Constrained Convolutional Neural Networks for Weakly Supervised Segmentation

Deepak Pathak Philipp Krähenbühl Trevor Darrell
University of California, Berkeley
{pathak,philkr,trevor}@cs.berkeley.edu

Abstract

We present an approach to learn a dense pixel-wise labeling from image-level tags. Each image-level tag imposes constraints on the output labeling of a Convolutional Neural Network (CNN) classifier. We propose Constrained CNN (CCNN), a method which uses a novel loss function to optimize for any set of linear constraints on the output space (i.e. predicted label distribution) of a CNN. Our loss formulation is easy to optimize and can be incorporated directly into standard stochastic gradient descent optimization. The key idea is to phrase the training objective as a biconvex optimization for linear models, which we then relax to nonlinear deep networks. Extensive experiments demonstrate the generality of our new learning framework. The constrained loss yields state-of-the-art results on weakly supervised semantic image segmentation. We further demonstrate that adding slightly more supervision can greatly improve the performance of the learning algorithm.

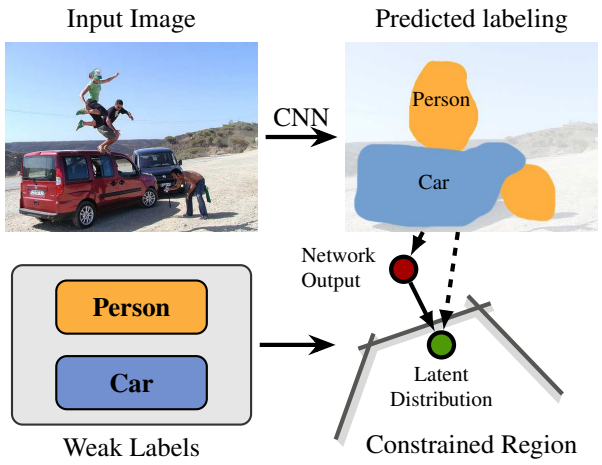


Figure 1: We train convolutional neural networks from a set of linear constraints on the output variables. The network output is encouraged to follow a latent probability distribution, which lies in the constraint manifold. The resulting loss is easy to optimize and can incorporate arbitrary linear constraints.

1. Introduction

In recent years, standard computer vision tasks, such as recognition or classification, have made tremendous progress. This is primarily due to the widespread adoption of Convolutional Neural Networks (CNNs) [11, 19, 20]. Existing models excel by their capacity to take advantage of massive amounts of fully supervised training data [28]. This reliance on full supervision is a major limitation on scalability with respect to the number of classes or tasks. For structured prediction problems, such as semantic segmentation, fully supervised, i.e. pixel-level, labels are both expensive and time consuming to obtain. Summarization of the semantic-labels in terms of weak supervision, e.g. image-level tags or bounding box annotations, is often less costly. Leveraging the full potential of these weak annotations is challenging, and existing approaches are susceptible to diverging into bad local optima from which recovery is difficult [6, 16, 25].

In this paper, we present a framework to incorporate weak supervision into the learning procedure through a series of linear constraints. In general, it is easier to express simple constraints on the output space than to craft regularizers or adhoc training procedures to guide the learning. In semantic segmentation, such constraints can describe the existence and expected distribution of labels from image-level tags. For example, given a car is present in an image, a certain number of pixels should be labeled as car.

We propose Constrained CNN (CCNN), a method which uses a novel loss function to optimize convolutional networks with arbitrary linear constraints on the structured output space of pixel labels. The non-convex nature of deep

The implementation code and trained models are available at the author's website.

nets makes a direct optimization of the constraints difficult. Our key insight is to model a distribution over latent "ground truth" labels while the output of the deep net follows this latent distribution as closely as possible. This allows us to enforce the constraints on the latent distribution instead of the network output, which greatly simplifies the resulting optimization problem. The resulting objective is a biconvex problem for linear models. For deep nonlinear models, it results in an alternating convex and gradient based optimization which can be naturally integrated into standard stochastic gradient descent (SGD). As illustrated in Figure 1, after each iteration the output is pulled towards the closest point on the constrained manifold of plausible semantic segmentation. Our Constrained CNN is guided by weak annotations and trained end-to-end.

We evaluate CCNN on the problem of multi-class semantic segmentation with varying levels of weak supervision defined by different linear constraints. Our approach achieves state-of-the-art performance on Pascal VOC 2012 compared to other weak learning approaches. It does not require pixel-level labels for any objects during the training time, but infers them directly from the image-level tags. We show that our constrained optimization framework can incorporate additional forms of weak supervision, such as a rough estimate of the size of an object. The proposed technique is general, and can incorporate many forms of weak supervision.

2. Related Work

Weakly supervised learning seeks to capture the signal that is common to all the positives but absent from all the negatives. This is challenging due to nuisance variables such as pose, occlusion, and intra-class variation. Learning with weak labels is often phrased as Multiple Instance Learning [8]. It is most frequently formulated as a maximum margin problem, although boosting [1, 36] and Noisy-OR models [15] have been explored as well. The multiple instance max-margin classification problem is non-convex and solved as an alternating minimization of a biconvex objective [2]. MI-SVM [2] or LSVM [10] are two classic methods in this paradigm. This setting naturally applies to weakly-labeled detection [31, 34]. However, most of these approaches are sensitive to the initialization of the detector [6]. Several heuristics have been proposed to address these issues [30, 31], however they are usually specific to detection.

