Semantic lattices for multiple annotation of images
Anne-Marie Toussch, Stéphane Herbin, Jean-Yves Audibert

To cite this version:
Anne-Marie Toussch, Stéphane Herbin, Jean-Yves Audibert. Semantic lattices for multiple annotation of images. 1st ACM international conference on Multimedia information retrieval, Oct 2008, Vancouver, Canada. pp.342-349, 10.1145/1460096.1460152. hal-00835102

HAL Id: hal-00835102
https://hal-enpc.archives-ouvertes.fr/hal-00835102
Submitted on 18 Jun 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Semantic Lattices for Multiple Annotation of Images

Anne-Marie Tousch
ONERA
&
Université Paris-Est
Ecole des Ponts ParisTech
CERTIS
Marne-La-Vallée, France
tousch@certis.enpc.fr

Stéphane Herbin
ONERA
Châtillon, France
stephane.herbin@onera.fr

Jean-Yves Audibert
CERTIS
&
Willow - ENS / INRIA
Paris, France
audibert@certis.enpc.fr

ABSTRACT
We address the problem of describing precisely an object present in an image. The starting point is a semantic lattice defining all possible coherent object descriptions through inheritance and exclusion relations. This domain knowledge is used in a learning process which outputs a set of coherent explanations of the image valued by their confidence level. Our first contribution is to design this method for multiple complexity level image description. Our secondary focus is to develop rigorous evaluation standards for this computer vision task which, to our knowledge, has not been addressed in the literature despite its possible use in symbolic annotation of multimedia database. A critical evaluation of our approach under the proposed standards is presented on a new appropriate car database that we have collected.

1. INTRODUCTION
1.1 Problem statement
This work is about generating multiple image annotations corresponding to various levels of semantic precision. The origin of the problem we address lies in the nature of semantic interpretation of data. Indeed, describing the content of an image is an ill-posed problem: the type of relevant description depends on the context of use which is not univocally defined by the image alone. For instance, when observing an image containing a car — the type of data that will illustrate our approach (see Fig. 1) — one may be interested in finding its brand or in characterizing its shape or its size. One may also be interested in identifying the car model name, or even its version.

In image retrieval problems, one solution to solve the inherent ambiguousness of data description is to make use of image content description based techniques, i.e. to rely only on universal image features such as color, shape or texture models. The claim of this kind of approach is that it is able to get around the semantic gap issue, using for instance relevance feedback querying strategies [35].

Another trend to carry out semantic analysis is to exploit knowledge representations such as ontologies on symbolic metadata. The use of semantic tools is expected to master the polysemy or imprecision of both symbolic annotations and queries. In this family of approaches, two subproblems need to be solved: the construction of relevant annotations, and the design of a similarity measure between the annotations and a compatible form of the research query [13].

The target context of this study is domain specific applications. This introduces several peculiarities:
- meaningful differences between data rely on very specific details which are hard to guess without expert knowledge;
- users usually master the specific concepts and vocabulary of the domain;
- the size of the database is large, but typically less than 10^5 items;
- annotated or reference data are scarce.

Content based image retrieval may not be the adequate framework for dealing with this kind of constraints, since efficient methods often rely on a rather large learning database, or address too coarse classification for domain specific applications. We propose to base our approach on the following features:
- image indexing is automatic and is based on few reference annotated data;
- annotations should be coherent with a domain knowledge representation;
- research queries are symbolic or textual;
- semantic ambiguousness is solved offline by generating multiple annotations,
this last point being the main contribution of our work.

Figure 1: images of cars from 7 different classes.
A multiple annotation has to be understood as a distribution of consistent annotations each valued by a confidence coefficient (Tab. 1). Each element in this list is assumed to address a given level of semantic precision roughly characterized by the number of labels.

One policy for computing localized image features is first choosing a location using interest, regularly spaced or random points, then characterizing the image locally. Marszalek and Schmid [25] use patches located by salient region and point detectors. Lazebnik et al. [17] use dense sampling on a regular grid. Nowak et al. [25] show that random sampling achieve comparable performance for bag-of-features type image classification. Chen et al. [12] use the regions texture and color properties of the segmented image coupled with a multiple-instance learning algorithm.

