Towards Weakly Supervised Semantic Segmentation by Means of Multiple Instance and Multitask Learning.

Alexander Vezhnevets Joachim M. Buhmann
ETH Zurich
8092 Zurich, Switzerland
{alexander.vezhnevets,jbuhmann}@inf.ethz.ch

Abstract

We address the task of learning a semantic segmentation from weakly supervised data. Our aim is to devise a system that predicts an object label for each pixel by making use of only image level labels during training – the information whether a certain object is present or not in the image. Such coarse tagging of images is faster and easier to obtain as opposed to the tedious task of pixelwise labeling required in state of the art systems. We cast this task naturally as a multiple instance learning (MIL) problem. We use Semantic Texton Forest (STF) as the basic framework and extend it for the MIL setting. We make use of multitask learning (MTL) to regularize our solution. Here, an external task of geometric context estimation is used to improve on the task of semantic segmentation. We report experimental results on the MSRC21 and the very challenging VOC2007 datasets. On MSRC21 dataset we are able, by using 276 weakly labeled images, to achieve the performance of a supervised STF trained on pixelwise labeled training set of 56 images, which is a significant reduction in supervision needed.

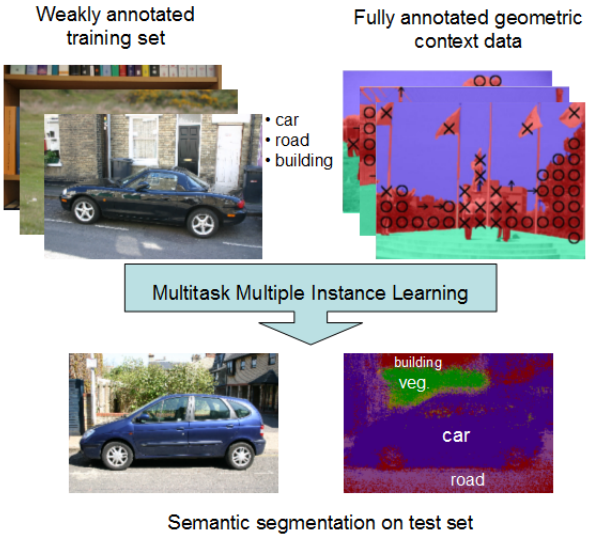

Figure 1. A schematic illustration of our approach. We use multitask multiple instance learning to perform the semantic segmentation of objects that were weakly annotated in training set using geometric context estimation as secondary task. Bellow is the result on test image.

1. Introduction

Semantic segmentation is a task of simultaneous object segmentation and recognition. For each pixel in the image $p_i \in I$ one has to predict a label $y_i \in \{1, ..., C\}$, which corresponds to semantic object class, like a car, a tree or a face. This is one of the most challenging and fundamental tasks in computer vision. In recent years there has been a great progress in solving this task [11, 10, 7]. These approaches rely on a training set of images annotated by a human supervisor, where for each pixel a corresponding label is known. This ground truth labeling is very tedious, frustrating and in the end expensive to obtain. As mentioned in [10], it takes between 15-60 minutes to obtain pixelwise annotation for just one image! The question we want to answer in this paper is the following: is a full, per pixel labeling of the data a necessary evil, or is a weaker, and thus cheaper and easier to obtain, way of supervision is suffcient? See Figure 1 for a schematic illustration of our approach.

We look into the case when supervision is given only through tags. These image level labels, specify whether a certain object is present or not in the image, but without any information on the object's spatial location or shape. Thus a given training set is a set of pairs $\{I_i, Y_i\}_{i=1}^N$, where I_i is an image and Y_i is a set of labels assigned to image I_i. Such labels can be produced with only few clicks per image. We specifically aim for the more challenging and realistic scenario where several objects may be present in a single image.

We will show that a weakly supervised semantic segmen-

tation can be naturally cast as multiple instance learning (MIL) problem. This setting is concerned with a learning scenario where samples come in multisets (bags) and labels are known only for these bags, but not for the instances themselves. Current MIL research approaches the problem by maximizing a classification margin over hypothesis space (parameters of the classifier) and possible labelings of the instances in bags. We present empirical evidence that a direct application of these techniques can severely suffer from overfitting. Building on the Semantic Texton Forest (STF) framework [11], we present two techniques that allow us to solve this problem. First is concerned with the probability estimates in the leaf nodes of the forest. The second improves the structure of the forest by means of multitask learning (MTL).

We report on the following contributions:

- We propose an algorithm for estimating unobserved pixel label probabilities from image label probabilities in the leaf nodes of the STF in a regularized way, improving over margin maximizing solution;
- We present an algorithm, that improves the structure of the STF by using a geometric context estimation task as a regularizer in MTL framework;
- By training on 276 weakly labeled images from MSRC21 dataset our method achieves the accuracy of a supervised algorithm, trained on a set of 56 pixelwise labeled images.

