Tree-structured CRF Models for Interactive Image Labeling

Thomas Mensink, Jakob Verbeek, Gabriela Csurka

Tree-structured CRF Models for Interactive Image Labeling

Thomas Mensink, *Student Member, IEEE,* Jakob Verbeek, *Member, IEEE,* and Gabriela Csurka

Abstract—We propose structured prediction models for image labeling that explicitly take into account dependencies among image labels. In our tree structured models, image labels are nodes, and edges encode dependency relations. To allow for more complex dependencies, we combine labels in a single node, and use mixtures of trees. Our models are more expressive than independent predictors, and lead to more accurate label predictions. The gain becomes more significant in an interactive scenario where a user provides the value of some of the image labels at test time. Such an interactive scenario offers an interesting trade-off between label accuracy and manual labeling effort. The structured models are used to decide which labels should be set by the user, and transfer the user input to more accurate predictions on other image labels. We also apply our models to attribute-based image classification, where attribute predictions of a test image are mapped to class probabilities by means of a given attribute-class mapping. Experimental results on three publicly available benchmark data sets show that in all scenarios our structured models lead to more accurate predictions, and leverage user input much more effectively than state-of-the-art independent models.

Index Terms—I.5.4.b Pattern Recognition Application Computer vision, I.5.5.a Pattern Recognition Interactive systems, I.4.8.e Object Recognition, H.3.1 Content Analysis and Indexing, I.5.1.e Statistical Pattern Recognition.

1 Introduction

Image labeling, image classification and image auto-annotation share the same goal of predicting the relevant terms from a given annotation vocabulary for a specific image. The label predictions are used for clustering, (attribute-based) classification, and image retrieval, hence they are an important source for any multimedia content management system, personal and stock photography database indexing, or photo sharing on social networks.

Most existing systems address the problem of image annotation either in a fully manual way (*e.g.* stock photo sites as Getty images), or in a fully automatic setting where image labels are automatically predicted without any user interaction. In the latter case most commonly used are either classifiers *e.g.* [1], ranking models *e.g.* [2], or nearest neighbor predictors [3]. While these methods (in general) do not explicitly model dependencies among the image labels, there are correlations in the classifier outputs, since the independent predictors use the same images to train/predict these labels.

In this paper we differentiate from this predominant line of work in two ways. First, we propose structured models that take into account the dependencies among the image labels explicitly. Since these models are more expressive, they lead to more accurate image label predictions. Second, we follow an interactive labeling scenario, where a user is asked to confirm or reject, at test time, some of the image labels. Such an interactive scenario is for example useful when indexing images for stock photography, where a high indexing quality is mandatory, yet fully manually indexing is very expensive and suffers from very low throughput.

The interactive scenario offers an interesting trade-off between accuracy and manual labeling effort. In this case the label dependencies in the proposed models can be leveraged in two ways. First, the structured models are able to transfer the user input for one image label to more accurate predictions on other image labels, which is impossible with independent prediction models. Second, using structured models, the system will not query, wastefully, for image labels that are either highly dependent on already provided labels, or predicted with high certainty from the image content. Through inference in the graphical model, the system fuses the information from the image content and the user responses, and is able to identify labels that are highly informative once provided by the user.

We conduct experiments using three public benchmark data sets: the Scene Understanding data set [4] (SUN'09), the data set of the ImageCLEF'10 Photo Annotation Task [5] (ImageCLEF), and the Animals with Attributes data set [6] (AwA). Our results without user input are comparable to the state-of-the-art reported on these data sets. The experiments also show that a relatively small amount of user input can substantially improve the results, in particular when we use our proposed models that capture label dependencies. To give an idea of the impact of user input, we illustrate the interactive image annotation process for two example images in Figure 1.

In addition to showing the effectiveness of structured models for interactive image labeling, we also explore how

- *Thomas Mensink works at XRCE and LEAR, Jakob Verbeek works at LEAR, and Gabriela Csurka works at XRCE.*
- *Xerox Research Centre Europe*
  E-mail: firstname.lastname@xrce.xerox.com
- *LEAR Team - INRIA Rhone Alpes*
  E-mail: firstname.lastname@inria.fr

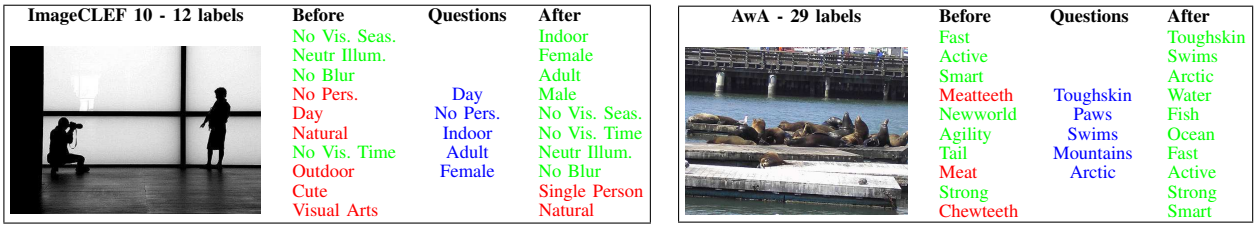

Fig. 1: Interactive image annotation for images from the ImageCLEF 2010 data set (left, with 12 true labels), and the AwA data set (right, with 29 true labels). We show the labels with highest confidence before and after user input (green labels are correct, red ones not), as well as the five labels selected by the system to be set by the user (blue).

the proposed structured models can be exploited in the context of attribute-based image classification [6], [7]. The attributes are shared between different classes and image classification is based on a given attribute-to-class mapping. Hence, attribute values are first predicted for the image and then the attribute-to-class mapping is used to obtain the class probabilities. Predicting the attribute values for an image can be seen as annotating an image with a set of (attribute) labels, therefore we use our structured models at the attribute level. The user interaction will also take place at the attribute level, but in this case the system will ask attribute labels as user input to improve the class predictions rather than the attribute predictions. Experiments on the AwA data set show that, also in this case, the structured models outperform independent attribute prediction, both in automatic and interactive scenarios. Furthermore, a small amount of user input on the attributes substantially improves the classification results.

This paper extends our earlier work [8], by (i) introducing a kernel based learning approach, which allows to learn all parameters in the model at once, (ii) proposing different strategies to obtain tractable structures for the tree based on mutual information and gradient information, (iii) extending accordingly the experimental evaluation, by comparing the different extensions, by adding further comparisons to the state-of-the art, and by testing our methods in a multi-word query based retrieval scenario.

The rest of the paper is organized as follows. In Section 2, we discuss how our work is related to recent work on image classification and annotation. Then, we present our structured prediction model in Section 3, and its extension to multi-label nodes in Section 4. Section 5 describes how to apply the structured models for attribute-based image classification. Finally, we present extended experimental results in Section 6, and our conclusions in Section 7.

2 Related Work

The dominant line of research for image annotation, object category recognition, and image categorization has focused on methods that deal with one label or object category at a time. The function that scores images for a given label is obtained by means of various machine learning algorithms, such as binary SVM classifiers using different (non-)linear kernels [1], [9], [10], nearest neighbor classifiers [3], [11], and ranking models trained for retrieval [2] or annotation [12]. Classification is more challenging when dealing with many classes, both when the aim is to assign a single label to an image from many possible ones [13], as well when for each image several labels should be predicted, *e.g.* all present object categories [4].

To address the latter, there has been a recent focus on contextual modeling. For example in object class recognition, the presence of one class may suppress (or promote) the presence of another class that is negatively (or positively) correlated, see *e.g.* [4], [14], [15]. In [15] the goal is to label the regions in a pre-segmented image with category labels, and a fully-connected conditional random field model over the regions is used. In [14] a contextual model is proposed to filter the windows reported by object detectors for several categories. The contextual model includes terms for each pair of object windows that will suppress or favor spatial arrangements of the detections (*e.g.* *boat* above *water* is favored, but *cow* next to *car* is suppressed). A similar goal is pursued in [4], where the scores of bounding boxes obtained by discriminatively trained object detectors is enhanced using a tree-structured model. This tree models the presence and location of the object category in the context of all other bounding boxes from the image. The parameters of the tree are learned in a generative way, from images with bounding-boxes. In our work, we also use tree structured models, but over global labels using only presences and absences of the labels, and we learn the complete model discriminatively.

The interactive image annotation scenario we address in this paper is related to active learning. In general, active learning systems attempt to overcome the labeling bottleneck, *i.e.* manually labeling thousands of images for each concept [16]. In active learning for classification, the learning algorithm disposes of a number of labeled and unlabeled examples. Iteratively, a classification model is learned from the labeled ones, and then using the learned model, the system determines which example (image) is the most valuable to be labeled next by the user. Such models have been used to learn from user input at different levels of granularity, *e.g.* by querying image-wide labels or precise object segmentation [17]. In our work, however, the system does not select images to be labeled at training time by a user to improve the model, but we assume that the training set is fully labeled. Instead, for a given image at test time, our system selects labels for which user-input is the most valuable in order to improve predictions on the other labels of the same image.

We also apply our approach to attribute-based image

classification, where an image is assigned to a given class based on a set of given attributes [6], [7]. The advantages of such a system are that (i) it can recognize unseen classes, based on an attribute-level description, (ii) the attribute representation can in principle encode an exponential number of classes, and (iii) more training examples are available for the attributes since they are shared across classes.