Another local image feature computing policy, closer to the one we are following in this work, is to build detectors adapted to very specific though simple characteristics. This has been used for a long time in face recognition, for instance, where eye, nose or mouth detectors form a basis used to detect or normalize face appearances [1].

1.2.2 Interpretation of features
Images may be described along different directions: Hollink et al. [15] propose to distinguish between non visual (date, photographer’s name, digital format . . . ), perceptual (color, shape, texture, spatial relations . . . ) or conceptual (event, name of person, place . . . ) levels. The fundamental issue of data annotation, often referred to as the semantic gap [34, 18], addresses the problem of relating the perceptual and the conceptual levels. The perceptual level, somehow likened to what is called the image content, may also be described using sophisticated symbolic representations [21].

The majority of studies have addressed image description either as detection or as categorization problem, i.e. the descriptions belong to conceptual spaces with simple topology. More recently, image retrieval issues have requested more flexible or semantically colored types of description inspired by document classification applications [14, 4, 8] and have addressed multi-label description [3, 19, 7, 9] or ontology based annotations reduced to taxonomic or hierarchical descriptions [29, 5, 30, 23, 27]. [20] introduce uncertainty on the labelling and compute a probability for each concept given an image. However, they treat each concept regardless of their semantic relations. To the best of our knowledge, semantic hierarchies or taxonomies have been used mostly to improve performance in recognition tasks [24], or have

1See the classification method description of the Pascal VOC Challenge 2007

<table>
<thead>
<tr>
<th>Description</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hatchback</td>
<td>0.78</td>
</tr>
<tr>
<td>Supermini</td>
<td>0.66</td>
</tr>
<tr>
<td>Hatchback, Supermini</td>
<td>0.6</td>
</tr>
<tr>
<td>Peugeot</td>
<td>0.55</td>
</tr>
<tr>
<td>Hatchback, Supermini, Peugeot</td>
<td>0.5</td>
</tr>
<tr>
<td>Hatchback, Supermini, Peugeot, 206</td>
<td>0.45</td>
</tr>
<tr>
<td>Hatchback, Supermini, Peugeot, 206, 3 doors</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Table 1: A possible sequence of outputs in our working scheme: the more precise the description, the less reliable.
been generated as a by-product of a classification scheme [32]; they have not been used to generate multiple description level characterization.

Our approach intensively uses machine learning techniques. In this field, several recent studies have addressed learning in complex or structured spaces extending kernel-based approaches or graphical models to structured spaces such as multilabels, strings or hierarchies [2, 11, 10, 31].

The conceptual level studied in this paper has a lattice topology, and is not organized as a true hierarchy: a description, i.e. a list of labels, may have more than one parent, i.e. may have more than one simpler consistent description. This peculiarity motivated the specific developments presented in following.

1.3 Overview of the approach

The objective of this work is to build a process for the description of images or objects with multiple levels of semantic precision. The output is a series of lists of labels, each valued with a reliability or confidence coefficient. The consistency of each list of labels is guaranteed by a semantic lattice aiming at representing domain knowledge.

Our global processing chain is divided into four main tasks (Fig. 3):

1. Extracting image information by computing a signature.
2. Calculating a probability for each list of labels based on the image signature.
3. Ensuring global coherence of the probabilities using a semantic lattice.
4. Issuing the series of consistent lists of labels ordered by their probability.

The rest of the article is organized as follows: the computation of the signature is detailed in section 2; the image annotation step using signatures is presented in section 3; section 4 is about designing an adequate criterion for performance evaluation; experiments and results are presented in section 5 along with the methods we compared with.

2. IMAGE FEATURES

Image representation based on the detection of local features have proven successful. Zhang et al. [33] have provided an in-depth study of the state-of-the-art methods using local features and kernels for object categorization. We choose to represent an image as a vector of binary values where each binary value corresponds to the presence or not of a particular local patch detector in the image, thus yielding a low-level semantic vocabulary. As in the bag of words approach, we do not represent spatial relationship between features. However, contrarily to the latter technique, the vocabulary is predefined through labeling in the training images of small patches potentially discriminant for identification of each class (logo, lights, ...; see Fig. 4).