The remainder of the paper is organized as follows. In Section 2 we review the area of semantic segmentation and describe the STF framework that we will be using. In Section 3 we survey the multiple instance learning literature. In the following Section 4 we cast the problem of weakly supervised semantic segmentation as MIL problem and proposed solutions are described. In Section 5 we present the experimental results and in Section 6 we conclude.

2. Semantic segmentation

In this section we give a brief overview of the semantic segmentation field. One of the first works [7] incorporated region and global label features to model shape and context in a conditional random field. Another approach is described in [12], where a conditional random field approach was also taken. In this work unary potentials were trained by boosting based classifier. They integrated color, texture and shape cues efficiently and were able to handle many classes at the same time. [11] proposed a Semantic Texton Forest (STF) framework yielding state of the art results. This approach is based on a random forest acting directly on pixel level. We have chosen STF as our framework, because it is fast, efficient and easy to reproduce.

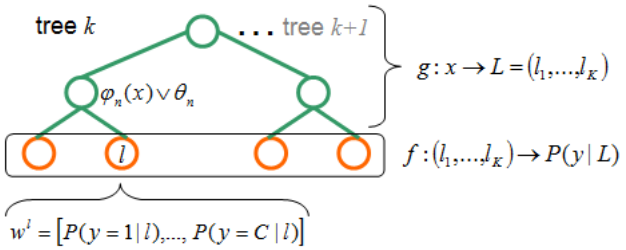

Figure 2. A schematic illustration of the STF. A forest consists of a structure $g(x)$, consisting of nodes with split functions, and probability estimates in the leaf nodes f.

The Semantic Texton Forest. A decision forest is an ensemble of K decision trees. A decision tree works by recursively branching left or right down the tree according to a learnt binary split function $\phi_n(x) : X \to \{0, 1\}$ at the node n, until a leaf node l is reached. Associated with each leaf l in the tree is a learned class distribution $P(c|l)$. Classification is done by averaging over the leaf nodes $L = (l_1, ..., l_K)$ reached for all K trees:

$$P(c|L) = \frac{1}{K} \sum_{k=1}^{K} P(c|l_k). \quad (1)$$

Conceptually, a forest consists of a structure, consisting of nodes with split functions, and probability estimates in the leaf nodes. We can represent a forest as a complex function $f(g(x))$, where $g : X \to \mathbb{N}^K$ maps the instance feature vectors to the vector of leaf indices, reached for each tree and $f : \mathbb{N}^K \to [0, 1]^C$ maps those indices to class probability estimates. Each leaf then has to store a vector $w^l = [P(y = 1|l), ..., P(y = C|l)]$.

Trees are trained independently on random subsets of the training data. Learning proceeds recursively, splitting the training data at node into left and right subsets according to a split function $\phi_n(x)$. At each split node, several candidates for $\phi_n(x)$ are generated randomly, and the one that maximizes the particular loss function is chosen. The recursive training usually continues to a fixed maximum depth without pruning. The class distributions $P(y|l)$ are estimated empirically as a histogram of the class labels y_i of the training examples i that reached leaf l. The split functions $\phi_n(x)$ in STFs act on small image patches p of size $d \times d$ pixels. These functions can be (i) the value $p_{i,j,b}$ of a single pixel (i, j) in color channel b, or (ii) the sum $p_{i_1,j_1,b_1} + p_{i_2,j_2,b_2}$, (iii) difference $p_{i_1,j_1,b_1} - p_{i_2,j_2,b_2}$, or (iv) absolute difference $|p_{i_1,j_1,b_1} - p_{i_2,j_2,b_2}|$ of a pair of pixels (i_1, j_1) and (i_2, j_2) from possibly different color channels b_1 and b_2.

STF, as presented above, provides prediction on the basis of local context only. To bring a global image context into play an Image Level Priors (ILP) are proposed. The

support vector machine (SVM) is trained to predict whether a certain object is present in the image. Output of SVM is scaled to the probability simplex and pixel level STF predictions are then multiplied by it. A kernel for the SVM is constructed by matching the amount of pixels in two given images that pass through the same nodes in the STF. We refer the reader to the original publication [11] for more details on ILP and STF training.

STF with ILP already provides the state of the art results for supervised semantic segmentation. In [12] a second level forest was described, which operated on the rectangular count features [12]. Training a second level forest involves estimation of spatial relation between semantic objects, which a very challenging task in weakly supervised scenario, thus we decided to leave for the future work.