In [7] a discriminative SVM object recognition system is combined with a generative class-attribute model: for each class the object attributes values reported by different users (allowing for erroneous user responses and ambiguous object-attribute relationships) are modeled independently. To leverage user input for classification, the system asks the user to label the attribute that reduces the entropy on the class label the most. Similarly, we also exploit user input at the level of attributes, but we learn recognition models for each attribute rather than for the object categories. This has the advantage that it allows for recognition of classes for which no training images are available, but only an attribute-based description is known, *i.e.* zero-shot classification [6]. As compared to the model of [6], we go one step further by modeling the dependencies between attribute labels. This allows us to improve the attribute-based recognition, but also to better exploit the user input by asking more informative questions.

3 Structured Annotation Models

Our goal is to obtain an expressive model that captures dependencies between the different image labels, but which still allows for tractable inference. To this end, we define a conditional random field model, where each node represents a label from the annotation vocabulary, and edges between nodes represent interaction terms between the labels.

Let $\boldsymbol{y} = (y_1, \ldots, y_L)^\top$ denote a vector of the L binary label variables, *i.e.* $y_i \in \{0, 1\}$. We use the Gibbs distribution to define the probability for a specific configuration $\boldsymbol{y}$ given the image $\boldsymbol{x}$:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\big(-E(\boldsymbol{y}, \boldsymbol{x})\big), \tag{1}$$

where $Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}} \exp\big(-E(\boldsymbol{y}, \boldsymbol{x})\big)$ is an image-dependent normalizing term known as the partition function, and $E(\boldsymbol{y}, \boldsymbol{x})$ is an energy function scoring the compatibility between an image $\boldsymbol{x}$ and a label vector $\boldsymbol{y}$. The binary label case, where each concept is either relevant or not relevant given an image, can be trivially extended to cases where labels can take three or more values.

The tractability for inference of these models depends on the complexity of computing the partition function, which in turn depends on the structure of the energy function. Inference is used to find marginal distributions on individual labels $p(y_i|\boldsymbol{x})$, the pairwise marginals $p(y_i, y_j|\boldsymbol{x})$, and the most likely joint labeling state $\boldsymbol{y}^* = \arg\max_{\boldsymbol{y}} p(\boldsymbol{y}|\boldsymbol{x})$.

Using our probabilistic formulation, label prediction and elicitation (*i.e.* selecting labels to be set by a user) are handled naturally using marginal probabilities and label entropy. In principle, the proposed models can also be formulated in a max-margin framework [18], but then it is less clear how to define label elicitation strategies.

3.1 Tree-structured models on image labels

We start with using tree-structured conditional random fields, since inference in tree models is tractable and can be performed by standard belief propagation algorithms [19].

The trees are defined such that each node represents a single label, and $\mathcal{E} = \{e_1, \ldots, e_{L-1}\}$ defines the edges in the tree over the label variables, where $e_l = (i, j)$ indicates the presence of an edge between y_i and y_j. For now we assume a given set of edges; in Section 3.1.2 we detail different approaches to obtain a tree structure. The energy for a configuration of labels $\boldsymbol{y}$ for an image $\boldsymbol{x}$ is given by:

$$E(\boldsymbol{y}, \boldsymbol{x}) = \sum_{i=1}^{L} \psi_i(y_i, \boldsymbol{x}) + \sum_{(i,j)\in\mathcal{E}} \psi_{ij}(y_i, y_j). \tag{2}$$

For the unary terms we use generalized linear functions:

$$\psi_i(y_i = l, \boldsymbol{x}) = \boldsymbol{\phi}_i(\boldsymbol{x})^\top \boldsymbol{w}_i^l, \tag{3}$$

where $\boldsymbol{\phi}_i(\boldsymbol{x})$ is a feature vector for the image which may depend on the label index i (see Section 3.1.1), and $\boldsymbol{w}_i^l$ is the weight vector for state $l \in \{0, 1\}$.

The pairwise potentials, defined by a scalar parameter for each joint state of the corresponding nodes, are independent of the image input:

$$\psi_{ij}(y_i = s, y_j = t) = v_{ij}^{st}. \tag{4}$$

Having a particular tree structure, we learn the parameters of the unary and pair-wise potentials by the maximum likelihood criterion. Given N training images $\boldsymbol{x}_n$ and their ground-truth annotations $\boldsymbol{y}_n$, we seek to maximize:

$$\mathcal{L} = \sum_{n=1}^{N} \mathcal{L}_n = \sum_{n=1}^{N} \ln p(\boldsymbol{y}_n|\boldsymbol{x}_n). \tag{5}$$

As the energy function is linear in the parameters, the log-likelihood function is concave and the parameters can be optimized using gradient-based methods. Computing the gradient requires evaluation of the marginal distributions on single variables and pairs of variables connected by edges in the tree. Using y_{in} to denote the value of variable y_i for training image n, we have:

$$\frac{\partial \mathcal{L}_n}{\partial \boldsymbol{w}_i^l} = \Big(p(y_i = l|\boldsymbol{x}_n) - [\![y_{in} = l]\!]\Big)\boldsymbol{\phi}_i(\boldsymbol{x}_n), \tag{6}$$

$$\frac{\partial \mathcal{L}_n}{\partial v_{ij}^{st}} = p(y_i = s, y_j = t|\boldsymbol{x}_n) - [\![y_{in} = s, y_{jn} = t]\!], \tag{7}$$

where we use the Iverson bracket notation, *i.e.* $[\![\cdot]\!]$ equals 1 if the expression is true, and 0 otherwise.

3.1.1 Unary Potentials

In this section we describe two unary potential functions we considered. The first uses a very compact feature vector based on classifier outputs. In the second case a full kernel learning method is applied using directly the (high-dimensional) feature vector representation of the images.

Classifier outputs: For the sake of efficiency, we can use very compact feature functions $\boldsymbol{\phi}_i(\boldsymbol{x}_n) = [s_i(\boldsymbol{x}_n), 1]^\top$, where $s_i(\boldsymbol{x})$ is an SVM score function associated with label variable y_i. This is a two-stage learning approach, which has the advantages that it allows for a flexible choice in the used classifier, and often for faster training, since the number of free parameters in the CRF is limited [20].

Kernel-based representation of unary potentials: Alternatively, we can set $\phi_i(\boldsymbol{x}_n) = \phi(\boldsymbol{x}_n)$, a high-dimensional feature vector of image n, *e.g.* a bag-of-words representation [21], GIST descriptor [22], or Fisher vector representation [10]. Since $\phi(\boldsymbol{x}_n)$ is now high-dimensional, we need to avoid overfitting by minimizing the negative log-likelihood plus a regularization term on the unary weights:

$$F = -\sum_n \ln p(\boldsymbol{y}_n|\boldsymbol{x}_n) + \lambda\, \Omega(\boldsymbol{w}), \tag{8}$$

where λ is the trade off parameter between the regularization term and log-likelihood.

From (6), we observe that $\boldsymbol{w}_i^l$ will be in the span of the data, hence $\boldsymbol{w}_i^l = \sum_m \alpha_{im}^l \phi(\boldsymbol{x}_m)$. Therefore, we can write:

$$\begin{aligned}\psi_i(y_i{=}l, \boldsymbol{x}_n) &= \sum_m \alpha_{im}^l \phi(\boldsymbol{x}_m)^\top \phi(\boldsymbol{x}_n), \\ &= \sum_m \alpha_{im}^l K_{mn} = \boldsymbol{k}_n^\top\ \boldsymbol{\alpha}_i^l,\end{aligned} \tag{9}$$

where K is the kernel matrix with $K_{mn} = \phi(\boldsymbol{x}_m)^\top \phi(\boldsymbol{x}_n)$, $\boldsymbol{k}_n$ denotes its n-th column and $\boldsymbol{\alpha}_i^l$ is the coefficient vector. Clearly, non-linear kernels, such as the RBF or intersection kernels, can be used as well.

The ℓ_2 regularizer to the unary weights can be written in terms of the kernel as

$$\Omega(\boldsymbol{w}) = \sum_{i,l} \frac{1}{2} ||\boldsymbol{w}_i^l||_2^2 = \sum_{i,l} \frac{1}{2} (\boldsymbol{\alpha}_i^l)^\top K \boldsymbol{\alpha}_i^l. \tag{10}$$

We use a gradient descent algorithm to minimize (8). For the interaction terms Eq. (7) still holds, while for $\boldsymbol{\alpha}_i^l$ the derivatives are given by:

$$\frac{\partial \mathcal{L}_n}{\partial \boldsymbol{\alpha}_i^l} = \Big(p(y_i = l|\boldsymbol{x}_n) - [\![y_{in} = l]\!]\Big)\boldsymbol{k}_n, \tag{11}$$

$$\frac{\partial \Omega}{\partial \boldsymbol{\alpha}_i^l} = \lambda\ K \boldsymbol{\alpha}_i^l. \tag{12}$$

3.1.2 Obtaining a tree structure

The interactions between the labels are defined by the structure of the tree. While all labels will interact with each other in the structure, labels which are close have more influence on each other. Finding the optimal tree structure for conditional models is generally intractable [23], therefore we resort to approximate methods for finding useful tree structures over the labels. We compare two methods to obtain a tree structure.