Traditionally, the problem of weak segmentation and scene parsing with image level labels has been addressed using graphical models, and parametric structured models [32, 33, 37]. Most works exploit low-level image information to connect regions similar in appearance [32]. Chen *et al*. [5] exploit top-down segmentation priors based on visual subcategories for object discovery. Pinheiro *et al*. [26] and Pathak *et al*. [25] extend the multiple-instance learning framework from detection to semantic segmentation using CNNs. Their methods iteratively reinforce well-predicted outputs while suppressing erroneous segmentations contradicting image-level tags. Both algorithms are very sensitive to the initialization, and rely on carefully pretrained classifiers for all layers in the convolutional network. In contrast, our constrained optimization is much less sensitive and recovers a good solution from any random initialization of the classification layer.

Papandreou *et al*. [24] include an adaptive bias into the multi-instance learning framework. Their algorithm boosts classes known to be present and suppresses all others. We show that this simple heuristic can be viewed as a special case of a constrained optimization, where the adaptive bias controls the constraint satisfaction. However the constraints that can be modeled by this adaptive bias are limited and cannot leverage the full power of weak labels. In this paper, we show how to apply more general linear constraints which lead to better segmentation performance.

Constrained optimization problems have long been approximated by artificial neural networks [35]. These models are usually non-parametric, and solve just a single instance of a linear program. Platt *et al*. [27] show how to optimize equality constraints on the output of a neural network. However the resulting objective is highly non-convex, which makes a direct minimization hard. In this paper, we show how to optimize a constrained objective by alternating between a convex and gradient-based optimization.

The resulting algorithm is similar to generalized expectation [22] and posterior regularization [12] in natural language processing. Both methods train a parametric model that matches certain expectation constraints by applying a penalty to the objective function. Generalized expectation adds the expected constraint penalty directly to objective, which for convolutional networks is hard and expensive to evaluate directly. Ganchev *et al*. [12] constrain an auxiliary variable yielding an algorithm similar to our objective in dual space.

3. Preliminaries

We define a pixel-wise labeling for an image I as a set of random variables $X = \{x_0, \dots, x_n\}$ where n is the number of pixles in an image. $x_i \in \mathcal{L}$ takes one of m discrete labels $\mathcal{L} = \{1, \dots, m\}$. CNN models a probability distribution $Q(X|\theta, I)$ over those random variables, where θ are the parameters of the network. The distribution is commonly modeled as a product of independent marginals $Q(X|\theta, I) = \prod_i q_i(x_i|\theta, I)$, where each of the marginal represents a softmax probability:

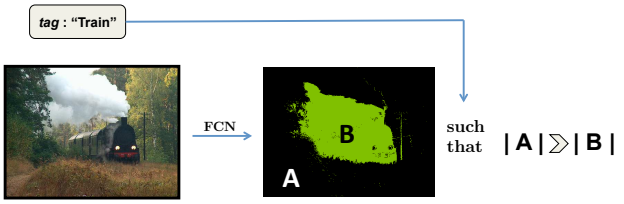


Figure 2: Overview of our weak learning pipeline. Input image is passed through a fully convolutional network (FCN) which produces an output labeling. The model is trained such that the output labeling follows a set of simple linear constraints imposed by image level tags.

$$q_i(x_i|\theta, I) = \frac{1}{Z_i} \exp\left(f_i(x_i; \theta, I)\right) \tag{1}$$

where $Z_i = \sum_{l \in \mathcal{L}} \exp\left(f_i(l; \theta, I)\right)$ is the partition function of a pixel i. The function f_i represents the real-valued score of the neural network. A higher score corresponds to a higher likelihood.

Standard learning algorithms aim to maximize the likelihood of the observed training data under the model. This requires full knowledge of the ground truth labeling, which is not available in the weakly supervised setting. In the next section, we show how to optimize the parameters of a CNN using some high-level constraints on the distribution of output labeling. An overview of this is given in Figure 2. In Section 5, we then present a few examples of useful constraints for weak labeling.

4. Constrained Optimization

For notational convenience, let $\vec{Q}_I$ be the vectorized form of network output $Q(X|\theta, I)$. The Constrained CNN (CCNN) optimization can be framed as:

$$\begin{aligned} \text{find} \quad & \theta \\ \text{subject to} \quad & A_I \vec{Q}_I \geq \vec{b}_I \qquad \forall I, \end{aligned} \tag{2}$$

where $A_I \in \mathbb{R}^{k \times nm}$ and $\vec{b}_I \in \mathbb{R}^k$ enforce k individual linear constraints on the output distribution of the convnet on image I. In theory, many outputs $\vec{Q}_I$ satisfy these constraints. However all network outputs are parametrized by a single parameter vector θ, which ties the output space of different $\vec{Q}_I$ together. In practice, this leads to an output that is both consistent with the input image and the constraints imposed by the weak labels.