3. Multiple Instance Learning

In this section we introduce the reader to the field of MIL. In MIL the training instances come in multisets (bags) $B_i \subset \mathbb{R}^d, i = 1, ..., N$. Each bag consists of a number of instances $B_i = \{x_i^1, x_i^2, ..., x_i^{m_i}\}$. Bag has a label Y_i associated with it, thus a training set consists of set of pairs $\{B_i, Y_i\}_{i=1}^N$. Usually, in MIL one considers only binary bag labels $Y_i \in \{-1, 1\}$, where negative label means that none of instances in the bag has positive label, while positive bag label means that at least one of them is positive. Our case is different, since we may have multiple labels per bag, thus our label is a set $Y_i \subseteq \{1, ..., C\}$. All labels are symmetric – if a bag has a certain label, then at least one instance has it, otherwise none does. Each instance x_i^j is assumed to have one certain label y_i^j associated with it, but this label is not observed during training. There are two possible objectives for MIL. First on is to learn a bag classifier $f^B : \mathbb{R}^d \to \mathcal{Y}^{\text{bag}} \subseteq \{1, ..., C\}^C$, which predicts labels for bags. We are interested in the the other objective, which is to learn instance classifier $f : X \to \{1, ..., C\}$ that predicts instance labels.

A common way to approach this problem is by a maximizing the classification margin over all possible parameters of the classifier and possible labelings of the instances in bags. SVM formulations are given in [1, 9]. And since the optimization task is non-convex, the authors use different heuristics to solve it. Essentially, the optimizer is free to vary the classifier's parameters and change the labels of the instances to achieve the best possible classification margin. In the binary case, the constraints are specified such that at least one sample from a positive bag is classified as positive and all samples from negative bags remain negative. We will call this approach an agnostic margin maximization, as it does not make any assumptions about bag and instance labels relation. A more detailed review can be found in [6]. We would like to note that while margin maximization has a solid theoretical foundation in supervised learning, there is no theoretical support for such principle in MIL, when instance labels are the target of prediction. This implies that a solution with a maximum margin is not necessarily the one with high generalization.

Most of the works in MIL only consider bag-level prediction [1, 13, 18]. In our case, we are interested in instance label prediction. This setting was considered specifically by [15], where authors train a face detector by considering patches around a user specified bounding box as a bag. Our setting is different in the number of labels we have and in the label interpretation. In our case there is no "negative" label, which if present on the bag states that there are no positive examples in it. Labels in our case are completely symmetric and only vote for their class being present.

There are few works that consider MIL with multiple labels [16, 18, 5]. Work [16] is closest to our setting, although they are concerned with bag-level classification for image categorization. They use a hidden conditional random field for image categorization with a brief mention of potential application to instance-level classification. We compare to them in Section 5. In [5] authors are constructing bags of multiple segmentation of an image, learning to extract the one that contains an object. They also use agnostic margin maximization approach. Authors only provide qualitative results for pixel label prediction and do not consider cases when one image can contain more then one object.

4. Multiple instance learning for semantic segmentation

In this section we cast a weakly supervised semantic segmentation as MIL problem. Each image is a bag B_i of pixels $\{x_i^j\}_{j=1}^{m_i}$, where each pixel x_i^j has only one label y_i^j. The image gets a label if at least one pixel in the image has it (signifying that there is an object of that class in the image), thus image can have multiple labels $Y_i \supseteq \{1, ..., C\}$. We observe only image/bag labels during training. In our particular case, if an image has label "grass" then there is at least one pixel of grass in the image. Thus, there is at least one instance in the bag with certain label if the bag has it, and none otherwise. Our main concern is on *predicting instances labels*, which corresponds to predicting semantic labels for pixels and solving the semantic segmentation problem.

In this section we first present a STF for MIL solution based on agnostic margin maximization. We demonstrate that it suffers from overfitting – although target loss is efficiently minimized, though accuracy of semantic segmentation is poor. Our conclusion is that this approach is not applicable. Therefore later we present two different techniques. First one, assumes that forest structure is fixed and aims to accurately estimate probabilities in the leaf nodes.

The second techniques addresses building better structure of the forest by means of multitask learning.

4.1. Agnostic Margin Maximizing Random Forest for MIL (AmmMIL-RF)

In this section we will describe a learning method for weakly supervised STF that completely follows principles of the margin maximization - **AmmMIL-RF**.

In the classical supervised case, for a training set X^s the loss is given by:

$$\mathcal{L}(f) = \sum_{(x,y)\in X^s} l(f(x), y), \tag{2}$$

where $l(\hat{y}, y) : \mathbb{R}^C \times \{1, ..., C\} \rightarrow \mathbb{R}$ is a loss for an individual instance-label pair.