Mutual information: We consider using the optimal tree structure for a generative model. This structure can be found using the Chow-Liu algorithm [24] as follows. We define a fully connected graph over the label variables with edge weights given by the mutual information between the label variables, where the mutual information between pairs of label variables is estimated from the empirical distribution of the labels in the training data. The optimal tree-structure for a generative model is then given by the maximum spanning tree in this graph.

Gradient based: We consider to obtain the tree structure by iteratively growing a tree, starting from a completely disconnected graph. In each iteration we (i) add a single edge to the tree based on the current gradient, and (ii) learn the parameters of the current graph, to maximize Eq. (5). We repeat this process until a tree model which spans over all nodes is obtained. Note that using Eq. (7), we can compute the gradient for any edge, including ones that are not used in the current model. As an indicator of the increase in log-likelihood, which we could obtain by including a particular edge, we use the ℓ_2 norm of the gradient w.r.t. the parameters of that edge. This is motivated by the fact that the ℓ_2 norm of the gradient is proportional to the increase in the log-likelihood by taking an infinitesimal step in the gradient direction.

3.2 Label elicitation for image annotation

In the interactive image annotation scenario, a user is asked iteratively to reveal the value of a selected label. While a random choice of labels is a possibility, we show in Section 6 that this is far from optimal. We propose a label selection strategy whose aim is to minimize the uncertainty of the remaining labels given the test image. The proposed strategy resembles query strategies used in active learning [16], and the maximum information gain criterion [7].

Our goal is to select the label y_i for which knowing its ground truth value minimizes the uncertainty on the other labels. To achieve this, we propose to minimize the entropy of the distribution on the label vector $\boldsymbol{y}$ given the user input for one label y_i, by varying i which indicates which label will be set by the user.

Let us use y_i^l to denote $y_i = l$, and $\boldsymbol{y}_{\setminus i}$ to denote all label variables except y_i. Since the value of y_i is not known prior to the moment that it is set by the user, we evaluate the expected conditional entropy,

$$H(\boldsymbol{y}_{\setminus i}|y_i, \boldsymbol{x}) = \sum\nolimits_l p(y_i = l|\boldsymbol{x}) H(\boldsymbol{y}_{\setminus i}|y_i^l, \boldsymbol{x}), \tag{13}$$

where

$$H(\boldsymbol{y}_{\setminus i}|y_i^l, \boldsymbol{x}) = -\sum\nolimits_{\boldsymbol{y}_{\setminus i}} p(\boldsymbol{y}_{\setminus i}|y_i^l, \boldsymbol{x}) \ln p(\boldsymbol{y}_{\setminus i}|y_i^l, \boldsymbol{x}). \tag{14}$$

Using the fact that $H(\boldsymbol{y}|\boldsymbol{x})$ does not depend on the selected variable y_i, and given the basic identity of conditional entropy, see *e.g.* [19], we have

$$H(\boldsymbol{y}|\boldsymbol{x}) = H(y_i|\boldsymbol{x}) + H(\boldsymbol{y}_{\setminus i}|y_i, \boldsymbol{x}). \tag{15}$$

We hence conclude that minimizing Eq. (13) for y_i is equivalent to maximizing $H(y_i|\boldsymbol{x})$ over i. Hence, we select the label variable $y_{i'} = \operatorname{argmax}_i H(y_i|\boldsymbol{x})$.

In order to select a collection of labels to be set by the user, we proceed sequentially by first asking the user to set only one label. We then repeat the procedure while conditioning on the labels already provided by the user. Another

State	Marginal	Nature	Sky	Clouds
1	3.4 %	0	0	0
2	0.0 %	0	0	1
3	9.8 %	0	1	0
4	59.9 %	0	1	1
5	0.4 %	1	0	0
6	0.0 %	1	0	1
7	2.6 %	1	1	0
8	23.9 %	1	1	1
Marginal on label		26.9%	96.2%	83.8%

Fig. 2: Example of a compound variable that combines three image labels and has $2^3 = 8$ states. The marginals for the individual labels are obtained by summing the marginal probabilities of the corresponding joint states.

possibility is to select a group of labels at once, which is nevertheless suboptimal as it cannot leverage information contained in the user input in the selection procedure. To compute the label marginals while conditioning on the user input, we introduce additional unary potentials that assign zero energy to the label value given by the user and infinite energy to the values incompatible with the user input.

For interactive image labeling it is interesting to evaluate the proposed methods using a user study, where several people are asked to annotate images using the proposed methods. In such a real life setting, the model should allow for ambiguous user annotations as in [7]. However, this falls beyond the scope of this paper.

4 Extensions of the Basic Model

While the tree structured models of Section 3 have the advantage to allow for tractable inference, they are limited in the dependencies they can model. We now propose two extensions that allow for more dependencies, and maintain tractable inference. First, we introduce a graphical model that is a tree over groups of label variables. Second, we consider mixture-of-trees structured models.

4.1 Trees over groups of label variables

To accommodate for more dependencies between labels in the model, we consider the extension where we group label variables, and then define a tree over these groups. A label group can be seen as a fully connected set of variables in the graphical model. A tree model over those groups implies that the underlying cyclic graphical model has a certain structure, it contains only local cycles, *i.e.* only cycles within each label group, and among neighboring groups in the tree, see Figure 3 for an example. The model remains tractable as long as it has a low treewidth [19].

We determine a group size k, and model each state of the labels explicitly as a state of the compound node, which has 2^k states; see Figure 2. If k equals the number of labels L, we have the fully connected model, in which inference is intractable. The group size k relates to the treewidth of the graphical model, and offers a trade-off between expressiveness of the model, computational tractability and the risk of over-fitting on the training data.

Using belief propagation we now obtain node marginals *i.e.* probabilities for each state of a node. However, we are still interested in the probability of label i being true for this image, *i.e.* $p(y_i = 1|\boldsymbol{x})$, since this label marginal is used to rank images for a specific label, to sort labels for a specific image, and for label elicitation. The label marginals are trivially obtained by summing the right entries of the node marginal; see Figure 2.

Grouping labels: To obtain a partitioning of the labels, we perform agglomerative clustering based on mutual information, fixing in advance a maximum group size k. In each step, we merge the label groups that have the maximum mutual information, while allowing at most k labels per group. In the final partitioning each label l is assigned to a single group g and no group is larger than k labels. With each group of variables, we associate a new variable $\boldsymbol{y}_g$ that takes as values the product space of the values of the labels in the groups.

The unary potentials are defined as in Eq. (3), where y_i is replaced with $\boldsymbol{y}_g$, and hence take one of the 2^k states according to the values that labels in the group can take. For each state l of the joint-node g a weight vector $\boldsymbol{w}_g^l$ is learned. When we use the pre-trained SVM scores as feature vector, we define $\boldsymbol{\phi}_g(\boldsymbol{x}) = [\{s_i(\boldsymbol{x})\}_{i \in \mathcal{G}_g}, 1]$ as the extended vector of SVM scores associated with the image labels in the group $\mathcal{G}_g$. The pairwise potential of Eq. (4) now links groups of k binary variables, and hence will be defined by 2^{2k} scalars. Therefore, the cost of message passing algorithms scales with $O(G2^{2k})$, where G is the number of groups. In order to maintain tractable inference, the group sizes should be fairly small ($k \leq 4$ in our experiments).

We determine a tree structure on the compound nodes using the same ideas as in Section 3.1.2. In Figure 3 we show a tree with group size 3, obtained with the Chow-Liu algorithm using the mutual information criteria. Although not forced, semantically related concepts are often grouped together (*e.g.* water related concepts in the *Water-River-Sea* node and plant related concepts in the *Plants-Flowers-Trees*) or they are in neighboring nodes (*e.g.* person related concepts around the *Single Person-No Person-Male* node).

Conditioning on user input: In order to compute the marginals when one or more labels have been set by a user, we add an additional unary term per node, which value depends on the user input. For compound nodes with $k > 1$ labels, we add zero energy to all joint-states that are compatible with the user input, and infinite energy to those that are not. In the example of Figure 2, if a user would set *Sky=true*, this would incur infinite energy for states $1, 2, 5$, and 6 of the 3-label node.