For notational simplicity, we derive our inference algorithm for a single image with $A = A_I$, $\vec{b} = \vec{b}_I$ and $\vec{Q} = \vec{Q}_I$. The entire derivation generalizes to an arbitrary number of images and constraints. Constraints include for example lower and upper bounds on the expected number of foreground and background pixel labels in a scene. For more examples, see Section 5. In the first part of this section, we assume that all constraints are satisfiable, meaning there always exists a parameter setting θ such that $A\vec{Q} \geq \vec{b}$. In Section 4.3, we lift this assumption by adding slack variables to each of the constraints.

While problem (2) is convex in the network output Q, it is generally not convex with respect to the network parameters θ. For any non-linear function Q, the matrix A can be chosen such that the constraint is an upper or lower bound to Q, one of which is non-convex. This makes a direct optimization hard. As a matter of fact, not even log-linear models, such as logistic regression, can be directly optimized under this objective. Alternatively, one could optimize the Lagrangian dual of (2). However this is computationally very expensive, as we would need to optimize an entire convolutional neural network in an inner loop of a dual descent algorithm.

In order to efficiently optimize problem (2), we introduce a latent probability distribution $P(X)$ over the semantic labels X. We constrain $P(X)$ to lie in the feasibility region of the constrained objective while removing the constraints on the network output Q. We then encourage P and Q to model the same probability distribution by minimizing their respective KL-divergence. The resulting problem is defined as

$$\begin{aligned} \underset{\theta, P}{\text{minimize}} \quad & D\left(P(X) \| Q(X|\theta)\right) \\ \text{subject to} \quad & A\vec{P} \geq \vec{b}, \qquad \sum_X P(X) = 1, \end{aligned} \tag{3}$$

where $D\left(P(X) \| Q(X|\theta)\right) = \sum_X P(X) \log P(X) - \mathbb{E}_{X \sim P}\left[\log Q(X|\theta)\right]$ and $\vec{P}$ is the vectorized version of $P(X)$. If the constraints in (2) are satisfiable then the problems (2) and (3) are equivalent with a solution of (3) at P that is equal to the feasible Q. This equality implies that $P(X)$ can be modeled as a product of independent marginals $P(X) = \prod_i p_i(x_i)$ without loss of generality, with a minimum at $p_i(x_i) = q_i(x_i|\theta)$. A detailed proof is provided in the supplementary material.

The new objective is much easier to optimize, as it decouples the constraints from the network output. For fixed network parameters θ, the problem is convex in P. For a fixed latent distribution P, the problem reduces to a standard cross entropy loss which is optimized using stochastic gradient descent.

In the remainder of this section, we derive an algorithm to optimize problem (3) using block coordinate descent. Section 4.1 solves the constrained optimization for P while keeping the network parameters θ fixed. Section 4.2 then incorporates this optimization into standard stochastic gradient descent, keeping P fixed while optimizing for θ. Each

step is guaranteed to decrease the overall energy of problem (3), converging to a good local optimum. At the end of this section, we show how to handle constraints that are not directly satisfiable by adding a slack variable to the loss.

4.1. Latent distribution optimization

We first show how to optimize problem (3) with respect to P while keeping the convnet output fixed. The objective function is convex with linear constraints, which implies Slaters condition and hence strong duality holds as long as the constraints are satisfiable [3]. We can therefore optimize problem (3) by maximizing its dual function, i.e.,

$$\mathcal{L}(\lambda) = \lambda^\top \vec{b} - \sum_{i=1}^{n} \log \sum_{l \in \mathcal{L}} \exp\left(f_i(l;\theta) + A_{i;l}^\top \lambda\right), \quad (4)$$

where $\lambda \geq 0$ are the dual variables pertaining to the inequality constraints and $f_i(l;\theta)$ is the score of the convnet classifier for pixel i and label l. $A_{i;l}$ is the column of A corresponding to $p_i(l)$. A detailed derivation of this dual function is provided in the supplementary material.

The dual function is concave and can be optimized globally using projected gradient ascent [3]. The gradient of the dual function is given by $\frac{\partial}{\partial \lambda}\mathcal{L}(\lambda) = \vec{b} - A\vec{P}$, which results into

$$p_i(x_i) = \frac{1}{Z_i} \exp\left(f_i(x_i;\theta) + A_{i;x_i}^\top \lambda\right),$$

where $Z_i = \sum_l \exp\left(f_i(l;\theta) + A_{i;l}^\top \lambda\right)$ is the local partition function ensuring that the distribution $p_i(x_i)$ sums to one for $\forall x_i \in \mathcal{L}$. Intuitively, the projected gradient descent algorithm increases the dual variables for all constraints that are not satisfied. Those dual variables in turn adjust the distribution p_i to fulfill the constraints. The projected dual gradient descent algorithm usually converges within fewer than 50 iterations, making the optimization highly efficient.

Next, we show how to incorporate this estimate of $P(X)$ into the standard stochastic gradient descent algorithm.