For decision trees, split function $\phi_n(x)$ at node n is selected based on a score which measures the purity of the node. Usual choices are the entropy or the gini index. These scores can be shown to optimize a certain loss function of the form in Eq. 2 [3]. These losses are margin maximizing, meaning that their minimization also maximizes the margin. In MIL setting we would want to minimize such loss not only over all possible classifiers f, but also over all possible labelings of training samples, with the only constraint that a sample can only be labeled by one of its bag labels, solving:

$$\min_{f, y:y_i\in Y_i} \mathcal{L}(f) = \sum_{(x_i,y_i)} l(f(x), y). \tag{3}$$

The difference from the supervised case is that now we have to search through all possible labelings of instances in the bags and not only through the classifiers parameters. This makes the problem non-convex. [6] propose to approximately solve this problem by using deterministic annealing (DA). We extetend their DA approach for random forest in somewhat similar fashion to semi-supervised random forest [3].

To relax the problem we introduce a distribution over the labels of the instances, $\hat{p}$, and enforce a controlled uncertainty into the optimization process. The new loss is the following:

$$\mathcal{L}_{DA}(f, \hat{p}, T) = \sum_{x\in X} \sum_{y} \hat{p}(y|x) l(f_y(x), y) + T \sum_{x\in X} H(\hat{p}), \tag{4}$$

where l could be any standard margin maximizing loss, T is a temperature parameter and $H(\hat{p}) = -\sum_{i=1}^{C} \hat{p}(i|x) log(\hat{p}(i|x))$ is an entropy. When the temperature is large the dominating term is the entropy, thus the model maintains a high level of the uncertainty and is nicely convex. When the temperature is lowered (as $T \rightarrow 0$), the original loss becomes more and more important. At fixed temperature $T = \tau$, the problem reads as follows:

$$(f^*, \hat{p}^*) = \arg\min_{f,\hat{p}} \mathcal{L}_{DA}(f, \hat{p}, \tau) \tag{5}$$

The algorithm to minimize it consists of two stages. First, with a fixed distribution over labels find an optimal classifier and in the second stage update the distribution.

With a fixed label distribution, we randomly choose labels for instances according to the current $\hat{p}$ and train a random forest. Then we recompute the optimal distribution according to the current classifier f. This can be done in an analytical form, by taking the derivative of the loss and setting it to zero (note that f is fixed).

$$\hat{p}^*(i|x) = \exp(-\frac{l(f(x)) + T}{T})/Z(x), \tag{6}$$

where $Z(x)$ is a normalization factor. We also set the probability of a label that is not present in the image corresponding to current pixel to zero: $\hat{p}^*(i|x_j) = 0; \forall x_j : i \notin Y_j$. These iterations are implemented together with a gradual cooling of T. We started with $T_0 = 10$, which essentially turns $\hat{p}$ uniform and used the exponential cooling scheme $T_i = T_0 \cdot 0.5^i$.

Unfortunately this naive margin maximizing approach overfits, apparent from our experiments. We evaluate this approach on the MSRC21 dataset for semantic segmentation. We train an ILP after each iteration using current random forest. Figure 3 show the results on test accuracy and training loss (3) for the first five iterations. We see that a decrease in the training loss does not result in the increase in the test accuracy as one would expect. In fact we know that the problem lies with this loss and and not with the optimization: we repeat the experiment with a good initial model (i.e., fully supervised STF model) and an appropriately low initial temperature. Whilst the loss is still decreasing at each iteration, the test accuracy drops from 64% to 40%. This overfitting effect may be due to the high degree of freedom when optimizing over both classifier parameters and the exponentially many possible labelings.

Since this naive solution fails for semantic segmentation MIL, where bags have multiple labels and the target is instance label prediction, we present a set of solutions, which is tailored for this MIL scenario. We concentrate on different parts of the STF, first on estimating correct probabilities in the leaf nodes and then on building a better structure of the tree, presenting two algorithms **PPinv** and **MT-STF**.

4.2. Estimating Leaf Probabilities (PPinv)

Now we will focus on producing good class probability estimates in the leaf nodes of the STF. Let us assume that the structure of the forest $g(x)$ (as defined in Section 2) is given and consider a particular leaf l in the forest. For each

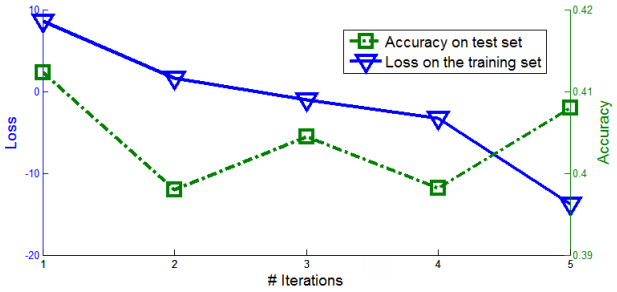

Figure 3. This plot depicts loss on training set and semantic segmentation accuracy on test set as a function of deterministic annealing iterations. As one can see, steady decline of loss does not lead to any improvement in semantic segmentation accuracy. Further iterations reduce the loss even further, but the accuracy does not improve. We did not plot them because of scale problems.

