BoxSup [9]	MCG	10k	64.6
BoxSup [9]	1.4k (+MCG)	9k	66.2
Bbox-Rect (Weak)	-	12k	54.2
Bbox-Seg (Weak)	-	12k	62.2
Bbox-Seg (Semi)	1.4k	10k	66.6
Bbox-EM-Fixed (Semi)	1.4k	10k	66.6
Bbox-Seg (Semi)	2.9k	9k	68.0
Bbox-EM-Fixed (Semi)	2.9k	9k	69.0
Strong [5]	12k	-	70.3

Table 4. VOC 2012 *test* performance for varying number of pixel-level (strong) and bounding box (weak) annotations (Sec. 4.4).

Test results In Table 4 we report our *test* results. We compare the proposed methods with the very recent BoxSup approach of [9], which also uses bounding box annotations on the VOC 2012 segmentation benchmark. Comparing our alternative Bbox-Rect (54.2%) and Bbox-Seg (62.2%) methods, we see that simple foreground-background segmentation provides much better segmentation masks for DCNN training than using the raw bounding boxes. BoxSup does 2.4% better, however it employs the MCG segmentation proposal mechanism [1], which has been trained with pixel-annotated data on the PASCAL *train* set; it thus indirectly relies on pixel-level annotations.

When we also have access to pixel-level annotated images, our performance improves to 66.6% (1.4k strong annotations) or 69.0% (2.9k strong annotations). In this semi-supervised setting we outperform BoxSup (66.6% *vs.* 66.2% with 1.4k strong annotations), although we do not use MCG. Interestingly, Bbox-EM-Fixed improves over Bbox-Seg as we add more strong annotations, and it performs 1.0% better (69.0% *vs.* 68.0%) with 2.9k strong annotations. This shows that the E-step of our EM algorithm can estimate the object masks better than the foreground-background segmentation pre-processing step when enough pixel-level annotated images are available.

Comparing with Sec. 4.3, note that 2.9k strong + 9k image-level annotations yield 68.5% (Table 2), while 2.9k strong + 9k bounding box annotations yield 69.0% (Table 3). This finding suggests that bounding box annotations add little value over image-level annotations when a sufficient number of pixel-level annotations is also available.

Method	#Strong COCO	#Weak COCO	val IOU
PASCAL-only	-	-	67.6
EM-Fixed (Semi)	-	123,287	67.7
Cross-Joint (Semi)	5,000	118,287	70.0
Cross-Joint (Strong)	5,000	-	68.7
Cross-Pretrain (Strong)	123,287	-	71.0
Cross-Joint (Strong)	123,287	-	71.7

Table 5. VOC 2012 *val* performance using strong annotations for all 10,582 *train_aug* PASCAL images and a varying number of strong and weak MS-COCO annotations (Sec. 4.5).

Method	test IOU
MSRA-CFM [8]	61.8
FCN-8s [25]	62.2
Hypercolumn [17]	62.6
TTI-Zoomout-16 [28]	64.4
DeepLab-CRF-LargeFOV [5]	70.3
BoxSup (Semi, with weak COCO) [9]	71.0
DeepLab-CRF-LargeFOV (Multi-scale net) [5]	71.6
Oxford_TVG_CRF_RNN_VOC [42]	72.0
Oxford_TVG_CRF_RNN_COCO [42]	74.7
Cross-Pretrain (Strong)	72.7
Cross-Joint (Strong)	73.0
Cross-Pretrain (Strong, Multi-scale net)	73.6
Cross-Joint (Strong, Multi-scale net)	73.9

Table 6. VOC 2012 *test* performance using PASCAL and MS-COCO annotations (Sec. 4.5).

