Discriminative Structure Learning of Hierarchical Representations for Object Detection

Paul Schnitzspan[1], Mario Fritz[2], Stefan Roth[1], and Bernt Schiele[1]
[1]Department of Computer Science, TU Darmstadt [2]UC Berkeley EECS & ICSI

Abstract

A variety of flexible models have been proposed to detect objects in challenging real world scenes. Motivated by some of the most successful techniques, we propose a hierarchical multi-feature representation and automatically learn flexible hierarchical object models for a wide variety of object classes. To that end we not only rely on automatic selection of relevant individual features, but go beyond previous work by automatically selecting and modeling complex, long-range feature couplings within this model. To achieve this generality and flexibility our work combines structure learning in conditional random fields and discriminative parameter learning of classifiers using hierarchical features. We adopt an efficient gradient based heuristic for model selection and carry it forward to discriminative, multidimensional selection of features and their couplings for improved detection performance. Experimentally we consistently outperform the currently leading method on all 20 classes of the PASCAL VOC 2007 challenge and achieve the best published results on 16 of 20 classes.

1. Introduction

Hierarchical and multi-feature representations have shown to be a powerful basis for achieving impressive results in object detection and recognition across a variety of different datasets [1, 6, 12, 24, 28]. These are often paired with discriminative learning approaches, such as support vector machines [6, 12]. The use of multiple features requires appropriate determination of the relative importance, *i.e.*, weighting, of the features. Beyond doing this manually, a number of recent approaches have attempted to learn these weights automatically [1, 24] using variants of multiple kernel learning. These learning mechanisms, however, only allow to identify and weigh the most discriminant features, but do not allow to identify and model the interplay between features that may prove important to representing objects well. In fact, one may posit that for many object classes the coupling between different features might be key to discriminating object classes from others. A number of recent conditional random field approaches allow modeling local as well as simple hierarchical couplings of features [7, 9, 13, 21, 26]. In particular, these approaches associate a label with each localized feature and model label dependencies by leveraging the interplay of the corresponding features. These approaches are limited, however, in that they model only simple, short-range dependency structures in the label space; the corresponding features are typically neighboring in space or scale.

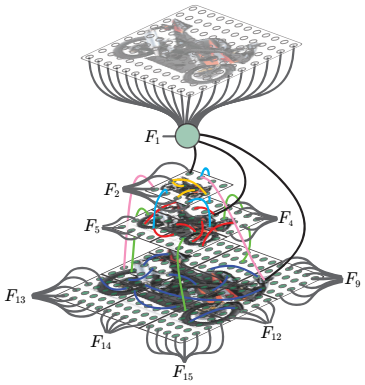


Figure 1. Schematic overview of our hierarchical model (Best viewed in color). The nodes of our graphical model are indicated as green dots; learned feature couplings are represented as colored lines. F refers to the discriminative unary classifiers.

To address these limitations, we propose an approach that allows to learn short-range as well as long-range dependencies, where the structure of these dependencies is identified and learned in a fully automatic manner. Unlike previous work, our approach does not require any notion of locality of the coupled features, but instead allows to find and model relevant (*i.e.* discriminant) couplings among arbitrary pairs of features. To enable learning of the interplay of features we cast the problem as one of structure learning in graphical models [14, 18, 20]. Specifically, we use a conditional random field to predict local labels from the image features and employ discriminative structure learning to identify dependencies whose modeling improves the discriminative power of the model.

We follow [1] by using a hierarchical and multi-feature

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

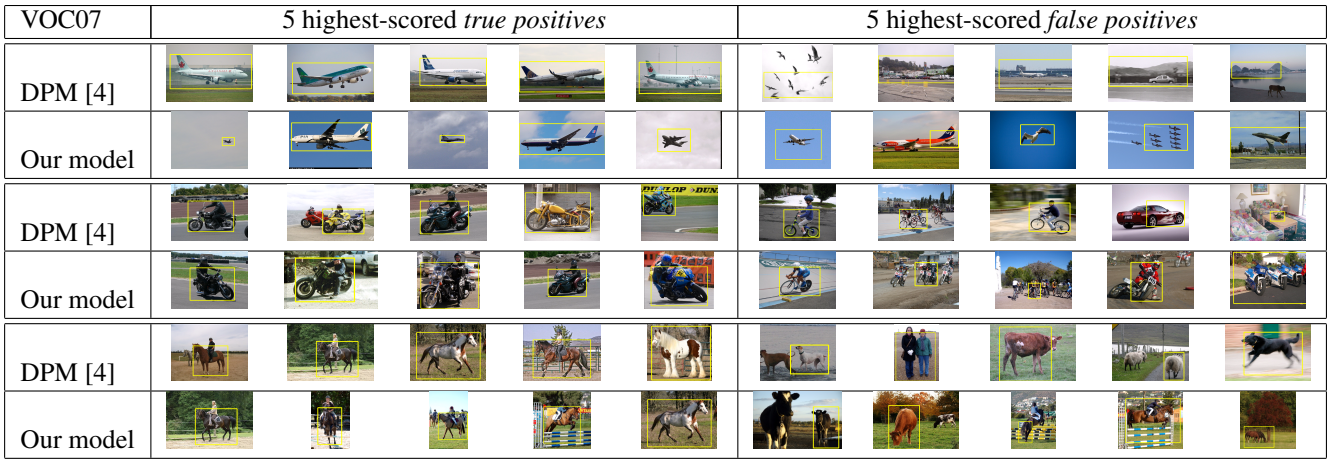


Figure 2. Highest-scored true and false positives of [4] and our model for the PASCAL VOC 2007 challenge (aeroplane, motorbike, horse). Our framework is more flexible in modeling viewpoint, appearance, and articulation changes.

representation of objects. In particular, our model is based on a hierarchy of HOG descriptors (histograms of oriented gradients, [2, 4]) and a hierarchical bag-of-visual-words (BoW) representation [5, 6, 12] (see Fig. 1 for a schematic overview). Our model extends previous work by learning the contribution of the different feature types and simultaneously including relevant long-range as well as short-range couplings between arbitrary pairs of image features. As such our framework could model the dependency between the prediction from a HOG feature at a certain level in the hierarchy and a BoW feature at another level (*c.f*. Fig. 1), if that improves the discriminative power of the model.

