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, 4, 6, 12, 22, 25, 38].

In this paper, we work with the DeepLab-CRF approach of [4, 38]. This combines a DCNN with a fully connected Conditional Random Field (CRF) [17], in order to get high resolution segmentations. This model achieves state-of-art results on the challenging PASCAL VOC segmentation benchmark [11], 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 [21], 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 [22] deliver significantly inferior results compared to their fully-supervised counterparts, only achieving 25.7%. Including additional trainable objectness [5] or segmentation [1] modules that largely increase the system complexity, [29] 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 [29] but
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 [10, 34, 35, 37]. Our work is most related to other recent DCNN models such as [28, 29], 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 [23] takes into account the weak labels when inferring the latent image segmentations. Moreover, [29] proposed to smooth the prediction results by region proposal algorithms (e.g., MCG [1]), learned on pixel-segmented images. Neither [28, 29] cover the semi-supervised setting.

Bounding box annotations have been utilized for semantic segmentation by [36, 39], while [13] describes a scheme exploiting both image-level labels and bounding box annotations in ImageNet [9]. [3] attained human-level accuracy for car segmentation by using 3D bounding boxes.

Bounding box annotations are also commonly used in interactive segmentation [19, 31]; 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, [7] employed MCG [1] (which requires training from pixel-level annotations) to infer object masks from bounding box labels during DCNN training.

3. Proposed Methods

We build on the DeepLab model for semantic image segmentation proposed in [4]. This uses a DCNN to predict the label distribution per pixel, followed by a fully-connected (dense) CRF [17] 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 [38].

Notation We denote by $x$ the image values and $y$ the segmentation map. In particular, $y_m \in \{0, \ldots, L\}$ is the pixel label at position $m \in \{1, \ldots, M\}$, 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 $z$, 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

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

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

$$P(y_m | x; \theta) \propto \exp(f_m(y_m | x; \theta)), \quad (2)$$

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

3.2. Image-level annotations

When only image-level annotation is available, we can observe the image values $x$ and the image-level labels $z$. 

![Figure 1. DeepLab model training from fully annotated images.](image-url)
Algorithm 1 Weakly-Supervised EM (fixed bias version)

Input: Initial CNN parameters \( \theta' \), potential parameters \( b_l \), 
\( l \in \{0, \ldots, L\} \), image \( x \), image-level label set \( z \).

E-Step: For each image position \( m \):
1. \( f_m(l) = f_m(l|x; \theta') + b_l \), if \( z_l = 1 \)
2. \( f_m(l) = f_m(l|x; \theta') \), if \( z_l = 0 \)
3. \( \hat{y}_m = \text{argmax}_y f_m(l) \)

M-Step:
4. \( Q(\theta; \theta') = \log P(\hat{y}|x; \theta) = \sum_{m=1}^{M} \log P(\hat{y}_m|x; \theta) \)
5. Compute \( \nabla_{\theta} Q(\theta; \theta') \) and use SGD to update \( \theta' \).

but the pixel-level segmentations \( y \) are latent variables. We have the following probabilistic graphical model:

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

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

\[
Q(\theta; \theta') = \sum_y P(y|x, z; \theta') \log P(y|x; \theta) \approx \log P(\hat{y}|x; \theta), \tag{4}
\]

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

\[
\hat{y} = \text{argmax}_y P(y|x; \theta') P(z|y) \tag{5}
\]

\[
= \text{argmax}_y \log P(y|x; \theta') + \log P(z|y) \tag{6}
\]

\[
= \text{argmax}_y \left( \sum_{m=1}^{M} f_m(y_m|x; \theta') + \log P(z|y) \right) \tag{7}
\]

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

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

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

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

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

\[
\hat{y}_m = \text{argmax}_{y_m} f_m(y_m) = f_m(y_m|x; \theta') + \phi(y_m, z). \tag{9}
\]