4.2. SGD

For a fixed latent distribution P, problem (3) reduces to the standard cross entropy loss

$$L(\theta) = -\sum_i \sum_{x_i} p_i(x_i) \log q_i(x_i|\theta). \quad (5)$$

The gradient of this loss function is given by $\frac{\partial}{\partial \vec{f}_i(x_i)} L(\theta) = \vec{q}_i(x_i|\theta) - \vec{p}_i(x_i)$. For linear models, the loss function (5) is convex and can be optimized using any gradient based optimization. For multi-layer deep networks, we optimize it using back-propagation and stochastic gradient descent (SGD) with momentum, as implemented in Caffe [17].

Theoretically, we would need to keep the latent distribution P fixed for a few iterations of SGD until the objective value decreases. Otherwise, we are not strictly guaranteed that the overall objective (3) decreases. However, in practice we found inferring a new latent distribution at every step of SGD does not hurt the performance and leads to a faster convergence.

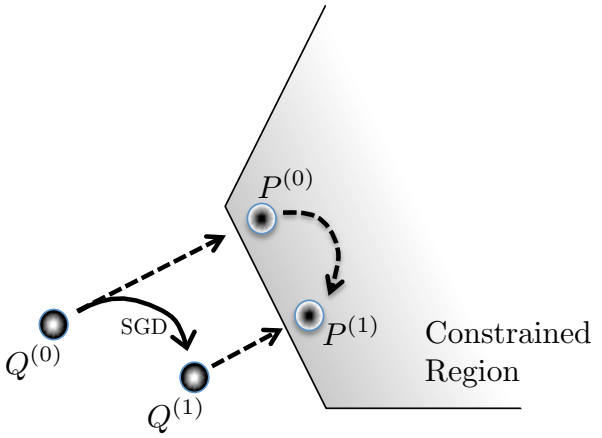


Figure 3: Illustration of our alternating convex optimization and gradient descent optimization for $t = 0$. At each iteration t, we compute a latent probability distribution $P^{(t)}$ as the closest point in the constrained region. We then update the convnet parameters to follow $P^{(t)}$ as closely as possible using Stochastic Gradient Descent (SGD), which takes the convnet output from $Q^{(t)}$ to $Q^{(t+1)}$.

In summary, we optimize problem (3) using SGD, where at each iteration we infer a latent distribution P which defines both our loss and loss gradient. Figure 3 shows an overview of the training procedure. For more details, see Section 6.

Up to this point, we assumed that all the constraints are simultaneously satisfiable. While this might hold for carefully chosen constraints, our optimization should be robust to arbitrary linear constraints. In the next section, we relax this assumption by adding a slack variable to the constraint set and show that this slack variable can then be easily integrated into the optimization.

4.3. Constraints with slack variable

We relax problem (3) by adding a slack $\xi \in \mathbb{R}^k$ to the linear constraints. The slack is regularized using a hinge loss with weight $\beta \in \mathbb{R}^k$. It results into the following optimization:

$$\begin{aligned} \underset{\theta,P,\xi}{\text{minimize}} \quad & D\left(P(X) \| Q(X|\theta)\right) + \beta^T \xi \\ \text{subject to} \quad & A\vec{P} \geq \vec{b} - \xi, \quad \sum_X P(X) = 1, \quad \xi \geq 0. \end{aligned} \quad (6)$$

This objective is now guaranteed to be satisfiable for any assignment to P and any linear constraint. Similar to (4), this is optimized using projected dual coordinate ascent. The dual objective function is exactly same as (4). The weighting term of the hinge loss β merely acts as an upper bound on the dual variable i.e. $0 \leq \lambda \leq \beta$. A detailed derivation of this loss is given in the supplementary material.

This slack relaxed loss allows the optimization to ignore certain constraints if they become too hard to enforce. It also trades off between various competing constraints.

5. Constraints for Weak Semantic Segmentation

We now describe all constraints we use for our weakly supervised semantic segmentation. For each training image I, we are given a set of image-level labels $\mathcal{L}_I$. Our constraints affect different parts of the output space depending on the image-level labels. All the constraints are complementary, and each constraint exploits the set of image-level labels differently.

Suppression constraint The most natural constraint is to suppress any label l that does not appear in the image.

$$\sum_{i=1}^{n} p_i(l) \leq 0 \qquad \forall\, l \notin \mathcal{L}_I. \tag{7}$$

This constraint alone is not sufficient, as a solution involving all background labels satisfies it perfectly. We can easily address this by adding a lower-bound constraint for labels present in an image.

Foreground constraint

$$a_l \leq \sum_{i=1}^{n} p_i(l) \qquad \forall\, l \in \mathcal{L}_I. \tag{8}$$

This foreground constraint is very similar to the commonly used multiple instance learning (MIL) paradigm, where at least one pixel is constrained to be positive [2, 16, 25, 26]. Unlike MIL, our foreground constraint can encourage multiple pixels to take a specific foreground label by increasing a_l. In practice, we set $a_l = 0.05n$ with a slack of $\beta = 2$, where n is the number of outputs of our network.

While this foreground constraint encourages some of the pixels to take a specific label, it is often not strong enough to encourage all pixels within an object to take the correct label. We could increase a_l to encourage more foreground labels, but this would over-emphasize small objects. A more natural solution is to constrain the total number of foreground labels in the output, which is equivalent to constraining the overall area of the background label.