We apply our approach to the problem of object detection and show that it consistently outperforms SVM classifiers, which may be seen as the de facto standard in discriminant object model learning. On the PASCAL VOC 2007 detection challenge, the proposed approach outperforms the currently leading SVM-based technique [4] on all 20 object categories. Moreover, we report the most accurate results in the literature on 16 of the 20 classes.

As the experimental results below show, our model profits from the use of powerful hierarchical and multi-feature representations. It is important to note, however, that the proposed approach is very general and can be used for any local or global feature representation, not just the features used here[1].

Fig. 2 shows the first true positives (TP) and false positives (FP) of our model as well as of DPM [4], the currently leading method on the dataset. DPM typically assigns high scores to canonical sideviews, while our work seems to show more flexibility in modeling variations in viewpoint, appearance and articulation. Instead of being wholly misclassified, many FPs are due to misaligned bounding boxes.

[1]Code for structure learning of arbitrary feature representations will be available at http://www.mis.informatik.tu-darmstadt.de.

Related work. Conditional random fields (CRF) [11] have proven to be effective for a wide variety of applications, including challenging segmentation and categorization tasks [15, 21, 22, 26]. In contrast to Markov random fields, CRFs are discriminative approaches that avoid modeling the dependencies between the input variables (such as the images) and instead focus on modeling the dependencies of the output variables (*e.g*., class labels). To express these output dependencies most CRF approaches rely on pairwise graph structures based on very local, short-range connections [21, 22, 26]. Of course, this limits the modeling power, but introducing long-range dependencies in a dense, brute force fashion is often computationally prohibitive. Our model, on the other hand, can benefit from long-range dependencies, while staying tractable and efficient.

To facilitate that, we rely on structure learning in graphical models, which allows to identify the graph structure that best models the dependency structure inherent in the data. Since optimal structure learning in general models is NP-hard, a number of efficient approximations have been proposed [18, 19, 20]. Since our goal is high discriminative power in object detection, we employ a discriminative variant of structure learning. An overview of various approaches to structure learning is given by Schmidt *et al*. [20], who apply these methods to the problem of heart abnormality detection. Despite their power, structure learning methods have not found widespread use in computer vision. A notable exception is the work of Tran and Forsyth [23], who propose to estimate the configuration of pedestrians and learn a discriminative classifier based on a global descriptor enriched with configuration features.

To keep the learned model efficient, it is necessary to ensure that the graph structure stays sparse despite allowing for long-range connections. To that end, various regularization approaches have been proposed [20]. Lee *et al*. [14], *e.g*., suggest to use L1-regularization for structure learning

in Markov networks with promising performance improvements. They evaluated different heuristics for feature selection and reported results for the MNIST digits dataset, but are restricted to binary features. One contribution of this paper is a gradient-based heuristic for the case of continuous-valued multi-dimensional feature vectors. We then apply this discriminative feature selection method to the problem of discriminating objects from the background.

2. CRF model

In our approach we rely on conditional random fields (CRFs), which has several motivations, among them that structure learning in graphical models is a well-established field. Our approach represents each object class as a CRF with a pairwise graph structure, which models the posterior probability $P(\mathbf{y}|\mathbf{x})$ of labels $\mathbf{y}$ given an image $\mathbf{x}$. Each node $i \in V$ of the underlying graph represents a binary label $y_i \in \{1, -1\}$ encoding the presence or absence of an object of a specific class. The set of all possible edges $\Omega = V \times V$ connecting the nodes is partitioned into the *active set* $\mathcal{A} \subset \Omega$ and the *inactive set* $\mathcal{I} \subset \Omega$ (with $\mathcal{A} \cup \mathcal{I} = \Omega$ and $\mathcal{A} \cap \mathcal{I} = \emptyset$). The active set $\mathcal{A}$ defines the edge structure of our CRF model. Later we will see how to learn $\mathcal{A}$ automatically from training data; for now we assume that $\mathcal{A}$ is already given. The posterior distribution is then defined as

$$P(\mathbf{y}|\mathbf{x};\theta,\mathcal{A}) = \frac{1}{Z(\theta,\mathcal{A})} \prod_{i \in V} \psi_i(y_i, \mathbf{x}; \theta) \cdot \prod_{(i,j) \in \mathcal{A}} \phi_{ij}(y_i, y_j, \mathbf{x}; \theta), \tag{1}$$

where ψ_i are the unary potentials, ϕ_{ij} are the pairwise or edge potentials, θ are the parameters of the model, and $Z(\theta, \mathcal{A})$ is the partition function (a normalization factor). The set of parameters $\theta = \{\boldsymbol{\alpha}, \mathbf{w}, \mathbf{e}\}$ includes parameters of the unary potentials $\boldsymbol{\alpha}$ and $\mathbf{w}$, as well as the parameters $\mathbf{e}$ of the edge potentials.