4.5. Exploiting Annotations Across Datasets

Validation results We present experiments leveraging the 81-label MS-COCO dataset as an additional source of data in learning the DeepLab model for the 21-label PASCAL VOC 2012 segmentation task. We consider three scenarios:

- *Cross-Pretrain (Strong)*: Pre-train DeepLab on MS-COCO, then replace the top-level network weights and fine-tune on Pascal VOC 2012, using pixel-level annotation in both datasets.
- *Cross-Joint (Strong)*: Jointly train DeepLab on Pascal VOC 2012 and MS-COCO, sharing the top-level network weights for the common classes, using pixel-level annotation in both datasets.
- *Cross-Joint (Semi)*: Jointly train DeepLab on Pascal VOC 2012 and MS-COCO, sharing the top-level network weights for the common classes, using the pixel-level labels from PASCAL and varying the number of pixel- and image-level labels from MS-COCO.

In all cases we use strong pixel-level annotations for all 10,582 *train_aug* PASCAL images.

We report our results on the PASCAL VOC 2012 *val* in Table 5, also including for comparison our best PASCAL-only 67.6% result exploiting all 10,582 strong annotations as a baseline. When we employ the weak MS-COCO annotations (*EM-Fixed (Semi)*) we obtain 67.7% IOU, which does not improve over the PASCAL-only baseline. However, using strong labels from 5,000 MS-COCO images (4.0% of the MS-COCO dataset) and weak labels from the remaining MS-COCO images in the *Cross-Joint (Semi)* semi-supervised scenario yields 70.0%, a significant 2.4% boost over the baseline. This *Cross-Joint (Semi)* result is also 1.3% better than the 68.7% performance obtained using only the 5,000 strong and no weak annotations from MS-COCO. As expected, our best results are obtained by using all 123,287 strong MS-COCO annotations, 71.0% for *Cross-Pretrain (Strong)* and 71.7% for *Cross-Joint (Strong)*. We observe that cross-dataset augmentation improves by 4.1% over the best PASCAL-only result. Using only a small portion of pixel-level annotations and a large portion of image-level annotations in the semi-supervised setting reaps about half of this benefit.

Test results We report our PASCAL VOC 2012 *test* results in Table 6. We include results of other leading models from the PASCAL leaderboard. All our models have been trained with pixel-level annotated images on the PASCAL *trainval_aug* and the MS-COCO 2014 *trainval* datasets.

Methods based on the DCNN+CRF pipeline of DeepLab-CRF [5] are the most competitive, with performance surpassing 70%, even when only trained on PASCAL data. Leveraging the MS-COCO annotations brings about 2% improvement. Our top model yields 73.9%, using the multi-scale network architecture of [5]. Also see [42], which also uses joint PASCAL and MS-COCO training, and further improves performance (74.7%) by end-to-end learning of the DCNN and CRF parameters.

4.6. Qualitative Segmentation Results

In Fig. 6 we provide visual comparisons of the results obtained by the DeepLab-CRF model learned with some of the proposed training methods.

5. Conclusions

The paper has explored the use of weak or partial annotation in training a state of art semantic image segmentation model. Extensive experiments on the challenging PASCAL VOC 2012 dataset have shown that: (1) Using weak annotation solely at the image-level seems insufficient to train a high-quality segmentation model. (2) Using weak bounding-box annotation in conjunction with careful segmentation inference for images in the training set suffices to train a competitive model. (3) Excellent performance is obtained when combining a small number of pixel-level annotated images with a large number of weakly annotated images in a semi-supervised setting, nearly matching the results achieved when all training images have pixel-level annotations. (4) Exploiting extra weak or strong annotations from other datasets can lead to large improvements.

Acknowledgments

This work was partly supported by ARO 62250-CS, and NIH 5R01EY022247-03. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of GPUs used for this research.