4.2 Mixture-of-trees

As a second extension, we consider mixture-of-trees to allow for more label dependencies. Mixtures are defined either over trees with different group sizes k or over trees with different structures over a fixed set of nodes. A mixture

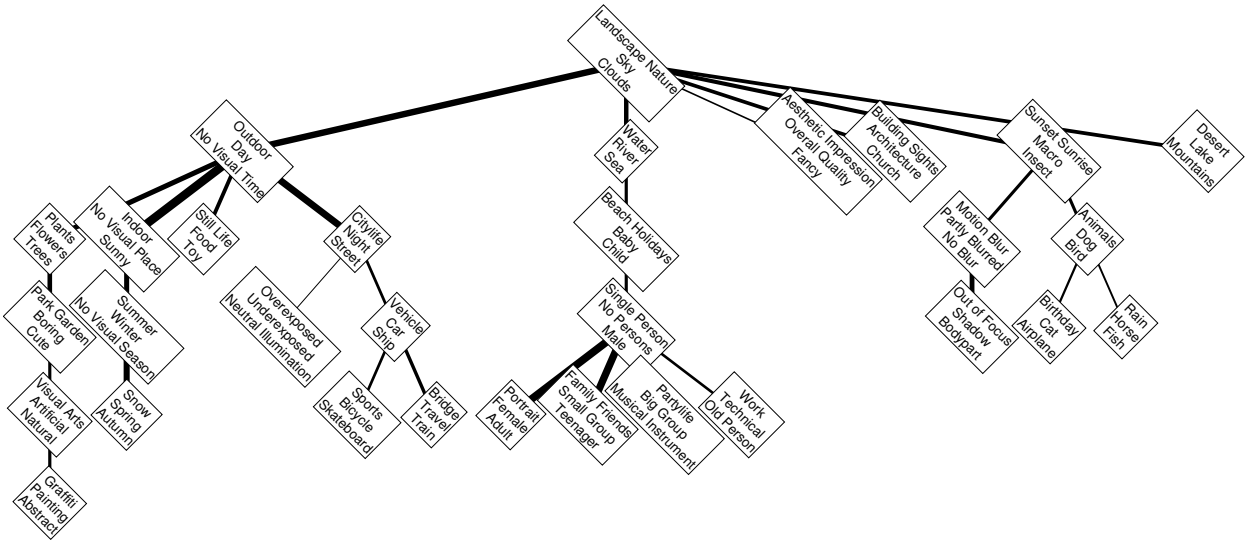

Fig. 3: An example of a tree over compound nodes with $k = 3$ labels on the $L = 93$ labels of the ImageCLEF data set. The edge thickness is proportional to the mutual information between the linked nodes. The root of the tree has been chosen as the vertex with highest degree.

of T different trees, indexed by t, is defined as:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{t=1}^{T} \pi_t p_t(\boldsymbol{y}|\boldsymbol{x}), \tag{16}$$

where π_t denotes the mixing weight, and $p_t(\boldsymbol{y}|\boldsymbol{x})$ denotes the different tree-structured models.

The label marginals $p(y_i|\boldsymbol{x})$ are in this case obtained as the "mixture of the marginals" computed in the component models. This is easily seen from the following identities:

$$\begin{aligned} p(y_i|\boldsymbol{x}) &= \sum_{\boldsymbol{y}\setminus i} p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\boldsymbol{y}\setminus i} \sum_t \pi_t p_t(\boldsymbol{y}|\boldsymbol{x}) \\ &= \sum_t \pi_t \sum_{\boldsymbol{y}\setminus i} p_t(\boldsymbol{y}|\boldsymbol{x}) = \sum_t \pi_t p_t(y_i|\boldsymbol{x}). \end{aligned} \tag{17}$$

In the first and last equations we use the definition of the marginal probability, in the second we use the definition of the mixture, and in the third we swap the two sums.

We train each tree model independently, and then average the predictions of the individual trees using $\pi_t = 1/T$. Alternatively, the mixing weights can be learned concurrently while learning the trees, *e.g.* by using the EM algorithm to infer which tree corresponds to which image, possibly improving results.

Our mixture-of-trees model is related to [25], where a mixture over random spanning trees is used for approximate learning and inference in a single underlying intractable CRF model. Different from their work, we perform inference and learning independently in each tree, and mix maximum spanning trees of different node sizes.

5 Attribute-based classification

In this section we consider how our structured prediction models can be used for attribute-based image classification, which refers to a classification paradigm where an image is assigned to a given class $z \in \{1, \ldots, C\}$ based on a set of attribute values [6], [7]. An image belongs to exactly one class, but attributes are shared between different classes. For example, in the Animals with Attributes (AwA) data set [6] different animals are defined in terms of attributes such as *has stripes, has paws, swims*, etc.

The advantages of an attribute-based classification system are that it can recognize unseen classes based on an attribute-level description only, and that the attribute representation can in principle encode an exponential number of classes. By sharing the attributes between different classes, classifiers for each of the attributes can be learned by pooling examples of different classes which increases the number of positive training examples per attribute as compared to the number of positive examples available for the individual classes.

5.1 Structured attribute prediction

We apply our structured prediction model at the level of attributes, *i.e.* we learn a tree structured model over attributes, and the binary values y_i now refer to the presence or absence of an attribute for an image. As in [6], we assume that the deterministic mapping between attributes and the C object (animal) classes is given, and denote the attribute configuration of class c by $\boldsymbol{y}_c$.

We define the distribution over classes by normalizing the likelihoods of the corresponding attribute configurations:

$$p(z=c|\boldsymbol{x}_n) = \frac{p(\boldsymbol{y}_c|\boldsymbol{x}_n)}{\sum_{c'} p(\boldsymbol{y}_{c'}|\boldsymbol{x}_n)} = \frac{\exp(-E_{nc})}{\sum_{c'} \exp(-E_{nc'})}, \tag{18}$$

where $E_{nc} = E(\boldsymbol{y}_c, \boldsymbol{x}_n)$. Note that the evaluation of $p(z|\boldsymbol{x})$ does not require belief-propagation: it suffices to evaluate $E(\boldsymbol{y}_c, \boldsymbol{x})$ for the C attribute configurations $\boldsymbol{y}_c$, since the partition function $Z(\boldsymbol{x})$ cancels out.

5.2 Correction Terms

When using our model as such, we observe that some classes tend to be much more often predicted than others, and the prediction errors are mainly caused by assigning images to these over-predicted classes. As this also holds for the independent attribute prediction model, we assume the reason might be that some classes have rare (combinations of) attribute values.

In order to overcome this, we introduce a correction term u_c for each class that plays a similar role as a class prior probability in a generative probabilistic model. We redefine the class prediction model of Eq. (18) as

$$\tilde{p}(z{=}c|\boldsymbol{x}_n) = \frac{\exp\big(-E_{nc}-u_c\big)}{\sum_{c'}\exp\big(-E_{nc'}-u_{c'}\big)}. \tag{19}$$

To set the correction terms, we appeal to logistic discriminant training. If we have ground truth class labels for the training images, given by z_n, we could optimize the log-likelihood of correct classification, which is a concave function of the u_c:

$$\begin{aligned}\tilde{\mathcal{L}} &= \sum_n \ln \tilde{p}(z = z_n|\boldsymbol{x}_n) \\ &= -\sum_n E_{nz_n} - \sum_n u_{z_n} - \sum_n \ln \sum_c \exp\big(-E_{nc}-u_c\big) \\ &= \text{const.} - \sum_c n_c u_c - \sum_n \ln \sum_c \exp\big(-E_{nc}-u_c\big), \end{aligned} \tag{20}$$

where $n_c = \sum_n [\![z_n = c]\!]$ denotes the number of examples of class c. The partial derivative w.r.t. u_c is obtained as:

$$\frac{\partial \tilde{\mathcal{L}}}{\partial u_c} = -n_c + \sum_n \tilde{p}(z = c|\boldsymbol{x}_n). \tag{21}$$

Both the log-likelihood and the partial derivative can be computed without access to the labels of the individual samples z_n; it suffices to know the label counts n_c.

Furthermore, from Eq. (21) we see that for the stationary point of $\tilde{\mathcal{L}}$ we have: $\sum_n \tilde{p}(z = c|\boldsymbol{x}_n) = n_c$. Therefore, setting the correction terms to maximize Eq. (20) will ensure that —in expectation— the test classes are predicted as often as they should.

Note that Lampert *et al*. [6] also integrates class specific correction term in their attribute-based classification model that uses independent attribute prediction models. They use $u_c = \ln p(\boldsymbol{y}_c)$, with $p(\boldsymbol{y}_c) = \prod_l^L \sum_{c'}^C \frac{1}{C} [\![y_{lc} = y_{lc'}]\!]$, *i.e*. classes with a high likelihood under a generative model are penalized in the discriminative model.

Setting the class counts: In attribute-based classification, the training data is only labeled at the attribute level, and we do not have access to the counts of the class labels on the training data. In this case we can set the class proportions uniformly, $n_c = N/C$, so that the model will, in expectation, predict all classes equally often. In reality, the test classes are not equally represented, and therefore, setting the u_c based on uniform n_c is, in principle, not optimal. However, experiments where we set the u_c to match the label count on the test set, we see only marginal further improvements in classification accuracy. Calibrating the models using the (true or uniform) label counts n_c can also be done using the test images, instead of the training images, leading to a transductive learning scenario, but again, this has only a minor impact on classification accuracy. We thus conclude that it is important to set the correction terms so as to avoid grossly over or under predicting certain classes, but that it is less important to finely tune these terms using other than uniform counts n_c or using the test images instead of the train images.

Effectiveness of correction terms: To show the effectiveness of the proposed correction terms, we conducted an experiment on two classification settings for the AwA data set. In the first classification setting, the test set consist of 10 classes, while training is performed on the other 40 classes. This is the setting used in [6], [8]. Our second setting uses all 50 classes of the AwA data set for testing, and training images are sampled from all classes to learn the attribute prediction models.

The results of this experiment are shown in Figure 4. On the top row, the confusion matrices are shown as obtained using (18), *i.e*. without incorporating the correction terms. It shows the confusion matrices both for the independent model (*i.e*. without pairwise terms) and for our mixture-of-trees structured model using the two classification settings. The bottom row shows the confusion matrices when the correction terms are used, as given in (19).