Unary potentials. The unary potentials in the CRF allow for local and global evidence aggregation; each potential ψ_i models the evidence from considering a specific image feature $f_i(\mathbf{x})$. Our representation relies on several levels of features in a hierarchy, where the feature functions at the lowest level extract local representations and the feature functions at higher levels aggregate a larger area until a global view of the object is obtained at the top level (*c.f.* Fig. 1 for the hierarchical view on objects). The features will be explained in more detail in Section 2.1.

We define the unary potential for a node i using the softmax function (*c.f.* [10])

$$\psi_i(y_i, \mathbf{x}; \theta) = \frac{\exp\left(y_i \cdot \mathbf{w}_i^T \mathbf{F}\left(\boldsymbol{\alpha}_i, f_i(\mathbf{x})\right)\right)}{\sum_{c \in \{-1,1\}} \exp\left(c \cdot \mathbf{w}_i^T \mathbf{F}\left(\boldsymbol{\alpha}_i, f_i(\mathbf{x})\right)\right)} \tag{2}$$

based on a weighted combination of the output of a bank of N different classifiers $\mathbf{F}\left(\boldsymbol{\alpha}_i, f_i(\mathbf{x})\right) = \left(F\left(\boldsymbol{\alpha}_{i,1}, f_i(\mathbf{x})\right), \ldots, F\left(\boldsymbol{\alpha}_{i,N}, f_i(\mathbf{x})\right)\right)^T$. Each classifier is assumed to yield a continuous-valued score. $\boldsymbol{\alpha}_i$ are the parameters of the classifier, and $\mathbf{w}_i$ are the weights. In Fig. 1 the classifiers are denoted with F. Interestingly, such a formulation can be seen as a probabilistic analog to multiple-kernel learning [1, 24] as it allows for a weighted combination of different classifiers.

Edge potentials. The edge potentials ϕ_{ij} model the interaction of two labels y_i and y_j based on the interaction of two features $f_i(\mathbf{x})$ and $f_j(\mathbf{x})$. These pairwise potentials are crucial for our model, as they allow us to capture the interplay of features and therefore to define the structure of objects. To that end, we realize the pairwise potentials with a linear classification of concatenated unary features that is passed through a softmax nonlinearity:

$$\phi_{ij}(y_i, y_j, \mathbf{x}; \theta) = \frac{\exp\left((f_i(\mathbf{x}), f_j(\mathbf{x}))^T \mathbf{e}^{ij}_{y_i y_j}\right)}{\sum_{c,d \in \{-1,1\}} \exp\left((f_i(\mathbf{x}), f_j(\mathbf{x}))^T \mathbf{e}^{ij}_{cd}\right)} \tag{3}$$

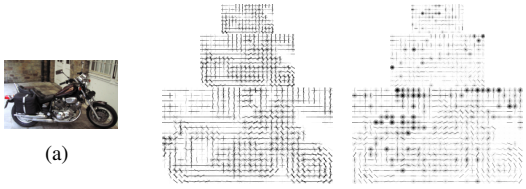
We use a specific classification vector $\mathbf{e}^{ij}_{cd}$ for each possible edge and each combination of labels that allows to model spatial dependencies and relations of different feature types. It is important here to emphasize that these potentials are not restricted to modeling only local neighborhood structures as in many recent approaches [7, 10, 13, 15, 21, 22, 26], but may also involve long-range dependencies of distant nodes.

Both, the unary and pairwise potentials, contribute to our discriminative framework in the sense that the unary potentials classify nodes in the hierarchy independently while the pairwise potentials encode dependencies and thus the spatial configuration and underlying structure of objects. What sets this work apart from previous approaches is that we are able to learn the graph structure $\mathcal{A}$ automatically, which gives us a sound and efficient way of modeling complex, long-range dependencies. This allows us to determine the structure of the underlying domain and simultaneously consider a powerful hierarchical view on objects.

2.1. Hierarchical features

Before showing how to learn the parameters and structure of the model, we will first introduce the features and classifiers that the CRF model is based on.

We use a hierarchical representation of objects, which provides a powerful descriptor and yet is flexible enough to capture appearance, articulation and viewpoint changes. It is furthermore based on a dense representation of multiple descriptors in order to aggregate different cues on objects. We include both hierarchical HOG (hHOG) [2] and



(a) (b) (c)

Figure 3. (a) Object instance. (b) Hierarchical HOG features of the instance weighted with parameters of our model. (c) Hierarchical HOG features of the instance weighted with linear SVM weights.

hierarchical bag-of-words (hBoW) [12] features to account for local and global representations of objects. In the following, we assume that each local classifier $F(\boldsymbol{\alpha}_n, f(\mathbf{x}))$ is actually the concatenation of a HOG and BoW classifier $F(\boldsymbol{\alpha}_n, f(\mathbf{x})) = \left(F^{\mathcal{H}}\left(\boldsymbol{\alpha}_n, f^{\mathcal{H}}(\mathbf{x})\right), F^{\mathcal{B}}\left(\boldsymbol{\alpha}_n, f^{\mathcal{B}}(\mathbf{x})\right)\right)$, which will be described in turn[2].

Hierarchical HOG descriptors. For computing the hierarchical HOG features, we compute a dense grid of non-overlapping cells of oriented gradients [2] over the image. As in [17], we extract multiple layers of those cell grids with increasing cell size at higher levels. Four neighboring cells are concatenated and normalized to one block, resulting in a dense grid of blocks (neighboring blocks overlap by 50%). We concatenate several blocks to form our local descriptors (details in Section 4). The global descriptor captures a holistic view on the object, since we concatenate all blocks of the bottom layer into a single global feature. The various descriptors are associated with the nodes of our model, indicated as green dots in Fig. 1.