Background constraint

$$a_0 \leq \sum_{i=1}^{n} p_i(0) \leq b_0. \tag{9}$$

Here $l = 0$ is assumed to be the background label. We apply both a lower and upper bound on the background label. This indirectly controls the minimum and maximum combined area of all foreground labels. We found $a_0 = 0.3n$ and $b_0 = 0.7n$ to work well in practice.

The above constraints are all complementary and ensure that the final labeling follows the image-level labels $\mathcal{L}_I$ as closely as possible. If we also have access to the rough size of an object, we can exploit this information during training. In our experiments, we show that substantial gains can be made by simply knowing if a certain object class covers more or less than 10% of the image.

Size constraint We exploit the size constraint in two ways: We boost all classes larger than 10% of the image by setting $a_l = 0.1n$. We also put an upper bound constraint on the classes l that are guaranteed to be small

$$\sum_{i=1}^{n} p_i(l) \leq b_l. \tag{10}$$

In practice, a threshold $b_l < 0.01n$ works slightly better than a tight threshold.

The EM-Adapt algorithm of Papandreou *et al*. [24] can be seen as a special case of a constrained optimization problem with just suppression and foreground constraints. The adaptive bias parameters then correspond to the Lagrangian dual variables λ of our constrained optimization. However in the original algorithm of Papandreou *et al*., the constraints are not strictly enforced especially when some of them conflict. In Section 7, we show that a principled optimization of those constraints, CCNN, leads to a substantial increase in performance.

6. Implementation Details

In this section, we discuss the overall pipeline of our algorithm applied for semantic image segmentation. We consider the weakly supervised setting i.e. only image-level labels are present during training. At test time, the task is to predict semantic segmentation mask for a given image.

Learning The CNN architecture used in our experiments is derived from VGG 16-layer network [29]. It was pretrained on Imagenet 1K class dataset, and achieved winning performance on ILSVRC14. We cast the fully connected layers into convolutions in a similar fashion as suggested in [21], and the last fc8 layer with 1K outputs is replaced by that containing 21 outputs corresponding to 20

Method	bgnd	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
MIL-FCN [25]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	24.9
MIL-Base [26]	37.0	10.4	12.4	10.8	05.3	05.7	25.2	21.1	25.2	04.8	21.5	08.6	29.1	25.1	23.6	25.5	12.0	28.4	08.9	22.0	11.6	17.8
MIL-Base w/ ILP [26]	**73.2**	25.4	**18.2**	22.7	21.5	28.6	39.5	44.7	46.6	11.9	**40.4**	11.8	**45.6**	**40.1**	35.5	35.2	20.8	**41.7**	17.0	34.7	30.4	32.6
EM-Adapt w/o CRF [24]	65.3	28.2	16.9	27.4	21.1	28.1	45.4	40.5	42.3	13.2	32.1	23.3	38.7	32.0	39.9	31.3	22.7	34.2	22.8	37.0	30.0	32.0
EM-Adapt [24]	67.2	**29.2**	17.6	**28.6**	**22.2**	29.6	**47.0**	44.0	44.2	14.6	35.1	**24.9**	41.0	34.8	41.6	32.1	24.8	37.4	**24.0**	38.1	31.6	33.8
CCNN w/o CRF	66.3	24.6	17.2	24.3	19.5	34.4	45.6	44.3	44.7	14.4	33.8	21.4	40.8	31.6	42.8	39.1	28.8	33.2	21.5	37.4	34.4	33.3
CCNN	68.5	25.5	18.0	25.4	20.2	**36.3**	46.8	**47.1**	**48.0**	**15.8**	37.9	21.0	44.5	34.5	**46.2**	**40.7**	**30.4**	36.3	22.2	**38.8**	**36.9**	**35.3**

Table 1: Comparison of weakly supervised semantic segmentation methods on PASCAL VOC 2012 validation set.

object classes in Pascal VOC and background class. The overall network stride of this fully convolutional network is 32s. However, we observe that the slightly modified architecture with the denser 8s network stride proposed in [4] gives better results in the weakly supervised training. Unlike [25, 26], we do not learn any weights of the last layer from Imagenet. Apart from the initial pre-training, all parameters are finetuned only on Pascal VOC. We initialize the weights of the last layer with random Gaussian noise.

The FCN takes in arbitrarily sized images and produces coarse heatmaps corresponding to each class in the dataset. We apply the convex constrained optimization on these coarse heatmaps, reducing the computational cost. The network is trained using SGD with momentum. We follow [21] and train our models with a batch size of 1, momentum of 0.99 and an initial learning rate of $1e$-6. We train for 60000 iterations, which corresponds to roughly 5 epochs. The learning rate is decreased by a factor of 0.1 every 20000 iterations. We found this setup to outperform a batch size of 20 with momentum of 0.9 [4]. The constrained optimization for single image takes less than 30 ms on a CPU single core, and could be accelerated using a GPU. The total training time is 8-9 hrs, comparable to [21, 24].

Inference At inference time, we optionally apply a fully connected conditional random field model [18] to refine the final segmentation. We used the default parameter provided by the authors for all our experiments.