leaf l, there is a group of corresponding pixels. Let y be a pixel label and Y a set of labels of an image that pixel belongs to. For each of these pixels we want to know the probability $P(y = j|l)$ of it having a label j. However, we can only observe the probability of this pixel belonging to an image that has a label i: $P(i \in Y|l)$. To shorten the notation, we will further omit conditioning on a leaf, since it appears ubiquitously, writing just $P(y = j)$ and $P(i \in Y)$. We can factorize $P(i \in Y)$ probabilities in the following form:

$$P(i \in Y) = \sum_{j=1}^{C} P(i \in Y|y = j)P(y = j). \qquad (7)$$

We see that image labels probabilities are connected to the desired pixel label probabilities through the conditional probabilities $P(i \in Y|y = j)$. These are not known, but having a good approximation of them would allow us to significantly reduce the degrees of freedom and uncertainty in the choice of estimates of $P(y = j)$. These conditionals could be rewritten as:

$$P(i \in Y|y = j) = P(i \in Y|j \in Y)\frac{P(y = j|j \in Y, i \in Y)}{P(y = j|j \in Y)}. \qquad (8)$$

Here we propose to make an assumption and let the fraction in the right hand side to be equal to one, thus approximating $P(i \in Y|y = j) \approx P(i \in Y|j \in Y)$. In words, the probability of seeing a "car" pixel in an image with a label "car" is not dependant on this image having an additional label "house". Formally:

$$P(y = j|j \in Y, i \in Y) = P(y = j|j \in Y). \qquad (9)$$

Let us introduce a few notation. Let w be a vector of pixel-level class probabilities in the leaf, such that $w_i = P(y = i)$. Let matrix A be the matrix of conditional probabilities, such that $A_{ij} = P(i \in Y|j \in Y)$ and let z be a vector of image-level probabilities, such that $z_i = P(i \in Y)$. Using this notation Eq.(7) can be written as linear system $Aw^T = z^T$. Matrix A is usually ill determined, thus many solutions are possible. We need a rule to select one of them. We choose to select the one with minimal L_2 norm, since it corresponds to the most uniform allocation of weights to the classes. In this formulation the problem is convex and has a global minimizer, which is easily computable by taking the pseudoinverse transform of matrix A. Thus our solution is $\hat{w}^T = A^\dagger z^T$. In principle, we should also take into account the probabilistic nature of w and restrict its domain to the simplex. In practice, the our $\hat{w}$ is always very close to the simplex, hence enforcing this constraint does not change our solution significantly, but significantly increases computation.

Our proposed algorithm, **PPinv**, requires only the pseudoinverse matrix computation for each leaf, in addition to learning the tree structure $g(x)$. Restricting the possible w to only those that satisfy the approximation of the linear system (7) dramatically reduces the degree of freedom and thus regularizes the solution, preventing overfitting.

4.3. Multitask Learning of STF Structure (MTL-RF)

In this section we will address the problem of building a better structure of the STF. The structure of the forest in STF corresponds to the generic arrangement of image patches, that is relevant to a structure of the visual world. Each split, ideally, tries to separate image patches into more homogenous groups. It is reasonable to assume that this structure is the same for different tasks, defined in the image domain. If we could use an external task, which is defined on the same domain (digital images) and has a fully labeled training set, we could "import" useful structures from that domain.

We formalize this intuition within a MTL framework. MTL is a machine learning paradigm that learns a task together with other related tasks at the same time, using a shared representation. Without the loss of generality we assume that we have two tasks. Given two training sets $\{(x_i^t, y_i^t)\}_{i=1}^{N^t}, t = \{1, 2\}$, where the domain of the instances are the same $\forall t, i : x_i^t \in \mathcal{X}$ the aim is to learn classifiers for both tasks $\mathcal{F}^t(\mathcal{G}(x)) : \mathcal{X} \rightarrow \mathcal{Y}^t$, such that a the sum of their errors is minimal. $\mathcal{G}(x)$ is shared by both classifiers and $\mathcal{F}^t$ is task dependant. By restraining classifiers to have a shared part, the degree of freedom is reduced. This can be relaxed so that $\mathcal{G}^t$ is also task dependant, but a regularizer that penalizes the difference between $\mathcal{G}^1$ and $\mathcal{G}^2$ is introduced [2]. An even more general Bayesian approach can be taken, where only a prior on the classifiers is shared [17]. One can also be interested only in the prediction for one main task, as in our case, then $\mathcal{F}^2$ can be

ignored later, although it is optimized during training. Here we show that MTL is very much applicable in a MIL scenario, where it can help to deal with a significant amount of ambiguity in the data.