Based on the local and global HOG descriptors, we train discriminative classifiers in order to represent local deformations as well as global statistics of objects. Therefore, we divide the grid of nodes in rectangular subregions (3×3 at the bottom layer) and train one SVM per subregion. Within the hierarchy, we reduce the number of subregions at higher levels: 2×2 at the second level and 1×1 at all other levels. Each classifier is a kernel-based SVM (*c.f*. Fig. 1):

$$F^{\mathcal{H}}\left(\boldsymbol{\alpha}_n^{\mathcal{H}}, f^{\mathcal{H}}(\mathbf{x})\right) = \sum_{\mathbf{s}\in S^n} \alpha_{n,\mathbf{s}}^{\mathcal{H}} K\left(\mathbf{s}, f^{\mathcal{H}}(\mathbf{x})\right) + \alpha_{n,0}^{\mathcal{H}}, \quad (4)$$

where S^n refers to the set of support vectors, K is an appropriate Mercer kernel, $\alpha_{n,\mathbf{s}}^{\mathcal{H}}$ denotes the support vector coefficients, and $\alpha_{n,0}^{\mathcal{H}}$ an offset. We employ linear kernels, though any Mercer kernel can be used.

In Fig. 3(b) we show the hierarchical HOG (hHOG) features of the shown object weighted with the parameters of our model and in Fig. 3(c) weighted with the weights of a linear SVM trained on the concatenation of all features. Note, with our model the real structure and shape is better represented, since our framework is able to learn feature couplings for capturing spatial dependencies of features.

[2]Here and in the remainder of the paper we drop the subscript i for parameters $\boldsymbol{\alpha}$ and feature functions f for notational simplicity.

Hierarchical BoW descriptors. For integrating a hierarchical bag of words (hBoW) approach [12] in our model we calculate SIFT descriptors [16] with radii $(5, 10, 15)$ and spacing of 10 pixels. Those descriptors are vector quantized into visual words with k-means clustering over the positive training instances ($k = 300$). We calculate one global BoW descriptor over the entire image and subsequently divide the image in regions according to the number of nodes of every level of our hierarchical model. In every subregion, we build a histogram of word occurrences and use it as the feature $f^{\mathcal{B}}(\mathbf{x})$. The hBoW features are also classified using a kernel-based SVM:

$$F^{\mathcal{B}}\left(\boldsymbol{\alpha}_n^{\mathcal{B}}, f^{\mathcal{B}}(\mathbf{x})\right) = \sum_{\mathbf{a}\in A^n} \alpha_{n,\mathbf{a}}^{\mathcal{B}} K\left(\mathbf{a}, f^{\mathcal{B}}(\mathbf{x})\right) + \alpha_{n,0}^{\mathcal{B}}, \quad (5)$$

where A^n refers to the set of support vectors, K is again a Mercer kernel, $\alpha_{n,\mathbf{a}}^{\mathcal{B}}$ denote the support vector coefficients, and $\alpha_{n,0}^{\mathcal{B}}$ is the offset.

Bootstrapping hard negatives. We bootstrap hard negative examples from the negative images and train the SVMs again with the additional negative images.

3. Model Learning

Given training data consisting of a set of images $\mathcal{X}$ and the corresponding set of node labels $\mathcal{Y}$, our goal is to estimate the model parameters $\theta = \{\boldsymbol{\alpha}, \mathbf{w}, \mathbf{e}\}$ and to identify a suitable graph structure represented by the active set $\mathcal{A}$.

3.1. Parameter learning

For now assuming a fixed graph structure $\mathcal{A}$, our goal is to train the parameters of the CRF model in a discriminative fashion. To that end, we consider the log-posterior of the parameters

$$\mathcal{L}(\theta) = \log P(\mathcal{Y}|\mathcal{X}; \theta, \mathcal{A}) + \log P(\theta), \quad (6)$$

which we aim to maximize. Here, $P(\theta) = P(\mathbf{w}) \cdot P(\mathbf{e})$ denotes a prior over the model parameters that regularizes parameter estimation to avoid overfitting. The SVM classifiers including the parameters $\boldsymbol{\alpha}$ are trained ahead of time decoupled from the rest of the model using standard quadratic programming, as *e.g*., in [7, 13, 22, 26]. This step attempts to optimally separate each object region from the background independently from other nodes in the hierarchy. Note that it would be also possible to train the $\boldsymbol{\alpha}$ during CRF training based on the primal form of the SVM (*c.f*. [21]), but we leave that for future work.

As usual in CRFs [11], it is not possible to find a closed form estimate for the parameters. Hence we rely on gradient ascent (see *e.g*. [15]) on the log-posterior to determine $\mathbf{w}$

and $\mathbf{e}$. Moreover, at each iteration we only consider a subset of the training data to improve efficiency, which yields a stochastic gradient ascent procedure.