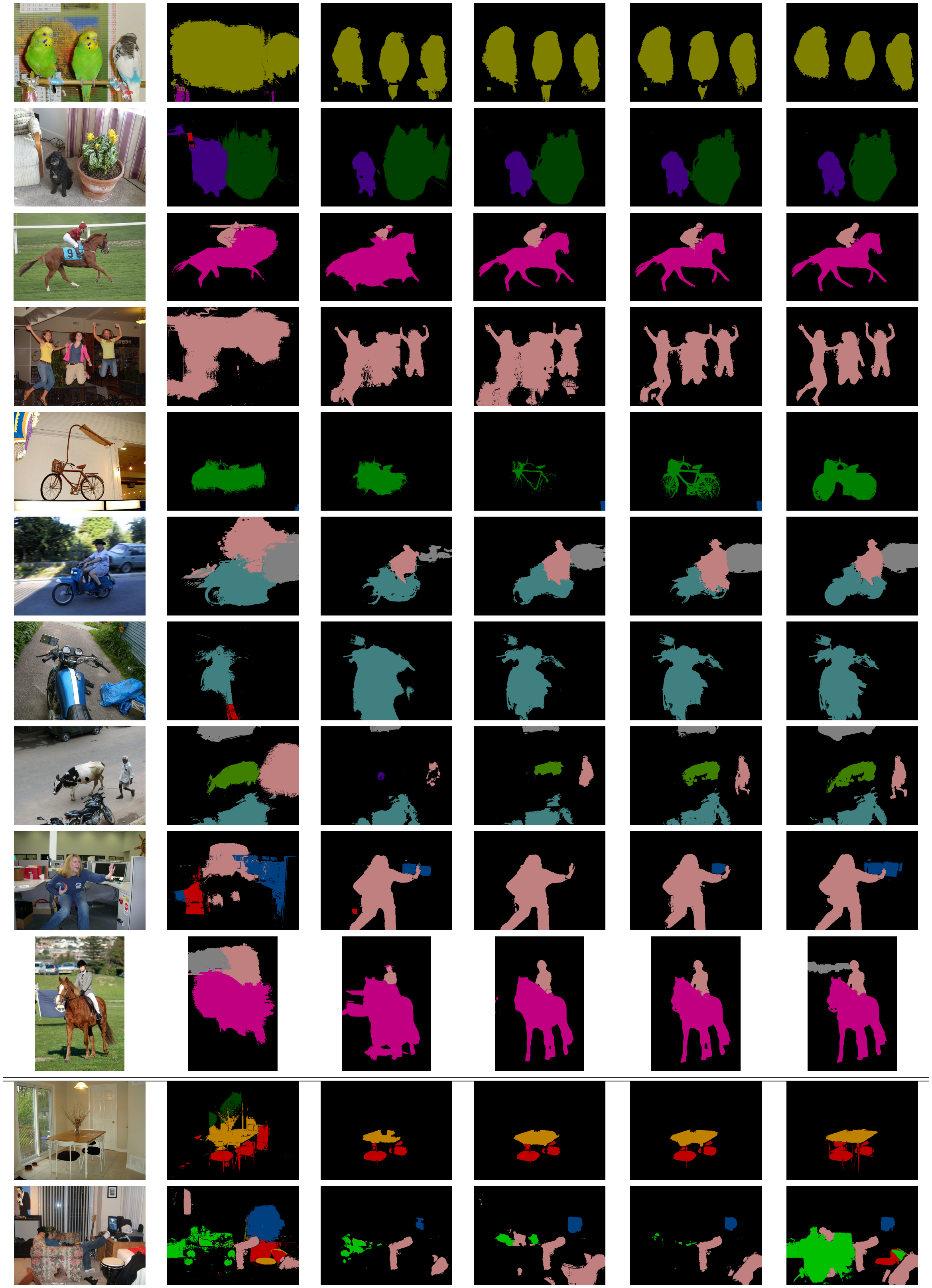

Figure 6. Qualitative DeepLab-CRF segmentation results on the PASCAL VOC 2012 *val* set. The last two rows show failure modes.

Supplementary Material

We include as appendix: (1) Details of the proposed EM-Adapt algorithm. (2) More experimental evaluations about the effect of the model's Field-Of-View. (3) More detailed results of the proposed training methods on PASCAL VOC 2012 test set.

A. E-Step with Cardinality Constraints: Details of our EM-Adapt Algorithm

Herein we provide more background and a detailed description of our *EM-Adapt* algorithm for weakly-supervised training with image-level annotations.