The four panels on the top row show the imbalance of the class predictions for any of the methods and settings. *E.g*. in the first panel, we see that using independent attribute prediction models, class 2 is hardly predicted for any test image, while in the second panel, we observe that the mixture-of-tree structured model only frequently predicts class 4 and 9 for the test images. In the two right-most panels, we also observe severe differences in how often the classes are predicted, *c.f*. the vertical stripes in the confusion matrices. This shows that the imbalance in the predictions is not due to using different test classes and training classes. The bottom row shows a more balanced prediction over the classes, which demonstrates how the correction terms can suppress or promote certain classes, allowing us to reduce the severe imbalance in how often the test classes are predicted.

Correction terms using mixture-of-trees: Here, we briefly discuss how we handle the correction terms when using the mixtures of trees. In this case, we mix the class predictions made by the different models as:

$$p(z = c|\boldsymbol{x}) = \sum_t \pi_t p_t(z = c|\boldsymbol{x}), \tag{22}$$

where the π_t are the mixing weights associated with different tree-structured models and $p_t(z = c|\boldsymbol{x})$ indicates the class prediction from one such a tree model.

To balance the class predictions of the mixture model, we learn separate correction terms for each component model $p_t(z = c|\boldsymbol{x})$ as described above. Doing so ensures that the mixture-of-trees model is also calibrated:

$$\sum_n p(z{=}c|\boldsymbol{x}_n) = \sum_t \pi_t \sum_n p_t(z = c|\boldsymbol{x}_n)$$

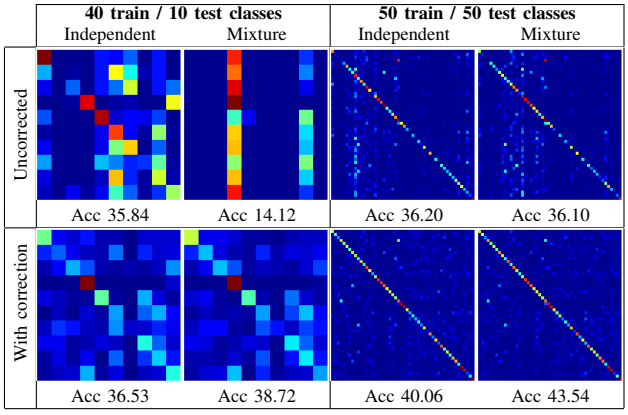

Fig. 4: Influence of the correction terms when using independent and mixtures of trees. See text for details.

$$= \sum_t \pi_t n_c = n_c. \quad (23)$$

Indeed, using in the first equality the definition of the mixture Eq. (22), and swapping the sums; in the second equality the fact that each tree of the mixture has been calibrated; and in the last one that the mixing weights sum to one, we see that the mixture model is calibrated as well.

5.3 Attribute elicitation for image classification

In the case of attribute-based image classification we could use the same label elicitation strategy as for image annotation. However, since the final aim is to improve the class prediction, we use an attribute elicitation criterion that is geared towards minimizing uncertainty on the class label, rather than uncertainty at the attribute level. The main insight is that the information obtained from a revealed attribute value depends on the agreement among the classes on this attribute. If some of the probable classes do not agree with the observed value it will rule out the classes with a contradicting attribute value and concentrate the probability mass on the compatible classes. Therefore, any informative question will at least rule out one of the possible classes, and thus at most $C-1$ attributes need to be set by the user.

In order to see which attribute should be set by the user, we minimize the conditional class entropy $H(z|y_i, \boldsymbol{x})$. Using the identity:

$$H(z, \boldsymbol{y}|\boldsymbol{x}) = H(y_i|\boldsymbol{x}) + H(z|y_i, \boldsymbol{x}) + H(\boldsymbol{y}_{\setminus i}|z, y_i, \boldsymbol{x}), \quad (24)$$

we make the following observations: (i) The left-hand-side of the equation does not depend on the choice of attribute y_i to elicit. (ii) The last term $H(\boldsymbol{y}_{\setminus i}|z, y_i, \boldsymbol{x})$ equals zero, since for each class there is a unique setting of the attribute values. Therefore, selecting the attribute to minimize the remaining entropy on the class label is equivalent to selecting the attribute with the largest marginal entropy $H(y_i|\boldsymbol{x})$.

Note that in the attribute-based classification model, $p(y_i|\boldsymbol{x})$ differs from the image annotation model. Here the probability $p(y_i|\boldsymbol{x})$ is implicitly defined through Eq. (18), which essentially rules-out all attribute configurations, except the ones that correspond to one of the C classes. Therefore, we have

$$p(\boldsymbol{y}|\boldsymbol{x}) = \sum_c p(z=c|\boldsymbol{x}) [\![\boldsymbol{y}=\boldsymbol{y}_c]\!], \quad (25)$$

$$p(y_i|\boldsymbol{x}) = \sum_{\boldsymbol{y}_{\setminus i}} p(\boldsymbol{y}|\boldsymbol{x}) = \sum_c p(z=c|\boldsymbol{x}) [\![y_i=y_{ic}]\!], \quad (26)$$

where y_{ic} denotes the value of attribute i for class c.

We note that the attribute elicitation mechanism for interactive attributed-based image classification is not changed when using different variants of the model (using correction terms, using trees over groups of attributes, or mixtures of such models). In all cases we obtain a class prediction model $p(z=c|\boldsymbol{x})$, which, combined with the class specific label configuration $\boldsymbol{y}_c$, is used to compute marginals over the attribute variables:

$$p(y_i = 1|\boldsymbol{x}) = \sum_c p(z=c|\boldsymbol{x}) y_{ic}. \quad (27)$$

The label marginals are used to select the attribute to be set by the user.

As for image annotation, sequences of user queries are generated progressively by conditioning on the image and all the attribute labels given so far to determine the next attribute to query.

TABLE 1: Basic statistics of the three data sets.

	ImageCLEF [5]	SUN'09 [4]	AwA [6]
# Train images	± 6400	4367	24295
# Test images	± 1600	4317	6180
# Labels	93	107	85
Train img/label	833	219	8812
Train label/img	12.1	5.34	30.8

6 Experimental Evaluation

In this section we describe our experimental evaluation. We first present the used data sets, features and evaluation measures. Followed by, in Section 6.2, the results on automatic and interactive image annotation, in which we experiment with different features for the unary terms, different structured models, and compare to state-of-the art methods. In Section 6.3 we present the results on attribute-based image classification and in Section 6.4 we show results of a multi-word query retrieval experiment.

6.1 Data sets, evaluation and implementation

We performed experiments on three recent public data sets, an overview of some basic statistics is given in Table 1:

ImageCLEF'10 data set: We use *ImageCLEF'10* to refer to the subset of the MIR-Flickr data set [26] that was used as training set in the ImageCLEF 2010 Photo Annotation Challenge [5]. For the challenge, the images were labeled with 93 diverse concepts, see Figure 3.

In this case we tackle a multi-modal labeling task, since for each image the corresponding set of Flickr-tags are

provided, at both train time and test time. Hence, in our experiments we use an early-fusion concatenation of visual and textual features. As visual features we use the improved Fisher vector representation [10] computed over SIFT and color features. This encoding includes a spatial pyramid [27] to take into account the rough geometry of a scene. As textual features, we use a binary vector denoting the presence of the 625 most common Flickr-tags in the data set. The same features have been used in our system that won the challenge [5]; for more details see [28].

In our experiments on ImageCLEF'10, we split the data into five folds, *e.g*. by using fold 1, we learn training classifiers and model parameters on fold 2 to 5, and evaluating the model on fold 1. We report results averaged over the folds, unless otherwise stated. For the sake of clarity we omit standard deviations since they are small compared to the differences between the prediction methods.

SUN'09 data set: The *SUN'09* data set was introduced in [4] to study the influence of contextual information on localization and classification. In contrast to the PASCAL VOC 2007 [29] data set, which has only 20 labels and over 50% of the images having only a single label, the *SUN'09* set contains more labels (107) and around 5 labels per image on average. For this data set we use the same visual features as for ImageCLEF'10.

Animals with Attributes data set: The *Animals with Attributes* (AwA) [6] data set contains images of 50 animal classes, and a definition of each class in terms of 85 attributes. We follow [6], using the provided features[1], the same sum of RBF-χ^2 kernels, and the same 40 train and 10 test classes. We use this data set both to test image annotation of the 85 attributes (Section 6.2) and attribute-based classification (Section 6.3).

Evaluation measures: For the image annotation and classification experiments we measure the performance of the methods using: (i) MAP, a retrieval performance measure, which is the mean average precision (AP) over all keywords, where AP is computed over the ranked images for a given keyword, and (ii) iMAP, which correlates to the number of corrections to obtain a correct image labeling, it is the mean AP over all images, where AP is computed over the ranked labels for an image.

Pre-trained unary potentials: In most of our experiments, we use pre-trained binary SVM classifier scores as unary potentials (Section 3.1.1) in our structured models. To obtain representative SVM classification scores for the train set, we use a method similar to Platt scaling [30], *i.e*. we use a subset of the training set to obtain classification scores for another subset of the training set. This is important because SVM classifiers will (almost) perfectly separate the training set, due to the high capacity dimensionality of our image features. Which makes any additional parameters for the structured model seem unnecessary, if we would train it using SVM scores directly obtained on the train set.

We split each train set into several subsets (in our experiments, we have used 4 or 5 subsets), and for each image n, the classification score $s_i(\boldsymbol{x}_n)$ is obtained by training a binary SVM for concept i on the union of subsets not containing the image n. This assures us that the obtained scores are unbiased, *i.e*. the data is not perfectly separated, and allows us to learn the parameters of the structured models.