Unary potentials. Assuming a Gaussian prior for the unary parameters ($P(\mathbf{w}) \sim \mathcal{N}(0,1)$), we derive the gradient of the log-posterior w.r.t. $\mathbf{w}_i$ as

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}_i} = \Big[\sum_{\mathbf{x} \in \mathcal{X}} E_{\mathcal{Y}|\mathbf{x}} \left[\mathbf{F}(\boldsymbol{\alpha}_i, f_i(\mathbf{x})) \cdot y_i \cdot \psi_i(y_i, \mathbf{x}) \right] - E_{P(\mathbf{y}|\mathbf{x})} \left[\mathbf{F}(\boldsymbol{\alpha}_i, f_i(\mathbf{x})) \cdot y_i \cdot \psi_i(y_i, \mathbf{x}) \right] \Big] - \mathbf{w}_i, \quad (7)$$

where $E_{\mathcal{Y}|\mathbf{x}}[\cdot]$ denotes the empirical expectation and $E_{P(\mathbf{y}|\mathbf{x})}[\cdot]$ denotes the expectation value under the posterior probability of our model. While the empirical expectation can be easily computed by plugging in the training label corresponding to $\mathbf{x}$, the expectation over the model distribution requires computing the marginal distribution $P(y_i|\mathbf{x})$. For a loopy graph as used here, this marginal cannot be computed in closed form. Consequently, we approximate it using loopy sum-product belief propagation (LBP) [27], as is widely done in the literature (*e.g.* [15]).

Note that learning the unary parameters corresponds to a simple form of structure learning that determines the relative importance of the features, much like multiple-kernel learning does in SVMs. Intuitively, the weight of a node should be small, if that node is classified incorrectly for most of the training instances. Otherwise, the weight should be high, if a node helps to discriminate foreground from background training instances.

Pairwise potentials. For the pairwise potentials, we proceed in a similar fashion. We put a Laplace prior $P(\mathbf{e}) \propto \exp(-||\mathbf{e}||)$ on the weights corresponding to a L1-regularization (see below), and derive the gradient of the log-posterior as:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{e}^{ij}_{y_i y_j}} = \Big[\sum_{\mathbf{x} \in \mathcal{X}} E_{\mathcal{Y}|\mathbf{x}} \left[(f_i(\mathbf{x}), f_j(\mathbf{x}))^T \phi_{ij}(y_i, y_j, \mathbf{x}) \right] - E_{P(\mathbf{y}|\mathbf{x})} \left[(f_i(\mathbf{x}), f_j(\mathbf{x}))^T \phi_{ij}(y_i, y_j, \mathbf{x}) \right] \Big] - \operatorname{sgn}(\mathbf{e}^{ij}_{y_i y_j}) \quad (8)$$

To compute the expectation over the model distribution, we require the marginals $P(y_i, y_j|\mathbf{x})$, which we again approximate using the beliefs from LBP.

The L1-regularization term not only avoids overfitting, but more importantly favors sparse solutions, where the majority of edges are inactive because of small weights [14]. Care needs to be taken near 0 since the L1-regularizer is non-differentiable there. We avoid numerical problems by approximating the L1-norm by $\sqrt{\|\mathbf{e}\|^2 + \epsilon}$.

3.2. Structure learning

The key contribution of our approach compared to other CRF models is that we not only learn the parameters, but also the appropriate graph structure. In particular, our goal is to find a sparse set of edges that best describes the relevant dependencies and feature interactions for a particular class of objects (we learn one active set $\mathcal{A}$ per class). Similar to [14], we do this in an iterative fashion, where at each iteration we add meaningful pairwise features to the active set $\mathcal{A}$ from the large pool of candidate edges (the inactive set $\mathcal{I}$) and simultaneously remove features from the model that have become irrelevant. Since any change of the graph structure may render the current set of parameters θ inappropriate, we interleave each update of the graph structure with parameter learning as described above (100 iterations of gradient ascent). The procedure starts with a disconnected graph ($\mathcal{A} = \emptyset$, $\mathcal{I} = \Omega$) and iteratively adds and removes edges.

Adding pairwise couplings. Since optimal feature selection is NP-hard, we use a gradient-based heuristic for estimating which feature most likely improves the model. We adapt the heuristic of [19], where at each step the feature with the largest likelihood-gradient is added to the active set. However, this method is only defined for generative models; here we carry this heuristic forward to discriminative structure learning with high-dimensional features. While such gradient-based heuristics are suboptimal, [14] showed that information gain based heuristics provide only slight improvements compared to gradient-based heuristics, but the latter are more efficient to compute.

The intuition behind this is that edge (i,j) with the largest log-likelihood gradient $\partial \log P(\mathbf{y} = \mathbf{1}|\mathbf{x}, \theta)/\partial \mathbf{e}^{ij}$ has the largest impact on changes of the target function (foreground likelihood) [19]. In a generative setting, this would help explaining the foreground object, because we can expect the highest increase in likelihood by adding that edge, and thus the largest improvement of the model. Here we take a discriminative approach instead, and not only look at the importance of explaining the object, but rather link the likelihood assuming object and the likelihood assuming background to each other. To that end, we consider the log-likelihood ratio $\left(\log \frac{P(\mathbf{y}|\mathbf{x},\theta)}{P(-\mathbf{y}|\mathbf{x},\theta)} \right)$, and find the edge from the inactive set that maximizes the log-likelihood ratio:

$$(i^*, j^*) = \operatorname*{arg\,max}_{(i,j) \in \mathcal{I}} \left\| \frac{\partial \mathcal{L}}{\partial \mathbf{e}^{ij}_{11}} - \frac{\partial \mathcal{L}}{\partial \mathbf{e}^{ij}_{-1-1}} \right\| \quad (9)$$

This criterion approximately finds the edge whose feature combination provides the largest improvement in discriminative power. The edge is subsequently added to the model ($\mathcal{A} \leftarrow \mathcal{A} \cup \{(i^*, j^*)\}$ and $\mathcal{I} \leftarrow \mathcal{I} \setminus \{(i^*, j^*)\}$).