As a reminder, $\boldsymbol{y}$ is the latent segmentation map, with $y_m \in \{0, \dots, L\}$ denoting the label at position $m \in \{1, \dots, M\}$. The image-level annotation is encoded in $\boldsymbol{z}$, with $z_l = 1$, if the l-th label is present anywhere in the image.

We assume that $\log P(\boldsymbol{z}|\boldsymbol{y}) = \phi(\boldsymbol{y}, \boldsymbol{z}) + (\text{const})$. We employ a cardinality potential $\phi(\boldsymbol{y}, \boldsymbol{z})$ which encourages at least a ρ_l portion of the image area to be assigned to class l, if $z_l = 1$, and enforce that no pixel is assigned to class l, if $z_l = 0$. We set the parameters $\rho_l = \rho_{\text{fg}}$, if $l > 0$ and $\rho_0 = \rho_{\text{bg}}$.

While dedicated algorithms exist for optimizing energy functions under such cardinality potentials [36, 33, 23], we opt for a simpler alternative that approximately enforces these area constraints and works well in practice. We use a variant of the *EM-Fixed* algorithm described in the main paper, updating the segmentations in the E-Step by $\hat{y}_m = \text{argmax}_l \, \hat{f}_m(l) \doteq f_m(l|\boldsymbol{x}; \boldsymbol{\theta}') + b_l$. The key difference in the *EM-Adapt* variant is that the biases b_l are *adaptively* set so as the prescribed proportion of the image area is assigned to the background or foreground object classes that are present in the image.

When only one label l is present (*i.e.* $z_l = 1, \sum_{l'=0}^{L} z_{l'} = 1$), one can easily enforce the constraint that at least ρ_l of the image area is assigned to label l as follows: (1) Set $b_{l'} = 0, l' \neq l$. (2) Compute the maximum score at each position, $f_m^{\max} = \max_{l'=0}^{L} f_m(l'|\boldsymbol{x}; \boldsymbol{\theta}')$. (3) Set b_l equal to the ρ_l-th percentile of the score difference $d_m = f_m^{\max} - f_m(l|\boldsymbol{x}; \boldsymbol{\theta}')$. The cost of this algorithm is $\mathcal{O}(M)$ (linear w.r.t. the number of pixels).

When more than one labels are present in the image (*i.e.* $\sum_{l'=0}^{L} z_{l'} > 1$), we employ the procedure above sequentially for each label that $z_l > 1$ (we first visit the background label, then in random order each of the foreground labels which are present in the image). We set $b_l = -\infty$, if $z_l = 0$, to suppress the labels that are not present in the image.

An implementation of this algorithm will become publicly available after this paper gets published.

B. Effect of Field-Of-View

In this section, we explore the effect of Field-Of-View (FOV) when training the DeepLab-CRF model with the proposed methods in the main paper. Similar to [5], we also employ the 'atrous' algorithm [27] in the DeepLab model. The 'atrous' algorithm enables us to arbitrary control the model's FOV by adjusting the input stride (which is equivalent to injecting zeros between filter weights) at the first fully connected layer of VGG-16 net [35]. Applying a large value of input stride increases the effective kernel size, and thus enlarges the model's FOV (see [5] for details).

Experimental protocol We employ the same experimental protocol as the main paper. Models trained with the proposed training methods and different values of FOV are evaluated on the PASCAL VOC 2012 *val* set.

EM-Adapt Assuming only image-level labels are available, we first experiment with the *EM-Adapt (Weak)* method when the value of FOV varies. Specifically, we explore the setting where the kernel size is 3×3 with various FOV values. The selection of kernel size 3×3 is based on the discovery by [5]: employing a kernel size of 3×3 at the first fully connected layer can attain the same performance as using the kernel size of 7×7, while being 3.4 times faster during training. As shown in Table 7, we find that our proposed model can yield the performance of 39.2% with FOV 96×96, but the performance degrades by 9% when large FOV 224×224 is employed. The original DeepLab model employed by [5] has a kernel size of 4×4 and input stride of 4. Its performance, shown in the first row of Table 7, is similar to the performance obtained by using a kernel size of 3×3 and input stride of 6. Both cases have the same FOV value of 128×128.