7. Experiments

We analyze and compare the performance of our constrained optimization for varying levels of supervision: image-level tags and additional supervision such as object size information. The objective is to learn models to predict dense multi-class semantic segmentation i.e. pixel-wise labeling for any new image. We use the provided supervision with few simple spatial constraints on the output, and don't use any additional low-level graph-cut based methods in training. The goal is to demonstrate the strength of training with constrained outputs, and how it helps with increasing levels of supervision.

7.1. Dataset

We evaluate CCNNs for the task of semantic image segmentation on PASCAL VOC dataset [9]. The dataset contains pixel-level labels for 20 object classes and a separate background class. For a fair comparison to prior work, we use the similar setup to train all models. Training is performed on the union of VOC 2012 train set and the larger dataset collected by Hariharan *et al*. [13] summing upto a total of 10,582 training images. The VOC12 validation set containing a total of 1449 images is kept held-out during ablation studies. The VGG network architecture used in our algorithm was pre-trained on ILSVRC dataset [28] for classification task of 1K classes [29].

Results are reported in the form of standard intersection over union (IoU) metric, also known as Jaccard Index. It is defined per class as the percentage of pixels predicted correctly out of total pixels labeled or classified as that class. Ablation studies and comparison with baseline methods for both the weak settings are presented in the following subsections.

7.2. Training from image-level tags

We start by training our model using just image-level tags. We obtain these tags from the presence of a class in the pixel-wise ground truth segmentation masks. The constraints used in this setting are described in Equations (7), (8) and (9). Since some of the baseline methods report results on the VOC12 validation set, we present the performance on both validation and test set. Some methods boost their performance by using a Dense CRF model [18] to post process the final output labeling. To allow for a fair comparison, we present results both with and without a Dense CRF.

Table 1 compares all contemporary weak segmentation methods. Our proposed method, CCNN, outperforms all prior methods for weakly labeled semantic segmentation by a significant margin. MIL-FCN [25] is an extension of learning based on maximum scoring instance based MIL to multi-class segmentation. The algorithm proposed by Pinheiro *et al*. [26] introduces a soft version of MIL. It is trained on 0.7 million images for 21 classes taken from

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
Fully Supervised:																					
SDS [14]	63.3	25.7	63.0	39.8	59.2	70.9	61.4	54.9	16.8	45.0	48.2	50.5	51.0	57.7	63.3	31.8	58.7	31.2	55.7	48.5	51.6
FCN-8s [21]	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
TTIC Zoomout [23]	81.9	35.1	78.2	57.4	56.5	80.5	74.0	79.8	22.4	69.6	53.7	74.0	76.0	76.6	68.8	44.3	70.2	40.2	68.9	55.3	64.4
DeepLab-CRF [4]	78.4	33.1	78.2	55.6	65.3	81.3	75.5	78.6	25.3	69.2	52.7	75.2	69.0	79.1	77.6	54.7	78.3	45.1	73.3	56.2	**66.4**
Weakly Supervised:																					
CCNN w/ tags	24.2	19.9	26.3	18.6	38.1	51.7	42.9	48.2	15.6	37.2	18.3	43.0	38.2	52.2	40.0	33.8	36.0	21.6	33.4	38.3	35.6
CCNN w/ size	36.7	23.6	47.1	30.2	40.6	59.5	54.3	51.9	15.9	43.3	34.8	48.2	42.5	59.2	43.1	35.5	45.2	31.4	46.2	42.2	43.3
CCNN w/ size (CRF tuned)	42.3	24.5	56.0	30.6	39.0	58.8	52.7	54.8	14.6	48.4	34.2	52.7	46.9	61.1	44.8	37.4	48.8	30.6	47.7	41.7	**45.1**

Table 2: Results on PASCAL VOC 2012 test. We compare our results to the fully supervised state-of-the-art methods.

ILSVRC13, which is 70 times more data than all other approaches used. They achieve boost in performance by re-ranking the pixel probabilities with the image-level priors (ILP) i.e. the probability of class to be present in the image. This suppresses the negative classes and smooths out the predicted segmentation mask. For the EM-Adapt [24] algorithm, we reproduced the models using their publicly available implementation[1]. We apply similar set of constraints on EM-Adapt to make sure it is purely a comparison of the approach. Note that unconstrained MIL based approach require the final 21-class classifier to be well initialized for reasonable performance. While our constrained optimization can handle arbitrary random initializations.

We also directly compare our algorithm against the EM-Adapt results as reported in Papandreou *et al*. [24] for weak segmentation. However, their training procedure uses random crops of both the original image and the segmentation mask. The weak labels are then computed from those random crops. This introduces limited information about the spatial location of the weak tags. Taken to the extreme, a 1×1 output crop reduces to full supervision. We thus present this result in the next subsection on incorporating increasing supervision.

7.3. Training with additional supervision

We now consider slightly more supervision than just the image-level tags. Firstly, we consider the training with tags on random crops of original image, following Papandreou *et al*. [24]. We evaluate our constrained optimization on the EM-Adapt architecture using random crops, and compare to the result obtained from their released caffemodel as shown in Table 3. Using limited spatial information our algorithm slightly outperforms EM-Adapt, mainly due to the more powerful background constraints. Note that the difference is not as striking as in the pure weak label setting. We believe this is due to the fact that the spatial information in combination with the foreground prior emulates the upper bound constraint on background, as a random crop is likely to contain much fewer labels.