As a supplementary task we consider geometry context estimation. The dataset from [8] is used. In this dataset, each pixel is labeled according to its geometric property – vertical, horizontal, porus, solid, etc (Figure 1, top right). We need to construct training sets $X^t = \{(x_i^t, y_i^t)\}_{i=1}^{N^t}, t = 1, 2$ for both tasks. For the geometric task pixelwise labeling is available. To construct a training set for the main task where semantic classes are objects, we sample labels for the pixels uniformly from the labels of corresponding images, this would correspond to a step of the deterministic annealing algorithm with high temperature. Such weak supervision helps classifier to distinguish between objects that do not appear in the same images.

Following the notation introduced in Section 2, we propose to let the classifiers share the structure $g(x)$ of a decision forest and let the probabilities in the leafs be task dependant. The loss for a random forest we optimize is the following:

$$\mathcal{L}(f^1, f^2, g) = \sum_{t=1}^{2} \frac{1}{|X^t|} \sum_{(x,y)\in X^t} l(f^t(g(x)), y) \qquad (10)$$

Here, l is the same as in Eq.(2). After building the structure of the forest in this way, we can apply the **PPinv** algorithm to acquire good probability estimates for the leafs. Training the SVM for ILP, once the forest is constructed, is no different from the single task case. The resulting classifier has the same form as a standard STF and requires no multitask related input during testing. It also does not require any geometry specific features for learning. The additional task is only used for the training and is ignored afterwards. In principle, any other secondary task can be considered, we can even consider several these simultaneously.

5. Experiments

We conducted experiments on the Microsoft Research Cambridge dataset (MSRC21) and VOC2007 [4]. The basic implementation of STF was kindly provided by authors of [11], parameters (number and depth of trees) where kept the same. We compare our three algorithms: **AmmMIL-RF**, **AmmMIL-RF+PPinv**, and **MTL-RF+PPinv**.

5.1. Semantic Segmentation

MSRC21 Dataset [12]. This is a multiclass dataset – with total 21 classes the average number of 3 objects per image and about 80% of images having more then one object in them. We use the same split into training and test set as [11]. First we investigate the accuracy of semantic segmentation. In Table 5.1 the results are presented. The accuracy without and with ILP are presented in the first and second columns respectively. As a good baseline we provide results for a pixelwise supervised STF trained on 56 (validation set only) and 276 images (training set) [11]. The reader is invited to visually inspect the results, presented in Figure. 5.1. **MTL-RF+PPinv** algorithm performs as good as a supervised one that is trained on 56 images.

Algorithms	**Test**		**Train**
	no ILP	ILP	ILP
AmmMIL-RF	0.14	0.41	0.54
AmmMIL-RF+PPinv	0.35	0.47	0.61
MTL-RF+PPinv	**0.38**	**0.51**	**0.63**
Supervised (56)	0.33	0.50	0.85
Supervised (276)	0.48	0.64	0.83

Table 1. Accuracy of the semantic segmentation on MSRC. Supervised (56) and (276) correspond to STF classifiers trained with full pixelwise supervision on only validation (56 images) and only training (276 images) set respectively.

We also evaluate the semantic segmentation accuracy on the training set (Table 5.1, last column) to investigate how well the unobserved pixel labels were inferred for images with observed imagewise labels. We can see, that **MTL-RF+PPinv** is again the winner and its accuracy is equal to test set accuracy of supervised algorithm. Essentially, if one can provide accurate image labels, then a semantic segmentation for these images can be produced within the accuracy of a supervised algorithm's generalization.

Authors of [16], which is concerned with image categorization (bag level prediction), briefly mention that their approach is applicable to predicting pixel labels, but the only experimental validation that they provide is the average area under the ROC curve (AUC). We outperform their AUC of 0.86 by achieving AUC of 0.89 with **MTL-RF+PPinv**. Further comparison is not possible since no accuracy or segmentation results where provided in that paper. In [14] a pLSA model over a codebook of features (SIFT, color, position) produced by unsupervised k-means clustering is employed. The output of pLSA is then post-processed by an MRF to enforce segments continuity. In contrast to this work, we focussed on learning discriminative low level features (STF) in a weakly supervised way. Compared to [14] we have a superior speed (in test time it takes 2 seconds for a label of every pixel to be inferred vs. 2-4 seconds for every *tenth* pixel along 2 dimensions, which is ≈ 100 slower, although the accuracy of Verbeek is superior: 60%.