So far, we argued for selecting edges according to Eq. (9), which requires computing the parameter gradient from Eq. (8). However, this is difficult to do as long as the edge is not added to the graph, but simply adding each potential candidate edge to the graph for computing

Eq. (8) is infeasible. The underlying issue is that we require probabilistic inference to compute the pairwise marginals $P(y_i, y_j|\mathbf{x})$ needed in Eq. (8). We can, however, approximate this pairwise marginal using LBP as described in [25]:

$$\tilde{b}_{ij}(y_i, y_j) \propto \psi_i(y_i, \mathbf{x}) \cdot \psi_j(y_j, \mathbf{x}) \cdot \phi_{ij}(y_i, y_j, \mathbf{x}) \cdot \prod_{s \in \Gamma_i \setminus j} M_{si}(y_i) \prod_{s \in \Gamma_j \setminus i} M_{sj}(y_j) \quad (10)$$

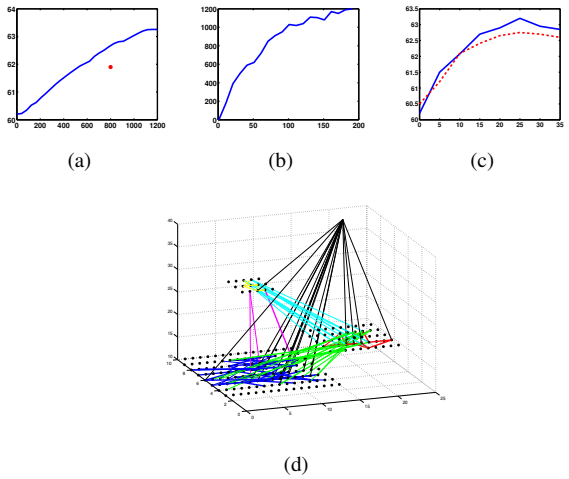
Here Γ_i refers to the neighborhood of i, *i.e.* all nodes in $\mathcal{A}$ that are connected to i, and $M_{si}(y_i)$ denotes the message that is passed from node s to node i.

Removing feature couplings. In order to avoid the model from becoming overly complex, which would make it inefficient and prone to overfitting, we follow two different strategies. The first is to use L1-regularization for the edge parameters, which encourages sparsity as discussed above. The other is to remove edges after each iteration of the structure learning procedure that are not crucial to the discriminative power. Whenever the weight of an active edge $(i, j) \in \mathcal{A}$ drops below a threshold ($\|\mathbf{e}^{ij}_{c_i c_j}\| \leq \tau_1$) and the weight gradient is below a threshold as well ($\|\frac{\partial \mathcal{L}}{\partial \mathbf{e}^{ij}_{c_i c_j}}\| \leq \tau_2$), we remove it from the active set ($\mathcal{A} \leftarrow \mathcal{A} \setminus \{(i, j)\}$ and $\mathcal{I} \leftarrow \mathcal{I} \cup \{(i, j)\}$). In this case the edge has no major influence on the log-likelihood ratio and since the gradient is small, one would expect the weights not to change significantly with more iterations of parameter learning. Thus, the edge and the coupling of the features can be removed without deteriorating the discriminative power considerably.

4. Experiments

We report experiments on the challenging PASCAL VOC 2007 dataset [3] to support our claims about the benefits of our structure learning approach. For all experiments we report the average precision (AP), the common evaluation criterion of the PASCAL challenge. Due to computational reasons we prefiltered object hypotheses $\tilde{\mathcal{X}}$ with the model of [4] and rescored them with our framework. Note, we do not leverage misclassifications of [4], but train our model on the provided training and validation bounding boxes and randomly cropped negative bounding boxes. Given our learned model (*i.e.* active edges and parameters $\boldsymbol{\alpha}, \mathbf{w}, \mathbf{e}$), for every $\tilde{\mathbf{x}} \in \tilde{\mathcal{X}}$ we compute the log-likelihood that the object of interest is present, $\log P(\mathbf{y} = \mathbf{1}|\tilde{\mathbf{x}})$, in the hypothesized bounding box $\tilde{\mathbf{x}}$ and use this as the score. Note that we do not perform inference during testing, which is due to the fact that we are interested in an efficient way of obtaining a detection score. Inference during testing would additionally allow us to obtain a segmentation.

In all experiments we used *SVM*^{light} [8] with linear kernels to train the parameters $\boldsymbol{\alpha}$. We did not add only a single edge per iteration but estimated and added the 20 best edges.



(a) (b) (c) (d)

Figure 4. (a) Comparison of the average precision for learned (blue) and fixed (red dot) structure. The learned structure is plotted *vs*. no. of edges. The fixed structure accounts for 800 edges. (b) No. of edges *vs*. no. of iterations. (c) AP vs. different percentage of connectedness for binary (blue) and multi-labels (red, dashed) (d) 2.5d visualization of the learned structure of our model.

In terms of pairwise parameters $\mathbf{e}$ we only optimize $\mathbf{e}_{+1,+1}$ and $\mathbf{e}_{-1,-1}$, and set $\mathbf{e}_{+1,-1} = \mathbf{e}_{-1,+1} = 0$, since we aim to classify whether bounding boxes contain the object or not. Thus, the case of changing signs is not represented in the training set and unlikely to appear during testing. Training the model takes approx. 10h, while calculating the score for one bounding box takes approx. 0.3s.

Feature descriptors. In our experiments the global HOG descriptor is the same as in [2], though we use different sizes of local HOG descriptors. They are specific to each object class and depend on the aspect ratio and the average size of the bounding box. We used sizes between 4×2 or 2×4 blocks and 9×5 or 5×9 blocks of local gradient histograms. Thus, each local descriptor covers an area between 40×24 and 80×48 pixels (or 24×40 and 48×80). The local descriptors are sampled densely over the bounding box and may overlap up to $\frac{2}{3}$. For the experiments using the hierarchical representation we deployed 3 levels of local descriptors and one global descriptor (see Fig. 4(d)).