Network architectures In the following experiments, we compare two network architectures trained with the methods proposed in the main paper. The first network is the same as the one originally employed by [5] (kernel size 4×4 and input stride 4, resulting in a FOV size 128×128). The second network we employ has FOV 224×224 (with kernel size of 3×3 and an input stride of 12). We refer to the first network as 'DeepLab-CRF with small FOV', and the second network as 'DeepLab-CRF with large FOV'.

Image-level annotations In Table 8, we experiment with the cases where weak image-level annotations as well as a varying number of pixel-level annotations are available. Similar to the results in Table 7, the DeepLab-CRF with small FOV performs better than that with large FOV when a small amount of supervision is leveraged. Interestingly, when there are more than 750 pixel-level annotations are

kernel size	input stride	receptive field	val IOU (%)
4×4	4	128×128	38.2
3×3	2	64×64	37.3
3×3	4	96×96	39.2
3×3	6	128×128	38.3
3×3	8	160×160	38.1
3×3	10	192×192	32.6
3×3	12	224×224	30.2

Table 7. Effect of Field-Of-View. The validation performance obtained by DeepLab-CRF trained with the method EM-Adapt (Weak) as the value of FOV varies.

Method	#Strong	#Weak	w Small FOV	w Large FOV
EM-Fixed (Weak)	-	10,582	20.8	19.9
EM-Adapt (Weak)	-	10,582	38.2	30.2
EM-Fixed (Semi)	200	10,382	47.6	38.9
	500	10,082	56.9	54.2
	750	9,832	58.8	59.8
	1,000	9,582	60.5	62.0
	1,464	5,000	60.5	63.2
	1,464	9,118	61.9	64.6
Strong	1,464	-	57.6	62.5
	10,582	-	63.9	67.6

Table 8. Effect of Field-Of-View. VOC 2012 *val* performance for varying number of pixel-level (strong) and image-level (weak) annotations (Sec. 4.3 of main paper).

available in the semi-supervised setting, employing large FOV yields better performance than using small FOV.

Bounding box annotations In Table 9, we report the results when weak bounding box annotations in addition to a varying number of pixel-level annotations are exploited. we found that DeepLab-CRF with small FOV attains better performance when trained with the three methods: Bbox-Rect (Weak), Bbox-EM-Fixed (Weak), and Bbox-Rect (Semi-1464 strong), whereas the model DeepLab-CRF with large FOV is better in all the other cases.

Annotations across datasets In Table 10, we show the results when training the models with the strong pixel-level annotations from PASCAL VOC 2012 *train_aug* set in conjunction with the extra annotations from MS-COCO [24] dataset (in the form of either weak image-level annotations or strong pixel-level annotations). Interestingly, employing large FOV consistently improves over using small FOV in all cases by at least 3%.

Main paper Note that in the main paper we report the results of the best architecture for each setup.

C. Detailed test results

In Table 11, Table 12, and Table 13, we show more detailed results on PASCAL VOC 2012 *test* set for all the reported methods in the main paper.

Method	#Strong	#Box	w Small FOV	w Large FOV
Bbox-Rect (Weak)	-	10,582	52.5	50.7
Bbox-EM-Fixed (Weak)	-	10,582	54.1	50.2
Bbox-Seg (Weak)	-	10,582	58.5	60.6
Bbox-Rect (Semi)	1,464	9.118	62.1	61.1
Bbox-EM-Fixed (Semi)	1,464	9,118	59.6	64.8
Bbox-Seg (Semi)	1,464	9,118	61.8	65.1
Strong	1,464	-	57.6	62.5
Strong	10,582	-	63.9	67.6

Table 9. Effect of Field-Of-View. VOC 2012 *val* performance for varying number of pixel-level (strong) and bounding box (weak) annotations (Sec. 4.4 of main paper).