VOC 2007 Segmentation Dataset [4]. This dataset contains 21 extremely challenging classes including background. We train on this data using the trainval split and keeping parameters as for MSRC21. Even supervised methods do not yield satisfactory results on this data. We report results of our algorithms, to broaden the scope of evalua-

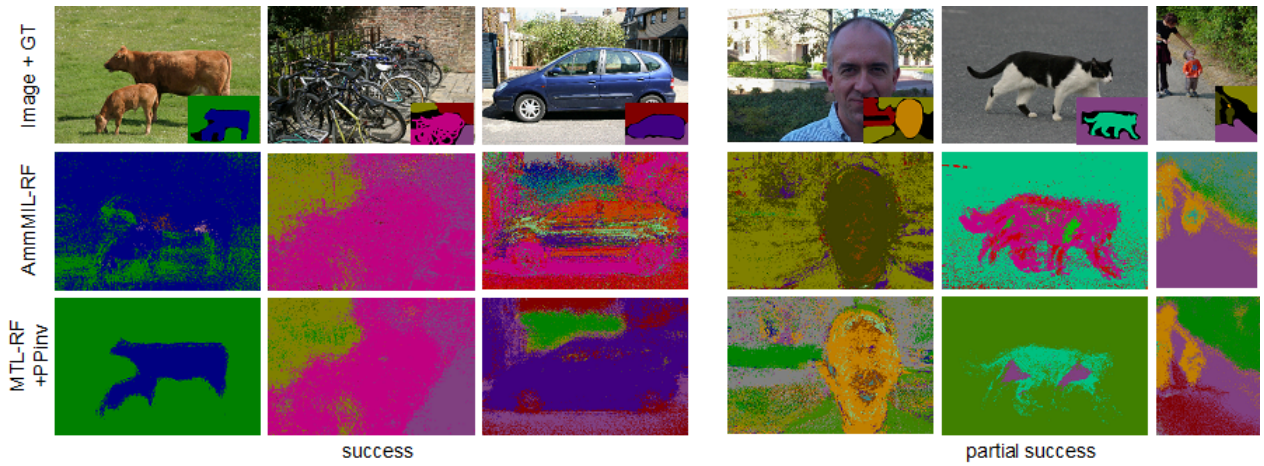

Figure 4. MSRC21 segmentation results. Note, that **MTL-RF+PPinv** is much more spatially consistent then AmmMIL-RF. Among less successful results on the right the typical confusions can be seen - road is confused with grass, body with face and bright buildings in the background with sky. Case of body and face is a particulary hard one, since in the most images they are seen together and there is no way to separate one from another without additional supervision.

tion. **AmmMIL-RF** has total per pixel accuracy of 0.32 and average per class accuracy of 0.07, while **MTL-RF+PPinv** has 0.41 and 0.07 respectively. These results are even comparable to some supervised approaches listed in [4] , like "Brookes" and "INRIANormal", having 0.085 and 0.077 average per class accuracy.

5.2. Detailed Analysis of MTL-RF and PPinv

The two conceptual components of the STF are forest structure and probability estimates in the leaf nodes. We proposed techniques for enhancing both. To analyze these enhancements independently, we perform the following set of experiments. We take three forests - one trained with the full pixelwise supervision, one trained by **AmmMIL-RF** algorithm and one trained by **MTL-RF**. Then, for each forest, we assign class probabilities in the leaf nodes using pixelwise supervision, **PPinv** and using random sampling of pixel label from the corresponding image labels (**AmmMIL-RF** solution with the highest temperature). Finally, we use ILP based on the respective forest for all methods. Such comparison would allow us to investigate how the structure and the probability estimate behaves in detail. The results are presented in Table 5.2. It is evident that both, forest structure and probability estimate in leafs are important. Structure of **MTL-RF** is almost as good as pixelwise supervised. **PPinv** probability estimation algorithm brings significant improvement to accuracy of semantic segmentation for all structures, although an improvement for **MTL-RF** and supervised one are more significant then for **AmmMIL-RF**. To validate the assumption (9), that was used to construct **PPinv** algorithm we measure the difference between true $P(i \in Y|y = j)$ its approximation

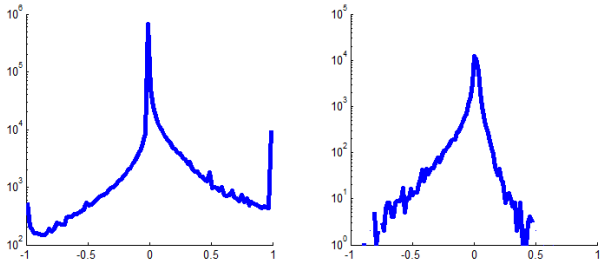

Figure 5. On the left is the histogram of the differences between true $P(i \in Y|y = j)$ and it's approximation $\approx P(i \in Y|j \in Y)$. On the right, the histogram of differences between estimations of $P(y = i|l)$ by **PPinv** and pixelwise labeling. Both are in logarithmic scale.