For the independent models, we use these unbiased scores to learn a sigmoid function, transforming SVM outputs into probabilistic outputs [30]. For images in the test set we use the SVM scores obtained by classifiers trained on all training images.

The classification scores, train/test splits for ImageCLEF'10 data set, and the multi word queries (see Section 6.4) are available for download[2].

6.2 Image annotation and classification

In this section we evaluate our structured predictions models in the fully automatic and interactive image annotation task on the three data sets. The comparison in this section is between the independent model and trees using the SVM based unary potentials, with the tree structure being obtained based on the mutual information. We also consider using mixtures of trees that have different node sizes.

6.2.1 Fully automatic image annotation

First, we analyze the influence of the structured models in the setting of fully automatic label prediction. Therefore, we evaluate the image annotation performance on MAP and iMAP, the results being shown in Figure 5, first row. For each data set, we compare the independent prediction model (blue) against trees with a group size of $k = 1 \dots 4$ (light-red), and to the mixture of these 4 trees (dark-red).

To the best of our knowledge, our independent classifiers (the blue bars in Figure 5) have state-of-the-art MAP performance on ImageCLEF'10 (conform Section 6.2.4). For the SUN'09 and AwA data sets, we are the first to report MAP over image labels/attributes. In Section 6.2.4 we show that our baseline classifier outperforms previously published results on SUN'09 using another evaluation measure. For the AwA set we compare our baseline classifier in Section 6.3 to the state-of-the art results in [6].

From Figure 5, we can observe that the MAP/iMAP performance of the structured prediction models is about $1 - 1.5\%$ higher than of the independent model. The performance differences between the models with different group sizes k should be seen as a trade-off between model expressiveness and overfitting on the training data. For all data sets the mixture-of-trees performs the best.

The improvement of the structured models over the independent model is relatively modest in the fully automatic setting. This might be due to the fact, that the trees only propagate visual information in this case, which is already very well captured by the independently trained SVM classifiers. In the next section, we will show that in an interactive annotation scenario, the tree based structures can much better exploit and propagate user input than an independent model.

1. http://attributes.kyb.tuebingen.mpg.de/

2. http://lear.inrialpes.fr/~mensink/data

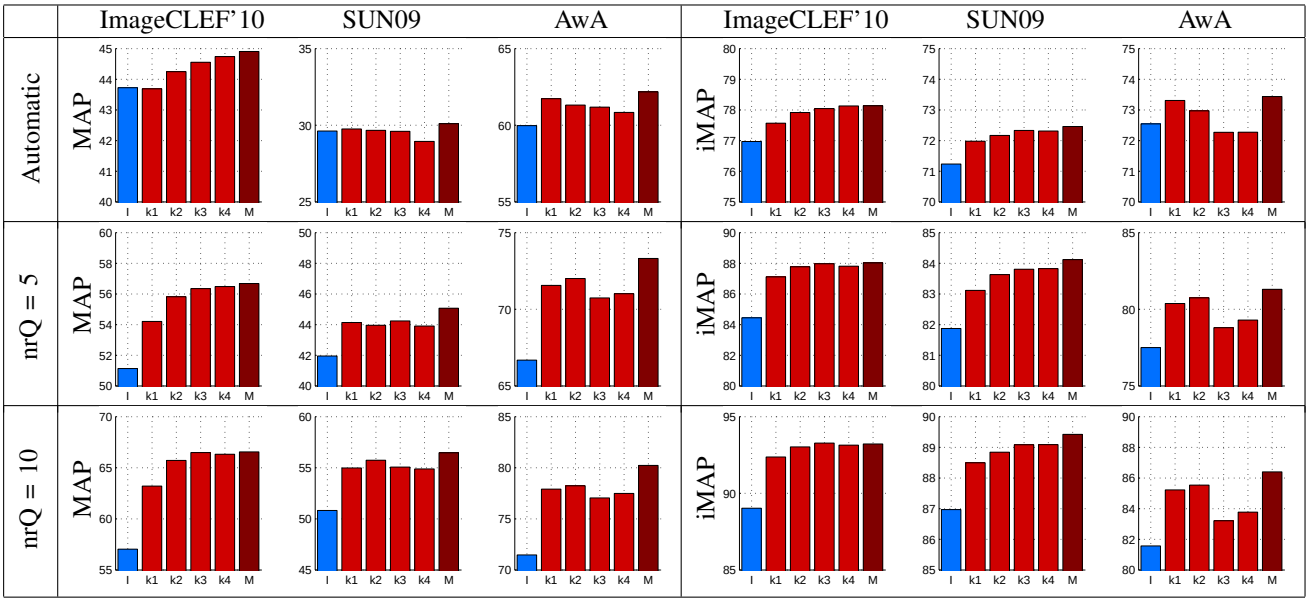

Fig. 5: Overview of the performance on the three data sets for the fully automated prediction setting (first row), and an interactive setting with 5 and 10 questions (second and third row). We compare results of the independent model (blue), the trees with group sizes $1 \leq k \leq 4$ (light-red), and the mixture-of-trees (dark-red). Note the different y-scales.

6.2.2 *Interactive image annotation*

In order to further show the benefit of the proposed structured model, we simulate an interactive image annotation system. The system iteratively selects a label based on the entropy selection criterion, to be set by the "oracle" (the ground truth in our experiments, but this could be a user). The annotation results obtained after 5 respectively 10 "*questions*" (*i.e.* labels asked to the oracle) are shown in the second and third rows of Figure 5.

As expected, in this setting the structured models benefit more from the user input, since they propagate the information provided by the user to update their belief of all labels. The independent model can only update the predictions of the questioned image/label combinations (setting them to either 1 or 0), which explains the increase in their MAP/iMAP performance. In the structured models, on the other hand, some of the label variables become observed due to the user input. These variables now, no longer propagate visual information, but they send messages based on their observed value to the variables connected to them. This new information translates to better predictions on the unknown labels in the tree.

Hence, the overall gain in annotation prediction accuracy for the tree structured models is much higher than for the independent model. Concerning the different tree models, again the mixture-of-trees generally performs the best.

6.2.3 *Further analysis of the proposed models*

In this section we further analyze some of the characteristics of our models including the tree structure, the unary potentials and the label elicitation strategies. All experiments are conducted on the ImageCLEF'10 data set.

Selecting effective dependency structures: The power of label predictions using structured models relies on the chosen dependency structure between the labels. Since both using a fully connected label dependency model, and obtaining the optimal tree structure for discriminative trained model, are intractable, we resort to approximate methods. In the experiments above, we have used tree structures obtained by using the Chow-Liu algorithm (Section 3.1.2) and mixture-of-trees with multiple labels per node (Section 4). In both cases the mutual information was exploited to compute the structure.

In this section, we conduct two further experiments with the aim of evaluating the effect of the selected tree structure on the annotation. In both cases, we use trees with a single label per node ($k = 1$).

In the first experiment we test the following hypothesis: *"the mixture-of-trees outperforms the individual tree models, only because it encodes multiple label dependencies"*. Therefore, we build several tree structures consecutively, by computing the maximum spanning tree (MST) over the mutual information matrix, such that each tree uses only edges, which were not used by any of previous trees. The first tree we obtain in this way equals the optimal tree according to the Chow-Liu algorithm. For each further step (up to 10), we consider not only the new tree built in step t, but also the mixture of the t trees (all having single label per nodes) obtained in the first t steps.

In Figure 6, we show the performance of the individual trees and the performance of the mixtures of these trees. From this figure we see that in the fully automatic setting, a mixture of these trees can slightly improve the performance over the individual trees including the Chow-Liu tree. However, in the interactive setting we observe that the model

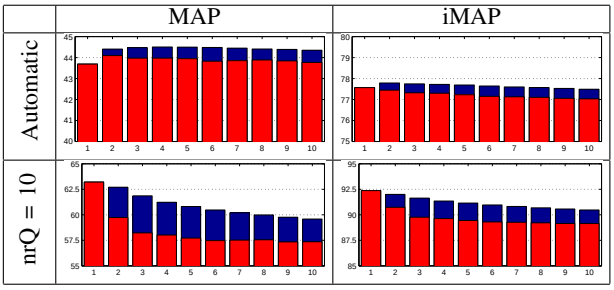

Fig. 6: Performance of different single label trees obtained by iterating the Chow-Liu algorithm (*red*), and mixtures of the trees up to step t (*blue*). We show results for the fully automatic setting (*top row*) and interactive setting with 10 questions (*bottom row*).

TABLE 2: Comparision of Trees using edges based on the gradient and on mutual information, using trees with k=1.

	MAP			iMAP		
	Auto	Q 5	Q 10	Auto	Q 5	Q 10
Gradient	43.6	54.3	63.3	77.5	87.1	92.3
Mut Info	43.7	54.2	63.2	77.6	87.1	92.4

using the Chow-Liu tree outperforms any of the other trees, or mixtures thereof, both on MAP and iMAP. Furthermore, comparing these results with those in Figure 5, it becomes clear that mixing different single node trees has a much lower improvement gain, than considering the mixture-of-trees with different group sizes k.