Method	#Strong	#Weak	w Small FOV	w Large FOV
PASCAL-only	-	-	63.9	67.6
EM-Fixed (Semi)	-	123,287	64.4	67.7
Cross-Joint (Semi)	5,000	118,287	66.5	70.0
Cross-Joint (Strong)	5,000	-	64.9	68.7
Cross-Pretrain (Strong)	123,287	-	68.0	71.0
Cross-Joint (Strong)	123,287	-	68.0	71.7

Table 10. Effect of Field-Of-View. VOC 2012 *val* performance using strong annotations for all 10,582 *train_aug* PASCAL images and a varying number of strong and weak MS-COCO annotations (Sec. 4.5 of main paper).

References

[1] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014. 1, 2, 6

[2] S. Bell, P. Upchurch, N. Snavely, and K. Bala. Material recognition in the wild with the materials in context database. *arXiv:1412.0623*, 2014. 1

[3] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *PAMI*, 34(7):1312–1328, 2012. 2

[4] L.-C. Chen, S. Fidler, A. L. Yuille, and R. Urtasun. Beat the mturkers: Automatic image labeling from weak 3d supervision. In *CVPR*, 2014. 2

[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, 5, 6, 7, 9, 11

[6] L.-C. Chen, A. Schwing, A. Yuille, and R. Urtasun. Learning deep structured models. In *ICML*, 2015. 2

[7] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. H. S. Torr. BING: Binarized normed gradients for objectness estimation at 300fps. In *CVPR*, 2014. 1, 6

[8] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. *arXiv:1412.1283*, 2014. 1, 7, 11

[9] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. *arXiv:1503.01640*, 2015. 2, 6, 7, 11

[10] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. *IJCV*, 2012. 3

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

Method	bkg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
MIL-FCN [31]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	25.7
MIL-sppxl [32]	74.7	38.8	19.8	27.5	21.7	32.8	40.0	50.1	47.1	7.2	44.8	15.8	49.4	47.3	36.6	36.4	24.3	44.5	21.0	31.5	41.3	35.8
MIL-obj [32]	76.2	42.8	20.9	29.6	25.9	38.5	40.6	51.7	49.0	9.1	43.5	16.2	50.1	46.0	35.8	38.0	22.1	44.5	22.4	30.8	43.0	37.0
MIL-seg [32]	78.7	48.0	21.2	31.1	28.4	35.1	51.4	55.5	52.8	7.8	56.2	19.9	53.8	50.3	40.0	38.6	27.8	51.8	24.7	33.3	46.3	40.6
EM-Adapt (Weak)	76.3	37.1	21.9	41.6	26.1	38.5	50.8	44.9	48.9	16.7	40.8	29.4	47.1	45.8	54.8	28.2	30.0	44.0	29.2	34.3	46.0	39.6
EM-Fixed (Semi-1464 strong)	91.3	78.9	37.3	**81.4**	57.1	57.7	83.5	77.5	**77.6**	22.5	70.3	56.1	72.2	**74.3**	80.7	72.4	42.0	81.3	43.1	72.5	60.7	66.2
EM-Fixed (Semi-2913 strong)	**92.0**	**81.6**	**42.9**	80.5	**59.2**	**60.8**	**85.5**	**78.7**	77.3	**26.9**	**75.2**	**57.6**	**74.0**	74.2	**82.1**	**73.1**	**52.4**	**84.3**	**43.8**	**75.1**	**61.9**	**68.5**

Table 11. VOC 2012 *test* performance for varying number of pixel-level (strong) and image-level (weak) annotations (Sec. 4.3 of main paper). Links to the PASCAL evaluation server are included in the PDF.