Method	Training Supervision	mIoU w/o CRF	mIoU
EM-Adapt [24]	Tags w/ random crops	34.3	36.0
CCNN	Tags w/ random crops	**34.4**	**36.4**
EM-Adapt [24]	Tags w/ object sizes	–	–
CCNN	Tags w/ object sizes	**40.5**	**42.4**

Table 3: Results using additional supervision during training evaluated on the VOC 2012 validation set.

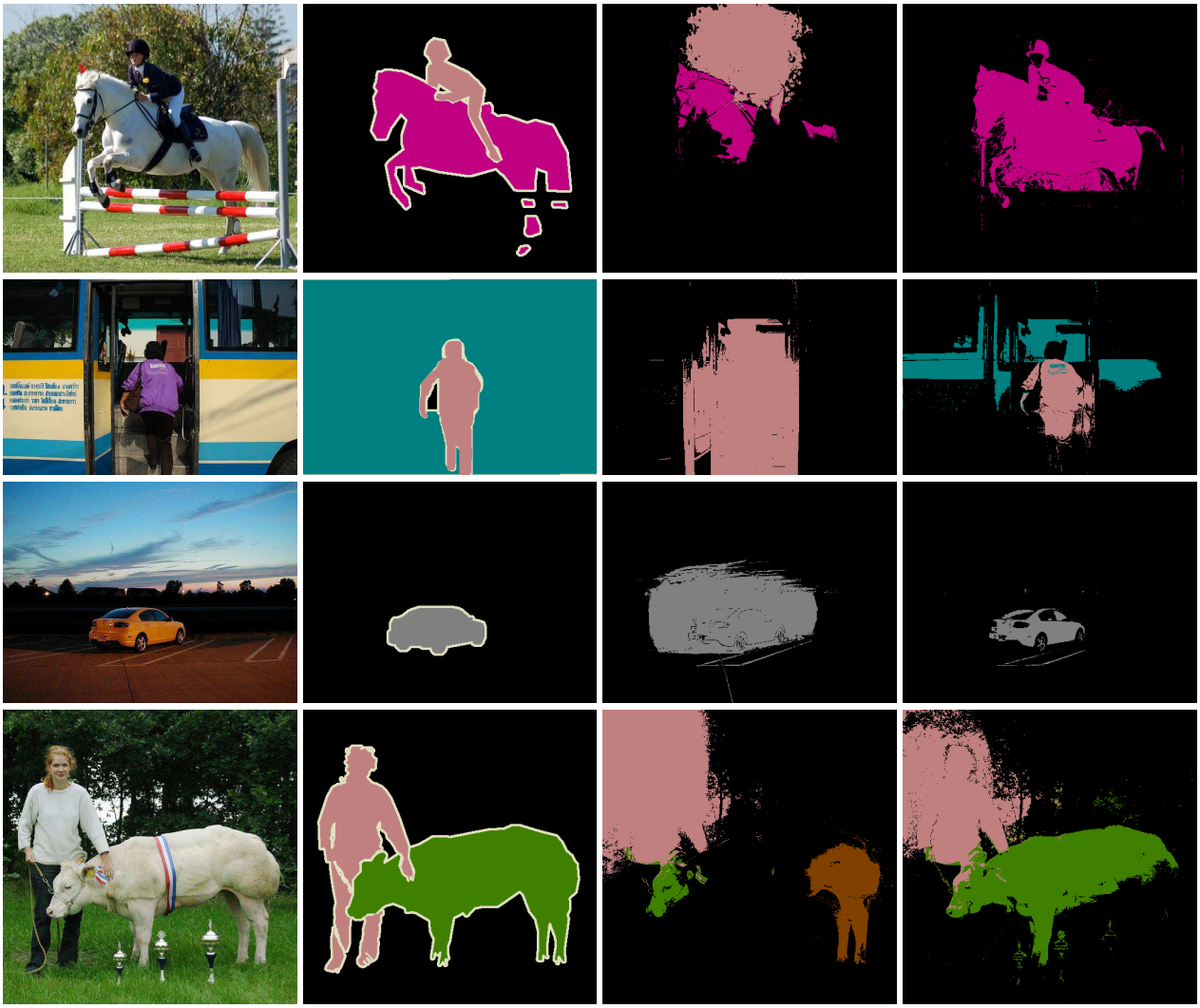
The main advantage of CCNN is that there is no restriction of the type of linear constraints that can be used. To demonstrate this further, we incorporate a simple size constraint. For each label, we use one additional bit of information: whether a certain class occupies more than 10% of the image or not. This additional constraint is described in Equation (10). As shown in Table 3, using this one additional bit of information dramatically increases the accuracy. Unfortunately, EM-Adapt heuristic cannot directly incorporate this more meaningful size constraint.

Table 2 reports our results on PASCAL VOC 2012 test server and compares it to fully supervised approaches. To better compare with these methods, we further add a result where the CRF parameters are tuned on 100 validation images. As a final experiment, we gradually add fully supervised images in addition to our weak objective and evaluate the model, i.e., semi-supervised learning. The graph is shown in the supplementary material. Our model makes good use of the additional supervision.