We assume that

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

We set the parameters \( b_l = b_{fg} \), if \( l > 0 \) and \( b_0 = b_{bg} \), with \( b_{fg} > b_{bg} > 0 \). Intuitively, this potential encourages a pixel to be assigned to one of the image-level labels \( z \). We choose \( b_{fg} > b_{bg} \), 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 \( \log P(z|y) = \phi(y, z) + \text{(const)} \), where \( \phi(y, z) \) takes the form of a cardinality potential [20, 30, 33]. In particular, we encourage at least a \( \rho_l \) 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 \( \rho_l = \rho_{fg} \), if \( l > 0 \) and \( \rho_0 = \rho_{bg} \). Similar area or label constraints appear in [8, 18].

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 [28, 29]. The method in [28] defines an MIL classification objective based on the per-class spatial maximum of the local label distributions of (2), \( \hat{P}(l|x; \theta) = \max_m P(y_m = l|x; \theta) \), and [29] adopts a softmax function. While this approach has worked well for image classification tasks [26, 27], 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).
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 (α% 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 [31]. 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.

4. Experimental Evaluation

4.1. Experimental Protocol

Datasets The proposed training methods are evaluated on the PASCAL VOC 2012 segmentation benchmark [11], 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 [14], 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 [21], 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 [16]. 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 [4], with parameters initialized from the VGG-16 ImageNet pretrained model of [32]. 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 [4]. In SGD training, 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 [4], 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 [17]. Note that the IOU scores are typically 3-5% worse if we don’t use the CRF for post-processing of the results.

<table>
<thead>
<tr>
<th>Method</th>
<th>#Strong</th>
<th>#Weak</th>
<th>val IOU</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM-Fixed (Weak)</td>
<td>-</td>
<td>10,582</td>
<td>29.8</td>
</tr>
<tr>
<td>EM-Adapt (Weak)</td>
<td>-</td>
<td>10,582</td>
<td>38.2</td>
</tr>
<tr>
<td>EM-Fixed (Semi)</td>
<td>200</td>
<td>10,382</td>
<td>47.6</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>10,082</td>
<td>56.9</td>
</tr>
<tr>
<td></td>
<td>750</td>
<td>9,832</td>
<td>59.8</td>
</tr>
<tr>
<td></td>
<td>1,000</td>
<td>9,582</td>
<td>62.0</td>
</tr>
<tr>
<td></td>
<td>1,464</td>
<td>5,000</td>
<td>63.2</td>
</tr>
<tr>
<td></td>
<td>1,464</td>
<td>9,118</td>
<td>64.6</td>
</tr>
<tr>
<td>Strong</td>
<td>1,464</td>
<td>-</td>
<td>62.5</td>
</tr>
<tr>
<td></td>
<td>10,582</td>
<td>-</td>
<td>67.6</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>#Strong</th>
<th>#Weak</th>
<th>test IOU</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIL-FCN [28]</td>
<td>-</td>
<td>10k</td>
<td>25.7</td>
</tr>
<tr>
<td>MIL-sppx [29]</td>
<td>-</td>
<td>760k</td>
<td>35.8</td>
</tr>
<tr>
<td>MIL-obj [29]</td>
<td>BING</td>
<td>760k</td>
<td>37.0</td>
</tr>
<tr>
<td>MIL-seg [29]</td>
<td>MCG</td>
<td>760k</td>
<td>40.6</td>
</tr>
<tr>
<td>EM-Adapt (Weak)</td>
<td>-</td>
<td>12k</td>
<td>39.6</td>
</tr>
<tr>
<td>EM-Adapt (Semi)</td>
<td>1.4k</td>
<td>10k</td>
<td>66.2</td>
</tr>
<tr>
<td></td>
<td>2.9k</td>
<td>9k</td>
<td>68.5</td>
</tr>
<tr>
<td>Strong [4]</td>
<td>12k</td>
<td>-</td>
<td>70.3</td>
</tr>
</tbody>
</table>

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

**Reproducibility** Upon acceptance, we plan to release our source code, configuration files, and trained models, to allow fully reproducing the results reported in this paper.