Method	bkg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
BoxSup-box [9]	-	80.3	31.3	82.1	47.4	62.6	75.4	75.0	74.5	24.5	68.3	56.4	73.7	69.4	75.2	75.1	47.4	70.8	45.7	71.1	58.8	64.6
BoxSup-semi [9]	-	**82.0**	33.6	74.0	55.8	57.5	81.0	74.6	80.7	27.6	70.9	50.4	71.6	70.8	78.2	76.9	53.5	72.6	**50.1**	72.3	**64.4**	66.2
Bbox-Rect (Weak)	82.9	43.6	22.5	50.5	45.0	62.5	76.0	66.5	61.2	25.3	55.8	52.1	56.6	48.1	60.1	58.2	49.5	58.3	40.7	62.3	61.1	54.2
Bbox-Seg (Weak)	89.2	64.4	27.3	67.6	55.1	64.0	81.6	70.5	76.0	24.1	63.8	**58.2**	72.1	59.8	73.5	71.4	47.4	76.0	44.2	68.9	50.9	62.2
Bbox-Seg (Semi-1464 strong)	91.3	75.3	29.9	74.4	59.8	64.6	84.3	76.2	79.0	27.9	69.1	56.5	73.8	66.7	78.8	76.0	51.8	80.8	47.5	73.6	60.5	66.6
Bbox-EM-Fixed (Semi-1464 strong)	91.9	78.3	36.5	**86.2**	53.8	62.5	81.2	**80.0**	83.2	22.8	68.9	46.7	78.1	72.0	82.2	78.5	44.5	81.1	36.4	74.6	60.2	66.6
Bbox-Seg (Semi-2913 strong)	92.0	76.4	34.1	79.2	**61.0**	**65.6**	**85.0**	76.9	81.5	**28.5**	69.3	58.0	75.5	69.8	79.3	76.9	**54.0**	81.4	46.9	73.6	62.9	68.0
Bbox-EM-Fixed (Semi-2913 strong)	**92.5**	80.4	**41.6**	84.6	59.0	64.7	84.6	79.6	**83.5**	26.3	**71.2**	52.9	**78.3**	**72.3**	**83.3**	**79.1**	51.7	**82.1**	42.5	**75.0**	63.4	**69.0**

Table 12. VOC 2012 *test* performance for varying number of pixel-level (strong) and bounding box (weak) annotations (Sec. 4.4 of main paper). Links to the PASCAL evaluation server are included in the PDF.

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 [8]	-	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 [25]	-	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
Hypercolumn [17]	-	68.7	33.5	69.8	51.3	**70.2**	81.1	71.9	74.9	23.9	60.6	46.9	72.1	68.3	74.5	72.9	52.6	64.4	45.4	64.9	57.4	62.6
TTI-Zoomout-16 [28]	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
CRF_RNN [42]	-	80.9	34.0	72.9	52.6	62.5	79.8	76.3	79.9	23.6	67.7	51.8	74.8	69.9	76.9	76.9	49.0	74.7	42.7	72.1	59.6	65.2
DeepLab-CRF-LargeFOV [5]	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
Oxford_TVG_CRF_RNN [42]	-	85.5	36.7	77.2	62.9	66.7	85.9	78.1	82.5	30.1	74.8	59.2	77.3	75.0	82.8	79.7	**59.8**	78.3	50.0	76.9	65.7	70.4
BoxSup-semi-coco [9]	-	86.4	35.5	79.7	65.2	65.2	84.3	78.5	83.7	30.5	76.2	62.6	79.3	76.1	82.1	81.3	57.0	78.2	55.0	72.5	68.1	71.0
DeepLab-MSc-CRF-LargeFOV [5]	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
Oxford_TVG_CRF_RNN_COCO [42]	-	**90.4**	**55.3**	**88.7**	**68.4**	69.8	**88.3**	82.4	85.1	**32.6**	78.5	**64.4**	79.6	**81.9**	**86.4**	81.8	58.6	82.4	53.5	77.4	**70.1**	**74.7**
Cross-Pretrain (Strong)	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	72.7
Cross-Joint (Strong)	93.3	88.5	35.9	88.5	62.3	68.0	87.0	81.0	**86.8**	32.2	**80.8**	60.4	**81.1**	81.1	83.5	81.7	55.1	84.6	57.2	75.7	67.2	73.0
Cross-Pretrain (Strong, Multi-scale net)	**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	73.6
Cross-Joint (Strong, Multi-scale net)	93.7	89.2	46.7	88.5	63.5	68.4	87.0	81.2	86.3	**32.6**	80.7	62.4	81.0	81.3	84.3	82.1	56.2	84.6	**58.3**	76.2	67.2	73.9

Table 13. VOC 2012 *test* performance using strong PASCAL and strong MS-COCO annotations (Sec. 4.5 of main paper). Links to the PASCAL evaluation server are included in the PDF.