PASCAL VOC 2006 motorbikes. A preliminary experiment on the motorbikes class of the PASCAL VOC 2006 challenge serves to shed light on the different aspects of our model. This dataset contains challenging multiscale, partially occluded, and multiview instances. We trained our model on the provided training and validation set. The results are summarized in Tab. 1 and detailed below. In Fig. 4(d) the most relevant feature couplings are shown. As it can be seen, our model includes short-range as well as long-range dependencies within but also between layers.

Our complete model (hHOG + hBoW features) yields

VOC 2006 motorbikes	lin. SVM/ HI SVM	Unary poten.	lin. SVM on unary	Our model structure lin. / HI
BoW	36.1 / 38.3	20.3	23.7	42.7 / **45.1**
hBoW	49.0 / 50.2	45.0	47.1	52.4 / **53.5**
HOG	49.1 / 50.0	47.3	48.5	51.0 / **53.3**
hHOG	60.1 / 61.0	59.1	60.0	62.8 / **63.4**
hHOG + hBoW	61.0 / 61.6	60.2	61.4	63.2 / **64.0**
train on [4]	-	-	-	**64.2** / -
sliding window	-	-	-	60.1 / -
MKL HI-kernel [1]	62.0	-	-	-
DPM [4]	58.2	-	-	-

Table 1. Summary of the results of different aspects of our model on the PASCAL VOC 2006 motorbikes. HI denotes the use of histogram intersection kernels.

a performance of 64.0% AP (histogram intersection kernel (HI)) and 63.2% (linear kernel), outperforming the baseline [4] (58.2%) by more than 5% AP. Multiple kernel learning with histogram intersection kernels [1] and the same features achieved 62.0%, which we outperform by 2% AP. This emphasizes the benefit of learning the structure of objects, since in [4] a fixed structure is assumed and in [1] no dependencies of features are learned. When we train on the output of [4] the performance of our model increases to 64.2% with linear kernels. When not using [4] as a pre-filter, but sliding window instead we achieve 60.1% still outperforming [4].

In Fig. 4(a) we compare our structure learning method *vs*. an instantiation with local, fixed pairwise couplings (as proposed in [21]), which amount to 800 pairwise edges. The model with fixed structure showed a performance of 61.9%, while our structure learning scheme achieved the same performance with fewer edges. When we look at the performance of structure learning with 800 automatically discovered edges, our framework achieved 62.7% AP.

For further investigating the stability of our model we experimented with different initializations of the active set $\mathcal{A}$ (empty and the structure of [21]), with different thresholds for removing edges and with different numbers of edges to be added to $\mathcal{A}$ in each iteration. For all these experiments our model learned similar structures and achieved similar performance. In Fig. 4(c) (blue line) we plot the performance vs. different degrees of connectedness (resulting from different thresholds). As it can be seen the performance does not differ dramatically for different thresholds when a certain level of connectedness is reached.

Furthermore, we compared our work against several baseline methods on the pre-filtered hypotheses of [4]: SVM classification (linear and HI kernels) on the concatenation of all features (column one of Tab. 1), unary classification alone (column two), and SVM classification on the output of our unary potentials (column three). SVM-based classification of the concatenation of our features showed 61.0% AP for linear kernels and 61.6% for HI kernels, which we outperform by 3.0% and 2.4% AP respectively. Concerning unary classification alone, we calculated only the unary potentials (*i.e*. no active edges) and added them up to one classification score yielding 60.2% AP. Compared to the latter, our complete model showed an improvement of 3.8% AP. In a different setting we train a support vector machine on the output of our unary potentials, yielding a comparable performance (61.4%) as pure SVM classification. Note that our model outperforms all other corresponding learning methods on the challenging dataset, which supports our claims about the flexibility and advantage of structure learning.

For further insights into our work, we evaluated our model when only using BoW features with one layer and with the hierarchy (hBoW), using only HOG features with one layer and with the hierarchy (hHOG). As can be seen in Tab. 1 our structure learning scheme consistently outperforms the other corresponding baseline models across all evaluated features. Note that using HI kernels consistently improves the performance compared to linear kernels.

Preliminary experiments with a multi-label setting as in [21] showed slightly worse performance than our binary label setting. Fig. 4(c) (red, dashed line) shows the performance vs. different degrees of connectedness.

PASCAL VOC 2007. In order to further support our claims about the advantages of our structure learning scheme, we evaluated our model using linear kernels on all 20 classes of the PASCAL VOC 2007 challenge. We compare our complete model using hierarchical HOG and hierarchical BoW features against using only hHOG features. Furthermore, we show the performance of the baseline of [4] and the best performance of the original challenge [3]. Note, we used [4] as baseline, since it is the leading model on the PASCAL dataset. All results are summarized in Tab. 2.

On average across classes, our model achieved a performance of 27.5% outperforming the baseline of [4] (25.9%) by 1.6% AP. Furthermore, our structure learning model consistently improves the detection performance of the baseline across all categories between 0.1% AP for chairs and 4.4% AP for horses. As can be seen in Fig. 2 our work is more flexible in terms of modeling different viewpoints, appearances, and articulated instances. Furthermore, the highest scored false positives of our model mainly account for misaligned bounding boxes containing the object of interest or sensible false alarms like bicycles recognized as motorbikes and cows recognized as horses. We conclude that our model helps in understanding the domain of interest and successfully discriminates object instances from background.