In the second experiment, we compare two different methods to build the tree, the first method is based on the mutual information (as in previous experiments) and the second method builds a tree using gradient information. To obtain the "gradient tree", we iteratively add edges based on the current gradients of the model (see Section 3.1.2). The results in Table 2 show that the MAP/iMAP performances of the two methods are very similar both for the fully automatic setting and for the interactive setting.

We conclude from these experiments that the structure obtained with the Chow-Liu algorithm – which gives the optimal tree for a generative model – and the mixture-of-trees with different number of labels per node are effective methods to obtain dependency structures for our model.

Joint learning of unary potentials: In the experiments so far, we have used the pre-trained SVM classifier scores in the unary potentials. Joint learning of the unary and pairwise potentials might be more effective since the unary potentials can take into account the effect of the pairwise potentials. To test this, we set $\phi(\boldsymbol{x}_n)$ to be the concatenation of visual and textual features, yielding a high-dimensional vector, and use the kernel representation of Eq. (9) to optimize the regularized log-likelihood defined in Eq. (8). We vary the regularization parameter λ in the range $[10^{-6}, 10^{-2}]$, and report the best results. In this experiment, we have used only fold 1 of the ImageCLEF'10 data set, instead of averaging over all folds. The reason is that the computational cost for this experiment is much higher, and since we have observed similar behavior on different folds, we expect that the results will generalize to the other folds as well.

TABLE 3: MAP performance of trees using unary potentials that are either pre-trained or jointly learned.

	Fully Automatic				
	$k=1$	$k=2$	$k=3$	$k=4$	Mixt
Pre-trained	43.3	44.0	44.6	44.8	44.9
Jointly-learned	41.6	41.6	42.0	40.4	41.8
	After 10 Questions				
Pre-trained	62.9	65.7	66.5	66.4	66.7
Jointly-learned	60.2	61.6	62.2	61.7	63.2

From Table 3, we see that the joint learning of the unary potentials never matches the performance obtained with the pre-trained SVM classifier based unary potentials. This observation is consistent along all tested settings, the fully automatic and the interactive evaluation setting, as well as different label group sizes per node.

These results contrast with those of [31], where pre-training is shown to be competitive yet outperformed by joint learning. Note that our work differs from theirs in at least two important ways: (i) Our unary potentials for image labeling —using global image features— are probably much stronger than their unary potentials for pixel-wise labeling —using only local features. (ii) Our pre-trained SVM scores resemble test-time prediction scores, since they are obtained in a cross-validation manner (see Section 6.1), while in [31] such a procedure is not followed. We interpret these findings as an indication that in the presence of strong unary potentials, it is important to use unary scores that are representative of the test data scores.

Label elicitation strategy: To show the benefit of the proposed label elicitation methods, we compare the performances of the independent model and the mixture-of-trees model using two different label elicitation strategies. The first strategy is the entropy based selection criteria, described in Section 3.2. The second is to randomly select labels, for which we report the mean performance over 10 evaluations using different randomly selected questions.

The results in Figure 7 show the performance of the independent predictors (*blue*) and our mixture model (*red*), from no user input to complete user-input. We can see that both models benefit more from the label entropy based elicitation mechanism compared to the randomly selected labels. Furthermore, we observe that our structured method achieves perfect labeling after significantly fewer questions than the independent predictors.

6.2.4 Comparison to related work

In this section we compare our methods to state-of-the-art results obtained on the used data sets.

ImageCLEF 2010 Photo Annotation Challenge: In the previous experiments we have used only the available train set of the ImageCLEF 2010 challenge, and split it into several train and test folds. However, to compare our methods to the ImageCLEF 2010 challenge results, we

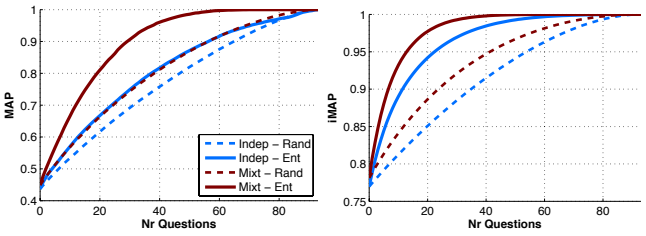

Fig. 7: MAP and iMAP scores as function of the number labels set by the user on ImageCLEF'10.

TABLE 4: Comparison on the ImageClef 2010 Visual Concept Detection and Annotation Task.

Team and method		Modality	MAP
XRCE - SVM EF	[28]	V&T	**45.5**
LEAR - TagProp D3	[28]	V&T	43.7
ISIS - MKL	[32]	V	40.7
XRCE - SVM EF	[28]	V	38.9
HHI	[33]	V	34.9
IJS	[34]	V	33.4
MEIJI	[35]	V&T	32.6
Mixture-of-trees		V&T	**46.7**
		V	40.0

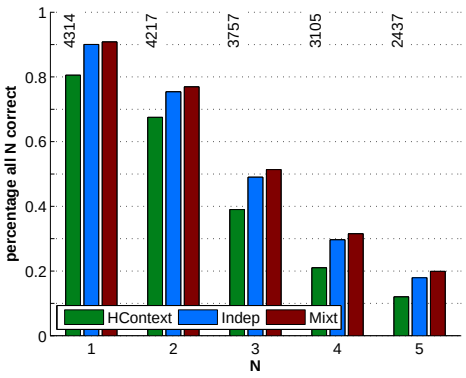

Fig. 8: Comparison of our independent and mixture-of-trees methods to the results of [4] on the SUN'09 data set.

TABLE 5: Zero-shot attribute-based classification accuracy of the independent and mixture-of-trees models. Initial results, and after user input for 1 up to 8 selected attributes.

	Init	1	2	3	4	5	6	7	8
Indep	38.1	55.5	71.0	79.9	86.1	91.1	95.3	97.7	99.6
Mixt	40.4	59.2	75.7	88.8	96.0	99.1	99.7	100.0	100.0

evaluated using the official training and test set[3]. Table 4 shows both top performing results of the participants in the ImageCLEF 2010 challenge (see [5] for an overview of the participants, different methods and results) and the performances of our methods. In this table we report the *interpolated* MAP (conform the challenge), while in the rest of the paper we reported non-interpolated MAP.

In Table 4, we further indicated whether only the visual (V) modality (image) or both the visual and textual (V&T) modalities (image, Flickr-tags and exif meta-information) were used. In the top part of the table, we show the top methods ranked by their interpolated MAP performance in the challenge. Our current baseline system (the independent model) corresponds to the "XRCE - SVM EF", which was the winner of the challenge when used both modalities (V&T). In the bottom part of the table, we show the performance of our method when using the mixture-of-trees model with the SVM based unary potentials, using both modalities (V&T) and when using the visual features (V) only. Again, we can observe that the structured models outperform the independent models by about 1% MAP (both in case of visual only and multi-modal system).

Comparison to the hierarchical context model: In this section, we compare our method to the state-of-the art results on the SUN'09 obtained with the hierarchical context method (HContext) proposed in [4]. Therefore, we used the evaluation method of [4], *i.e.* the percentage of images in which the top N predicted labels are all correct, taken over the images with at least N labels. Results for our independent and structured models along with the results published in [4] are shown in Figure 8.

We notice that the independent method clearly outperforms the HContext method, even in spite of the fact that the HContext model uses the object bounding boxes during training, while our independent method uses only global image labels. The performance difference can be partially explained by the stronger image representation (Fisher vectors) we use compared to their GIST [22] features. This comparison shows the strength of our baseline.

6.3 Attribute-based prediction of unseen classes

The AwA data set was introduced for transfer learning by means of sharing attributes used to represent different classes. We use the zero-shot prediction paradigm, where the test classes and train classes are disjoint. Hence, in this section we evaluate the performance of our structured models in predicting class labels (see Section 5) of images from unseen classes based on the class specific configuration of the 85 attributes.

To compare our approach to the state-of-the art, we use the same settings and the same evaluation measure (mean of the diagonal of the normalized confusion matrix) as in [6]. Table 5 shows the performance of the independent model[4] and our mixture-of-trees model.

Note that the tree structured model learns attribute dependencies for the train classes which are different from the test classes, *i.e.* during testing we have combinations of attributes which have never been seen before. Still, our model is able to take advantage of the learned attribute dependencies to significantly improve over the results of the independent model.

3. The ground-truth annotation of the test set will be released publicly by the organizers soon on http://www.imageclef.org.

4. Our baseline result of 38.1 is somewhat below the result of 40.5 reported in [6]. After conversation with the authors, we conclude that this is probably due to the use of different class correction terms.

TABLE 6: Performance in MAP for multi-word queries of different length.

Query length	1	2	3	4	5
Number of queries	93	1,535	9,343	28,929	53,807
Independent model	43.7	26.7	21.2	19.1	18.3
Mixture-of-trees	44.9	27.8	22.2	20.0	19.2

6.4 Multi-word query retrieval

When the image annotation performance is evaluated using MAP for a specific label, it resembles the evaluation of an image retrieval system where the query consists of a single label. In a general purpose image retrieval system however, users tend to use multi-word queries to find images or documents. Therefore, in this experiment we evaluate our proposed models for multi-word queries using the ImageCLEF'10 data set. All results are averaged over the 5 test folds of the ImageCLEF'10 data set.

For this experiment we have created a query set containing all multi-word queries up to length 5, with at least 5 positive images present in all of the test folds. A positive image means that all words from the query are relevant for this image according to the ground truth. This yields a query set of about 95.000 queries.