[12] P. Duygulu, K. Barnard, J. F. de Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, 2002. 2

[13] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserma. The pascal visual object classes challenge a retrospective. *IJCV*, 2014. 1, 4

[14] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *PAMI*, 2013. 1

[15] M. Guillaumin, D. Küttel, and V. Ferrari. Imagenet auto-annotation with segmentation propagation. *IJCV*, 110(3):328–348, 2014. 2

[16] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 4

[17] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. *arXiv:1411.5752*, 2014. 7, 11

[18] Y. Jia et al. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014. 5

[19] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011. 1, 2, 5

[20] H. Kuck and N. de Freitas. Learning about individuals from group statistics. In *UAI*, 2005. 3

[21] M. P. Kumar, H. Turki, D. Preston, and D. Koller. Learning specific-class segmentation from diverse data. In *ICCV*, 2011. 2

[22] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *ICCV*, 2009. 2

[23] Y. Li and R. Zemel. High order regularization for semi-supervised learning of structured output problems. In *ICML*, 2014. 3, 9

[24] T.-Y. Lin et al. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1, 4, 10

[25] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *arXiv:1411.4038*, 2014. 1, 7, 11

[26] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy. Learning to track and identify players from broadcast sports videos. *PAMI*, 2013. 2

[27] S. Mallat. *A Wavelet Tour of Signal Processing*. Acad. Press, 2 edition, 1999. 9

[28] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. *arXiv:1412.0774*, 2014. 1, 7, 11

[29] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Weakly supervised object recognition with convolutional neural networks. In *NIPS*, 2014. 3

[30] G. Papandreou, I. Kokkinos, and P.-A. Savalle. Untangling local and global deformations in deep convolutional networks for image classification and sliding window detection. *arXiv:1412.0296*, 2014. 3

[31] D. Pathak, E. Shelhamer, J. Long, and T. Darrell. Fully convolutional multi-class multiple instance learning. *arXiv:1412.7144*, 2014. 2, 3, 5, 6, 11

[32] P. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, 2015. 1, 2, 3, 5, 6, 11

[33] P. Pletscher and P. Kohli. Learning low-order models for enforcing high-order statistics. In *AISTATS*, 2012. 3, 9

[34] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: Interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004. 2, 4

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014. 5, 9

[36] D. Tarlow, K. Swersky, R. S. Zemel, R. P. Adams, and B. J. Frey. Fast exact inference for recursive cardinality models. In *UAI*, 2012. 3, 9

[37] J. Verbeek and B. Triggs. Region classification with markov field aspect models. In *CVPR*, 2007. 2

[38] A. Vezhnevets, V. Ferrari, and J. M. Buhmann. Weakly supervised structured output learning for semantic segmentation. In *CVPR*, 2012. 2

[39] W. Xia, C. Domokos, J. Dong, L.-F. Cheong, and S. Yan. Semantic segmentation without annotating segments. In *ICCV*, 2013. 2

[40] J. Xu, A. G. Schwing, and R. Urtasun. Tell me what you see and I will show you where it is. In *CVPR*, 2014. 2

[41] J. Xu, A. G. Schwing, and R. Urtasun. Learning to segment under various forms of weak supervision. In *CVPR*, 2015. 2

[42] 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. *arXiv:1502.03240*, 2015. 1, 2, 7, 11

[43] J. Zhu, J. Mao, and A. L. Yuille. Learning from weakly supervised data by the expectation loss svm (e-svm) algorithm. In *NIPS*, 2014. 2