We also evaluate the sensitivity of our model to the parameters of the constraints. We performed line search along each of the bounds while keeping others fixed. In general, our method is very insensitive to wide range of constraint bounds due to the presence of slack variables. The standard deviation in accuracy, averaged over all parameters, is 0.73%. Details are provided in the supplementary material.

Qualitative results are shown in Figure 4.

[1] https://bitbucket.org/deeplab/deeplab-public/overview



(a) Original image (b) Ground truth (c) Image tags (d) Image tags + size

Figure 4: Qualitative results on the VOC 2012 dataset for different levels of supervision. We show the original image, ground truth, our trained classifier with image level tags and with size constraints. Note that the size constraints localize the objects much better than just image level tags at the cost of missing small objects in few examples.

7.4. Discussion

We further experimented with bounding box constraints. We constrain 75% of pixels within a bounding box to take a specific label, while we suppress any labels outside the bounding box. This additional supervision allows us to boost the IoU accuracy to 54%. This number is competitive with a baseline for which we train a model on all pixels within a bounding box, which gives 52.3% [24]. However it is not yet competitive with more sophisticated systems that use more segmentation information within bounding boxes [7, 24]. Those systems perform at roughly $58.5 - 62.0\%$ IoU accuracy. We believe the key to this performance is a stronger use of the pixel level segmentation information.

In conclusion, we presented CCNN which is a constrained optimization framework to optimize convolutional networks. The framework is general and can incorporate arbitrary linear constraints. It naturally integrates into standard Stochastic Gradient Descent, and can easily be used in publicly available frameworks such as Caffe [17].

We showed that constraints are a natural way to describe the desired output space of a labeling and can reduce the amount of strong supervision CNNs require.

Acknowledgments This work was supported by DARPA, AFRL, DoD MURI award N000141110688, NSF awards IIS-1427425 and IIS-1212798, and the Berkeley Vision and Learning Center.

References

[1] K. Ali and K. Saenko. Confidence-rated multiple instance boosting for object detection. In *CVPR*, 2014. 2

[2] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, 2002. 2, 5

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 4

[4] 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. 6, 7

[5] X. Chen, A. Shrivastava, and A. Gupta. Enriching visual knowledge bases via object discovery and segmentation. In *CVPR*, 2014. 2

[6] R. G. Cinbis, J. Verbeek, C. Schmid, et al. Multi-fold mil training for weakly supervised object localization. In *CVPR*, 2014. 1, 2

[7] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. *arXiv preprint arXiv:1503.01640*, 2015. 8

[8] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 1997. 2

[9] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2014. 6

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Tran. PAMI*, 2010. 2

[11] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980. 1

[12] K. Ganchev, J. Graça, J. Gillenwater, and B. Taskar. Posterior regularization for structured latent variable models. *JMLR*, 2010. 2

[13] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 6

[14] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, 2014. 7

[15] D. Heckerman. A tractable inference algorithm for diagnosing multiple diseases. *arXiv preprint arXiv:1304.1511*, 2013. 2

[16] J. Hoffman, D. Pathak, T. Darrell, and K. Saenko. Detector discovery in the wild: Joint multiple instance and representation learning. In *CVPR*, 2015. 1, 5

[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia, MM*, 2014. 4, 8

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

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[20] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 1

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 5, 6, 7

[22] G. S. Mann and A. McCallum. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *JMLR*, 2010. 2

[23] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *CVPR*, 2015. 7

[24] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. *arXiv preprint arXiv:1502.02734*, 2015. 2, 5, 6, 7, 8

[25] D. Pathak, E. Shelhamer, J. Long, and T. Darrell. Fully convolutional multi-class multiple instance learning. In *ICLR*, 2015. 1, 2, 5, 6

[26] P. O. Pinheiro and R. Collobert. Weakly supervised semantic segmentation with convolutional networks. In *CVPR*, 2015. 2, 5, 6

[27] J. C. Platt and A. H. Barr. Constrained differential optimization for neural networks. 1988. 2

[28] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 1, 6

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 5, 6

[30] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*. 2012. 2

[31] H. Song, R. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, and T. Darrell. On learning to localize objects with minimal supervision. In *ICML*, 2014. 2

[32] A. Vezhnevets and J. M. Buhmann. Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. In *CVPR*, 2010. 2

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

[34] C. Wang, W. Ren, K. Huang, and T. Tan. Weakly supervised object localization with latet category learning. In *ECCV*, 2014. 2

[35] S. H. Zak, V. Upatising, and S. Hui. Solving linear programming problems with neural networks: a comparative study. *IEEE transactions on neural networks/a publication of the IEEE Neural Networks Council*, 6(1):94–104, 1994. 2

[36] C. Zhang, J. C. Platt, and P. A. Viola. Multiple instance boosting for object detection. In *NIPS*, 2005. 2

[37] L. Zhang, Y. Gao, Y. Xia, K. Lu, J. Shen, and R. Ji. Representative discovery of structure cues for weakly-supervised image segmentation. *Multimedia, IEEE Transactions on*, 2014. 2