### 4.2. Pixel-level annotations

We have first reproduced the results of [4]. 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 [4, Table 1].

### 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 \( b_{fg} = 5 \) and \( b_{bg} = 3 \) as fixed foreground and background biases. We found the results to be quite sensitive to the difference \( b_{fg} - b_{bg} \) but not very sensitive to their absolute values. In the adaptive EM-Adapt variant we constrain at least \( \rho_{df} = 40\% \) of the image area to be assigned to background and at least \( \rho_{fg} = 20\% \) 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 a 1,464 pixel-level and 9,118 image-level annotations in the EM-Fixed semi-supervised setting significantly improves per-

formance, 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 [28, 29], 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 [28] by a large 13.9% margin. As [29] shows, MIL can become more competitive if additional segmentation in-

formation is introduced: Using low-level superpixels, MIL-sppx1 [29] yields 35.8% and is still inferior to our EM algorithm. Only if augmented with BING [5] or MCG [1] can MIL obtain results comparable to ours (MIL-obj: 37.0%, MIL-seg: 40.6%) [29]. 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 [4] (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 cen-

ter bounding box area constrained to be foreground). We varied α from 20% to 80%, finding that α = 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 annota-


tions 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.

Test results In Table 4 we report our test results. We compare the proposed methods with the very recent BoxSup approach of [7], 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 segmenta-

tion 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Strong</th>
<th>Box</th>
<th>val IOU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bbox-Rect (Weak)</td>
<td>-</td>
<td>10,582</td>
<td>52.5</td>
</tr>
<tr>
<td>Bbox-EM-Fixed (Weak)</td>
<td>-</td>
<td>10,582</td>
<td>54.1</td>
</tr>
<tr>
<td>Bbox-Seg (Weak)</td>
<td>-</td>
<td>10,582</td>
<td>60.6</td>
</tr>
<tr>
<td>Bbox-Rect (Semi)</td>
<td>1,464</td>
<td>9,118</td>
<td>62.1</td>
</tr>
<tr>
<td>Bbox-EM-Fixed (Semi)</td>
<td>1,464</td>
<td>9,118</td>
<td>64.8</td>
</tr>
<tr>
<td>Bbox-Seg (Semi)</td>
<td>1,464</td>
<td>9,118</td>
<td>65.1</td>
</tr>
<tr>
<td>Strong</td>
<td>1,464</td>
<td>10,582</td>
<td>62.5</td>
</tr>
</tbody>
</table>

Table 3. VOC 2012 val performance for varying number of pixel-

<table>
<thead>
<tr>
<th>Method</th>
<th>Strong</th>
<th>Box</th>
<th>test IOU</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoxSup [7]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCG</td>
<td></td>
<td>1.4k (+MCG)</td>
<td>66.2</td>
</tr>
<tr>
<td>Bbox-Rect (Weak)</td>
<td>-</td>
<td>12k</td>
<td>54.2</td>
</tr>
<tr>
<td>Bbox-Seg (Weak)</td>
<td>-</td>
<td>12k</td>
<td>62.2</td>
</tr>
<tr>
<td>Bbox-Seg (Semi)</td>
<td>1.4k</td>
<td>12k</td>
<td>66.6</td>
</tr>
<tr>
<td>Bbox-EM-Fixed (Semi)</td>
<td>1.4k</td>
<td>12k</td>
<td>66.6</td>
</tr>
<tr>
<td>Bbox-Seg (Semi)</td>
<td>2.9k</td>
<td>9k</td>
<td>68.0</td>
</tr>
<tr>
<td>Bbox-EM-Fixed (Semi)</td>
<td>2.9k</td>
<td>9k</td>
<td>69.0</td>
</tr>
<tr>
<td>Strong</td>
<td>1.4k</td>
<td>9k</td>
<td>70.3</td>
</tr>
</tbody>
</table>

Table 4. VOC 2012 test performance for varying number of pixel-

level (strong) and bounding box (weak) annotations (Sec. 4.4).
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.