Structure	**Leaf Probabilities**		
	Rand.	PPinv	Supervised
Supervised (276)	0.36	0.51	0.64
AmmMIL-RF	0.41	0.46	0.59
MTL-RF	0.37	0.51	0.62

Table 2. Accuracy of different combinations of STF structure and estimation of class probabilities in leafs. Supervised (276) correspond to an STF classifier trained with full pixelwise supervision on full training set of 276 images.

$P(i \in Y|j \in Y)$. Also we measured the difference between **PPinv** estimated probabilities and the ones obtained from pixelwise labeling evaluated on MSRC21 dataset (Figure 5.2). The differences are usually around zero, although there are some deviations. This implies the assumption (9);

Algorithms	Mean Av.Prec.	Av. AUC
AmmMIL-RF	0.68	0.92
MTL-RF	**0.70**	**0.93**
Supervised (56)	0.68	0.93
Supervised (276)	0.70	0.94

Table 3. Image categorization results on MSRC.

5.3. Image Categorization

Image categorization is itself an important task, also image level priors for the semantic segmentation are derived from it. We compared the mean average precision and average AUC of categorizers, obtained by training forest by **AmmMIL-RF** and by **MTL-RF**. Results are presented in Table 5.3. For the purity of the experiment, we report the accuracy of the SVM predictor alone, in contrast to [11], where results of SVM where multiplied by the average response of the random forest for the pixels in the image. This is done to better separate the bag level prediction and instance level prediction. We outperform results of [16]; there only average AUC results were reported. In this experiments MTL is able to improve the performance of the categorizer to the level of the supervised one and outscores the competitor [16].

6. Conclusion

In this paper we have presented techniques for weakly supervised learning for semantic segmentation. We casted this task as an MIL problem. We discovered that a naive application of the standard MIL techniques, based on margin maximization over all possible parameters of the classifier and possible labelings of the instances in bags, overfitts and thus fails to solve the problem. We proposed a novel regularized way to estimate the unobserved instance label probabilities in the STF framework, which does not suffer from overfitting. We also made use of the external dataset for geometric context estimation to further regularize our solution. By this we demonstrated that the amount of supervision required to learn a semantic segmentation classifier can be significantly reduced.

As future work we plan to investigate the scenario where we have weak supervision, but a small amount of pixelwise annotation is available. This small supervision could be employed to obtain better estimate of matrix of conditional probabilities in Eq.(7). Furthermore, a second level segmentation forest used in [11] may by trained be utilizing this small pixelwise supervision. Considering more different supplementary tasks for MTL is another perspective for further improvements. In future, we aim to reach the level of the state of the art systems that require fully annotated data for training.

This work has been partially supported by the Swiss National Science Foundation under grant #200021-117946 and partially by the EU under the SIMBAD project, (FP7-FET #213250).

References

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*. MIT Press, 2003. 3

[2] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *NIPS*, 2007. 5

[3] J. S. H. B. Christian Leistner, Amir Saffari. Semi-supervised random forests. In *ICCV*, 2009. 4

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 6, 7

[5] C. Galleguillos, B. Babenko, A. Rabinovich, and S. Belongie. Weakly supervised object localization with stable segmentations. In *ECCV*, 2008. 3

[6] P. Gehler and O. Chapelle. Deterministic annealing for multiple-instance learning. In *AIStats*, 2007. 3, 4

[7] X. He, R. Zemel, and M. Carreira-Perpinan. Multiscale conditional random fields for image labeling. In *CVPR*, 2004. 1, 2

[8] D. Hoiem, a.a. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*. 6

[9] O. L. Mangasarian and E. W. Wild. Multiple instance classi cation via successive linear programming. technical report 05-02. *Data Mining Institute, University of Wisconsin*, 2005. 3

[10] P. T. Pushmeet Kohli, Lubor Ladicky. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008. 1

[11] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *ECCV*, 2008. 1, 2, 3, 6, 8

[12] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006. 2, 3, 6

[13] Q. Tao, S. Scott, N. V. Vinodchandran, and T. T. Osugi. Svm-based generalized multiple-instance learning via approximate box counting. In *ICML*, 2004. 3

[14] J. Verbeek and B. Triggs. Region classification with markov field aspect models. In *CVPR*, 2007. 6

[15] P. Viola, J. Platt, and C. Zhang. Multiple instance boosting for object detection. In *NIPS*, 2006. 3

[16] Z.-j. Zha, X.-s. Hua, T. Mei, J. Wang, G.-j. Qi, Z. Wang, I. M. Group, and M. R. Asia. Joint multi-label multi-instance learning for image classification. In *CVPR*, 2008. 3, 6, 8

[17] J. Zhang, Z. Ghahramani, and Y. Yang. Learning multiple related tasks using latent independent component analysis. In *NIPS*, 2006. 5

[18] Z. Zhou and M. Zhang. Multi-instance multi-label learning with application to scene classification. In *NIPS*, 2006. 3