For each query we rank the images according to the likelihood $p(\{\boldsymbol{y}_q\}|\boldsymbol{x})$, *i.e.* the marginal that the query terms are relevant. For the tree model, we have,

$$p(\{\boldsymbol{y}_q\}|\boldsymbol{x}) = \frac{1}{Z} \sum_{\substack{\boldsymbol{y} \\ \text{s.t.} \forall i \in q: y_i = 1}} \exp\big(-E(\boldsymbol{y}, \boldsymbol{x})\big) = \frac{Z_q}{Z},$$

where Z is the partition function. The term Z_q can easily be computed using standard BP: it equals to the partition function while clamping the labels $\{\boldsymbol{y}_q\}$ to 1. For the mixture-of-trees model we use $p(\{\boldsymbol{y}_q\}|\boldsymbol{x}) = \sum_t \pi_t p(\{\boldsymbol{y}_q\}|\boldsymbol{x}, t)$. For the independent model we use:

$$p(\{\boldsymbol{y}_q\}|\boldsymbol{x}) = \prod_{i \in q} p(y_i|\boldsymbol{x}).$$

In Table 6 we compare the mixture-of-trees model to the independent model. We observe an improvement of about 1% in MAP when using the mixture-of-trees over the independent model, regardless of the query length.

In contrast to what Table 6 suggests, surprisingly the difficulty of a query does not depend so much on its length, but is mainly determined by the number of positive images available for that query in the data set. To illustrate this, in Figure 9 we show the performance in MAP as function of the query length and the frequency of positive documents in the test set. This figure shows that the MAP performance is much more influenced by the number of positive images available for the query, than the number of words in the query. Indeed, for short queries there tend to be many more positive images, so the overall performance is higher than for longer queries (as we can see in Table 6). Furthermore, if we fix the query length, we observe an increase in performance when the number of positive images in the test set increases.

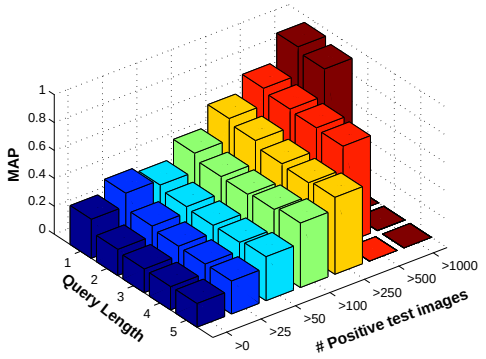

Fig. 9: Performance of multi-word queries grouped in the query length and the number of occurrences.

7 Conclusion

We introduced a class of structured image labeling models to capture label dependencies. To capture more dependencies we extended the basic tree model by using multiple labels per node, and using mixtures of such models. We explored (i) different strategies to learn the unary potentials (pre-trained SVM classifiers and joint learning with the pairwise potentials), (ii) various graphical structures (trees, trees over label groups and mixture-of-trees), and (iii) methods to obtain these structures (using mutual information and gradient information).

We find that best performance is obtained using a mixture-of-trees with different label group sizes, where the unary potentials are given by pre-trained SVM classifiers. During training, the SVM scores are obtained in a cross-validation manner, to ensure that the quality of the SVM scores is representative of that of test images.

While capturing complex label dependencies, the low tree width of our models still allows for tractable inference for label prediction, model learning and label elicitation. Our models were tested on different image labeling application scenarios, including automatic and semi-automatic image annotation, attribute-based image classification, and multi-word query retrieval.

Although the proposed models offer only moderate improvements over independent baseline models in a fully automatic setting, their main strength appears in an interactive setting. In this case, the system asks a user to set the value of a small number of labels at test time, which offers a trade-off between label accuracy and labeling effort.

The proposed structured models are able to transfer user input to other image labels yielding more accurate predictions. This holds even more when the labels are selected following the entropy based criterion to reduce the remaining uncertainty of the other labels. We observed a similar trend in the case of attribute-based image classification, *i.e.* our structured models obtain higher accuracy than the independent model using a small amount of user input on the attribute level.

REFERENCES

[1] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *IJCV*, vol. 73, no. 2, pp. 213–238, 2007.

[2] D. Grangier and S. Bengio, "A discriminative kernel-based model to rank images from text queries," *PAMI*, vol. 30, no. 8, pp. 1371–1384, 2008.

[3] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, "Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation," in *ICCV*, 2009.

[4] M. Choi, J. Lim, A. Torralba, and A. Willsky, "Exploiting hierarchical context on a large database of object categories," in *CVPR*, 2010.

[5] S. Nowak and M. Huiskes, "New strategies for image annotation: Overview of the photo annotation task at ImageCLEF 2010," in *Working Notes of CLEF*, 2010.

[6] C. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *CVPR*, 2009.

[7] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie, "Visual recognition with humans in the loop," in *ECCV*, 2010.

[8] T. Mensink, J. Verbeek, and G. Csurka, "Learning structured prediction models for interactive image labeling," in *CVPR*, 2011.

[9] V. Vapnik, *The Nature of Statistical Learning Theory*. Spinger, 1995.

[10] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *ECCV*, 2010.

[11] A. Makadia, V. Pavlovic, and S. Kumar, "A new baseline for image annotation," in *ECCV*, 2008.

[12] J. Weston, S. Bengio, and N. Usunier, "Large scale image annotation: Learning to rank with joint word-image embeddings," in *ECML*, 2010.

[13] J. Deng, A. Berg, K. Li, and F.-F. Li, "What does classifying more than 10,000 image categories tell us?" in *ECCV*, 2010.

[14] C. Desai, D. Ramanan, and C. Fowlkes, "Discriminative models for multi-class object layout," in *ICCV*, 2009.

[15] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in context," in *ICCV*, 2007.

[16] B. Settles, "Active learning literature survey," University of Wisconsin-Madison, Tech. Rep. 1648, 2009.

[17] S. Vijayanarasimhan and K. Grauman, "Multi-level active prediction of useful image annotations for recognition," in *NIPS*, 2009.

[18] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *JMLR*, vol. 6, pp. 1453–1484, 2005.

[19] C. Bishop, *Pattern recognition and machine learning*. Spinger, 2006.

[20] S. Nowozin and C. Lampert, "Structured learning and prediction in computer vision," *Foundations and Trends in Computer Graphics and Vision*, vol. 6, pp. 185–365, 2011.

[21] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.

[22] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *IJCV*, vol. 42, pp. 145–175, 2001.

[23] J. Bradley and C. Guestrin, "Learning tree conditional random fields," in *ICML*, 2010.

[24] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[25] P. Pletscher, C. Ong, and J. Buhmann, "Spanning tree approximations for conditional random fields," in *AISTATS*, 2009.

[26] M. Huiskes and M. Lew, "The MIR Flickr retrieval evaluation," in *ACM MIR*, 2008.

[27] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006.

[28] T. Mensink, G. Csurka, F. Perronnin, J. Sánchez, and J. Verbeek, "LEAR and XRCE's participation to Visual Concept Detection Task - ImageCLEF 2010," in *Workshop ImageCLEF*, 2010.

[29] 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.

[30] J. Platt, "Probabilities for SV machines," in *Advances in Large Margin Classifiers*, 2000.

[31] S. Nowozin, P. Gehler, and C. Lampert, "On Parameter Learning in CRF-based Approaches to Object Class Image Segmentation," in *ECCV*, 2010.

[32] K. van de Sande and T. Gevers, "The University of Amsterdam's Concept Detection System at ImageCLEF 2010," in *Workshop ImageCLEF*, 2010.

[33] E. Mbanya, C. Hentschel, S. Gerke, M. Liu, A. Nürnberger, and P. Ndjiki-Nya, "Augmenting Bag-of-Words - Category Specific Features and Concept Reasoning," in *Workshop ImageCLEF*, 2010.

[34] I. Dimitrovski, D. Kocev, S. Loskovska, and S. Džeroski, "Detection of Visual Concepts and Annotation of Images Using Predictive Clustering Trees," in *Workshop ImageCLEF*, 2010.

[35] N. Motohashi, R. Izawa, and T. Takagi, "Meiji University at ImageCLEF2010 Visual Concept Detection and Annotation Task," in *Workshop ImageCLEF*, 2010.

Thomas Mensink received the cum laude MSc degree in artificial intelligence from the University of Amsterdam, The Netherlands in 2007. He is currently working toward the PhD degree, jointly at the TVPA group of Xerox Research Centre Europe and at the LEAR team of INRIA Rhone-Alpes, France. His research interests include computer vision and machine learning.

Jakob Verbeek received a PhD degree in computer science in 2004 from the University of Amsterdam, The Netherlands. After being a postdoctoral researcher at the University of Amsterdam and at INRIA Rhône-Alpes, he has been a full-time researcher at INRIA, Grenoble, France, since 2007. His research interests include machine learning and computer vision, with special interest in applications of statistical models in computer vision.

Gabriela Csurka is a senior scientist at XRCE, her research interest include monomodal and multi-modal image retrieval, categorization, and segmentation. Graduated from the University of Timisoara, Rumania (1991), she obtained her PhD degree in Computer Science from the University of Nice, France (1996). She prepared her PhD thesis, entitled "Projective modeling of three-dimensional objects in computer vision" at INRIA Sophia Antipolis.