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 3, 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 train\_val\_aug and the MS-COCO 2014 train\_val datasets.

Methods based on the DCNN+CRF pipeline of DeepLab-CRF [4] 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 [4]. This is the second best result reported on the PASCAL VOC 2012 segmentation task, only surpassed by [38], which also use joint training on PASCAL and MS-COCO data, but get improved performance (74.7%) due to 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
Figure 6. Qualitative DeepLab-CRF segmentation results on the PASCAL VOC 2012 val set. The last two rows show failure modes.
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.

**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, \( y \) is the latent segmentation map, with \( y_m \in \{0, \ldots, L\} \) denoting the label at position \( m \in \{1, \ldots, M\} \). The image-level annotation is encoded in \( z \), with \( z_l = 1 \), if the \( l \)-th label is present anywhere in the image.

We assume that \( \log P(z|y) = \phi(y, z) + \text{(const)} \). We employ a cardinality potential \( \phi(y, z) \) which encourages at least a \( \rho_l \) 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 \( \rho_l = \rho_{bg} \), if \( l > 0 \) and \( \rho_0 = \rho_{bg} \).

While dedicated algorithms exist for optimizing energy functions under such cardinality potentials [33, 30, 20], 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 \( \hat{y}_m = \arg\max_{l} \hat{f}_m(l) = f_m(l|x; \theta') + b_l \). 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 \( \rho_l \) 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, \( f_{m}^{\text{max}} = \max_{l'=0}^{L} f_m(l'|x; \theta') \). (3) Set \( b_l \) equal to the \( \rho_l \)-th percentile of the score difference \( d_m = f_{m}^{\text{max}} - f_m(l|x; \theta') \). The cost of this algorithm is \( O(M) \) (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 [4], we also employ the ‘atrous’ algorithm [24] 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 [32]. Applying a large value of input stride increases the effective kernel size, and thus enlarges the model’s FOV (see [4] 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 \( 3 \times 3 \) with various FOV values. The selection of kernel size \( 3 \times 3 \) is based on the discovery by [4]: employing a kernel size of \( 3 \times 3 \) at the first fully connected layer can attain the same performance as using the kernel size of \( 7 \times 7 \), 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 \( 96 \times 96 \), but the performance degrades by 9% when large FOV \( 224 \times 224 \) is employed. The original DeepLab model employed by [4] has a kernel size of \( 4 \times 4 \) 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 \( 3 \times 3 \) and input stride of 6. Both cases have the same FOV value of \( 128 \times 128 \).

**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 [4] (kernel size \( 4 \times 4 \) and input stride 4, resulting in a FOV size \( 128 \times 128 \)).
second network we employ has FOV $224 \times 224$ (with kernel size of $3 \times 3$ 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 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 [21] 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.

### References


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.