When comparing against the original VOC 2007 challenge, we achieved the best results for 16 of 20 classes. On average, we improved the best performance of the challenge (23.3%) by 4.2% AP. Note, in that measure we do not compare against one single model, but against the performance of the best model for every object class.

VOC 2007 (lin. kernels)	aero	bicyc	bird	boat	bottle	bus	car	cat	chair	cow
Our model hHOG+hBoW	**31.7**	**56.3**	1.7	**15.1**	**27.6**	**41.3**	**48.0**	15.2	9.5	**18.3**
Our model hHOG	30.0	56.1	1.5	15.0	27.2	41.1	47.5	14.5	9.5	18.1
DPM [4]	28.1	55.4	1.4	14.5	25.4	38.9	46.6	14.3	9.4	16.0
Best VOC07 [3]	26.2	40.9	**9.8**	9.4	21.4	39.3	43.2	**24.0**	**12.8**	14.0

	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	average
Our model hHOG+hBoW	**26.1**	11.3	**48.5**	**38.9**	**35.8**	**14.8**	**17.7**	**18.8**	**34.1**	**39.8**	**27.5**
Our model hHOG	25.2	10.8	47.3	37.4	35.5	13.7	16.3	18.6	32.4	37.6	26.8
DPM [4]	22.8	10.6	44.1	37.0	35.2	13.6	16.1	18.5	31.8	36.9	25.9
Best VOC07 [3]	9.8	**16.2**	33.5	37.5	22.1	12.0	17.5	14.7	33.4	28.9	23.3

Table 2. Results of our algorithm on the PASCAL VOC 2007 challenge.

Furthermore, we tested our complete model (hBoW and hHOG features) in comparison to only using hHOG features. On average, using hBoW and hHOG improves the performance of using only hHOG (26.8%) by 0.7% AP. Again, the complete model consistently shows equal (chairs) or better performance (all other classes) up to an improvement of 2.2% AP. Thus, including different features helps our framework to model complex object classes and increases the detection performance.

5. Conclusions

This paper presented a novel discriminative structure learning framework applied to hierarchical representations for object detection. Our model is defined as a structure learning extension to standard CRF models that allows to preserve the discriminative notion and increase the expressiveness of the model for object detection. The model is capable of capturing inherent structure of the domain of interest, as it flexibly learns local as well as long-range feature couplings. Paired with discriminative hBoW and hHOG based classification, our scheme lends itself to modeling the spatial layout of objects, which is crucial for detection in challenging real world scenes. As the experiments show, our model can represent a higher variation in viewpoint, appearance and articulation than the currently leading method on the PASCAL VOC challenge. In future work, we will explore global context information and other complementary features in our model. Furthermore, joint learning of all model parameters $(\boldsymbol{\alpha}, \mathbf{w}, \mathbf{e})$ will be investigated.

Acknowledgments. We thank Joris Mooij for making libDAI available online. This work has been funded, in part, by GRK 1362 of the German Research Foundation (DFG) and a Feodor Lynen Fellowship granted by the Alexander von Humboldt Foundation.

References

[1] A. Bosch, A. Zisserman, and X. Muoz. Image classification using ROIs and multiple kernel learning. *IJCV*, 2008. Submitted.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR'05*.

[3] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL VOC challenge 2007.

[4] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR'08*.

[5] R. Fergus, A. Zisserman, and P. Perona. Object class recognition by unsupervised scale invariant learning. In *CVPR'03*.

[6] K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *JMLR*, 8:725–760, 2007.

[7] D. Hoiem, C. Rother, and J. Winn. 3D layout CRF for multi-view object class recognition and segmentation. In *CVPR'07*.

[8] T. Joachims. *Making Large-Scale SVM Learning Practical*. Advances in Kernel Methods - Support Vector Learning, 1999.

[9] A. Kapoor and J. Winn. Located hidden random fields: Learning discriminative parts for object detection. In *ECCV'06*.

[10] S. Kumar, J. August, and M. Hebert. Discriminative random fields. *IJCV*, 68(2):179–201, 2006.

[11] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML'01*.

[12] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bag of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR'06*.

[13] C. H. Lee, R. Greiner, and O. Zaianen. Efficient spatial classification using decoupled conditional random fields. In *PKDD'06*.

[14] S.-I. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of Markov networks using L1-regularization. In *NIPS'06*.

[15] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. In *ECCV'06*.

[16] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[17] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR'08*.

[18] S. Parise and M. Welling. Structure learning in Markov random fields. In *NIPS'06*.

[19] S. Perkins, K. Lacker, J. Theiler, I. Guyon, and A. Elisseeff. Grafting: Fast, incremental feature selection by gradient descent in function space. *JMLR*, 3:1333–1356, 2003.

[20] M. Schmidt, K. Murphy, G. Fung, and R. Rosales. Structure learning in random fields for heart motion abnormality detection. In *CVPR'08*.

[21] P. Schnitzspan, M. Fritz, and B. Schiele. Hierarchical support vector random fields: Joint training to combine local and global features. In *ECCV'08*.

[22] 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'06*.

[23] D. Tran and D. Forsyth. Configuration estimates improve pedestrian finding. In *NIPS'07*.

[24] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *ICCV'07*.

[25] M. J. Wainwright, T. Jaakkola, and A. S. Willsky. Tree-based reparameterization for approximate estimation on graphs with cycles. In *NIPS'02*.

[26] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR'06*.

[27] J. Yedidia, W. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *Exploring Artificial Intelligence in the New Millennium*. 2003.

[28] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 73(2):213–238, 2007.