<table>
<thead>
<tr>
<th>Method</th>
<th>bkg</th>
<th>aero</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>chair</th>
<th>cow</th>
<th>table</th>
<th>dog</th>
<th>horse</th>
<th>mbike</th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIL-FCN (5)</td>
<td>74.7</td>
<td>38.8</td>
<td>19.8</td>
<td>27.5</td>
<td>21.7</td>
<td>32.8</td>
<td>40.0</td>
<td>50.1</td>
<td>47.1</td>
<td>7.2</td>
<td>44.8</td>
<td>15.8</td>
<td>49.4</td>
<td>47.3</td>
<td>36.6</td>
<td>36.4</td>
<td>24.3</td>
<td>44.5</td>
<td>21.0</td>
<td>31.5</td>
<td>41.3</td>
</tr>
<tr>
<td>MIL-sqdf (29)</td>
<td>76.2</td>
<td>42.8</td>
<td>20.9</td>
<td>29.6</td>
<td>25.9</td>
<td>38.5</td>
<td>40.6</td>
<td>51.7</td>
<td>49.0</td>
<td>9.1</td>
<td>43.5</td>
<td>16.2</td>
<td>50.1</td>
<td>46.0</td>
<td>35.8</td>
<td>38.0</td>
<td>22.1</td>
<td>44.5</td>
<td>22.4</td>
<td>30.8</td>
<td>43.0</td>
</tr>
<tr>
<td>MIL-sqg (29)</td>
<td>78.7</td>
<td>46.2</td>
<td>21.3</td>
<td>30.4</td>
<td>26.4</td>
<td>35.1</td>
<td>55.4</td>
<td>58.5</td>
<td>58.5</td>
<td>7.8</td>
<td>52.6</td>
<td>19.9</td>
<td>59.3</td>
<td>50.3</td>
<td>43.6</td>
<td>38.7</td>
<td>27.8</td>
<td>51.8</td>
<td>24.3</td>
<td>33.3</td>
<td>46.6</td>
</tr>
</tbody>
</table>

- **EM-Fixed (Semi-1404 strong)**
  - MIL-FCN (5): 91.3
  - MIL-sqdf (29): 92.6
  - MIL-sqg (29): 92.6

- **EM-Fixed (Semi-2913 strong)**
  - MIL-FCN (5): 91.5
  - MIL-sqdf (29): 92.3
  - MIL-sqg (29): 92.3

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>bkg</th>
<th>aero</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>chair</th>
<th>cow</th>
<th>table</th>
<th>dog</th>
<th>horse</th>
<th>mbike</th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoxSup-box [1]</td>
<td>-</td>
<td>80.3</td>
<td>31.3</td>
<td>82.1</td>
<td>47.8</td>
<td>62.6</td>
<td>75.4</td>
<td>75.0</td>
<td>74.5</td>
<td>24.3</td>
<td>68.8</td>
<td>56.3</td>
<td>73.7</td>
<td>69.4</td>
<td>75.2</td>
<td>75.1</td>
<td>47.4</td>
<td>30.8</td>
<td>45.7</td>
<td>71.8</td>
<td>58.8</td>
</tr>
<tr>
<td>BoxSup-sems [1]</td>
<td>82.0</td>
<td>33.6</td>
<td>74.4</td>
<td>55.8</td>
<td>57.5</td>
<td>61.0</td>
<td>74.6</td>
<td>80.7</td>
<td>72.6</td>
<td>37.9</td>
<td>50.4</td>
<td>71.6</td>
<td>70.8</td>
<td>78.2</td>
<td>76.9</td>
<td>53.5</td>
<td>72.6</td>
<td>50.1</td>
<td>72.1</td>
<td>64.4</td>
<td>66.2</td>
</tr>
</tbody>
</table>

- **EM-Adapt (Strong)**
  - MIL-FCN (5): 92.5
  - MIL-sqdf (29): 92.5
  - MIL-sqg (29): 92.5

- **EM-Adapt (Weak)**
  - MIL-FCN (5): 92.5
  - MIL-sqdf (29): 92.5
  - MIL-sqg (29): 92.5

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

- **Cross-Pretrain (Strong)**
  - MIL-FCN (5): 93.4
  - MIL-sqdf (29): 93.4
  - MIL-sqg (29): 93.4

- **Cross-Joint (Strong Multi-scale net)**
  - MIL-FCN (5): 93.7
  - MIL-sqdf (29): 93.7
  - MIL-sqg (29): 93.7

- **Cross-Joint (Strong Multi-scale net)**
  - MIL-FCN (5): 93.7
  - MIL-sqdf (29): 93.7
  - MIL-sqg (29): 93.7


[37] 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