A detector is designed for each word of the vocabulary. It is based on

- representing the patches by a SIFT-like descriptor [22] which is known to cope well with illumination and contrast variations: we use $4 \times 4$ local histograms corresponding to a $4 \times 4$ square grid, each histogram containing 8 bins corresponding to possible orientations of the gradient in one of the $4 \times 4$ squares.
- a one-class-SVM [28] with an histogram intersection
kernel [6] which has already been successfully used with the SIFT descriptors [17]. The advantage of using one-class-SVMs is essentially to avoid the definition of the "negative" class through instances: an object is better defined by what it is than what it is not.

Due to the limited size of the training set and in order to improve the one-class-SVM classifier, we artificially increase the training set size by applying small affine transformations of the training images. In order to control the false alarms, we look for the patch corresponding to a particular word in the training set.

To find the presence in an image of a word of this vocabulary, we use a sliding window technique, that is we look for a few notations to define it. The one-class-SVMs is essentially to avoid the definition of the "negative" class through instances: an object is better defined by what it is than what it is not.

Due to the limited size of the training set and in order to improve the one-class-SVM classifier, we artificially increase the training set size by applying small affine transformations of the training images. In order to control the false alarms, we look for the patch corresponding to a particular word only at scales and positions similar to the one observed in the training set.

To find the presence in an image of a word of this vocabulary, we use a sliding window technique, that is we look for a few notations to define it. The one-class-SVMs is essentially to avoid the definition of the "negative" class through instances: an object is better defined by what it is than what it is not.

Due to the limited size of the training set and in order to improve the one-class-SVM classifier, we artificially increase the training set size by applying small affine transformations of the training images. In order to control the false alarms, we look for the patch corresponding to a particular word only at scales and positions similar to the one observed in the training set.

To find the presence in an image of a word of this vocabulary, we use a sliding window technique, that is we look for a few notations to define it. The one-class-SVMs is essentially to avoid the definition of the "negative" class through instances: an object is better defined by what it is than what it is not.

Due to the limited size of the training set and in order to improve the one-class-SVM classifier, we artificially increase the training set size by applying small affine transformations of the training images. In order to control the false alarms, we look for the patch corresponding to a particular word only at scales and positions similar to the one observed in the training set.

To find the presence in an image of a word of this vocabulary, we use a sliding window technique, that is we look for a few notations to define it. The one-class-SVMs is essentially to avoid the definition of the "negative" class through instances: an object is better defined by what it is than what it is not.

Due to the limited size of the training set and in order to improve the one-class-SVM classifier, we artificially increase the training set size by applying small affine transformations of the training images. In order to control the false alarms, we look for the patch corresponding to a particular word only at scales and positions similar to the one observed in the training set.

To find the presence in an image of a word of this vocabulary, we use a sliding window technique, that is we look for a few notations to define it. The one-class-SVMs is essentially to avoid the definition of the "negative" class through instances: an object is better defined by what it is than what it is not.

Due to the limited size of the training set and in order to improve the one-class-SVM classifier, we artificially increase the training set size by applying small affine transformations of the training images. In order to control the false alarms, we look for the patch corresponding to a particular word only at scales and positions similar to the one observed in the training set.

To find the presence in an image of a word of this vocabulary, we use a sliding window technique, that is we look for a few notations to define it. The one-class-SVMs is essentially to avoid the definition of the "negative" class through instances: an object is better defined by what it is than what it is not.

Due to the limited size of the training set and in order to improve the one-class-SVM classifier, we artificially increase the training set size by applying small affine transformations of the training images. In order to control the false alarms, we look for the patch corresponding to a particular word only at scales and positions similar to the one observed in the training set.

To find the presence in an image of a word of this vocabulary, we use a sliding window technique, that is we look for a few notations to define it. The one-class-SVMs is essentially to avoid the definition of the "negative" class through instances: an object is better defined by what it is than what it is not.

Due to the limited size of the training set and in order to improve the one-class-SVM classifier, we artificially increase the training set size by applying small affine transformations of the training images. In order to control the false alarms, we look for the patch corresponding to a particular word only at scales and positions similar to the one observed in the training set.

To find the presence in an image of a word of this vocabulary, we use a sliding window technique, that is we look for a few notations to define it. The one-class-SVMs is essentially to avoid the definition of the "negative" class through instances: an object is better defined by what it is than what it is not.

Due to the limited size of the training set and in order to improve the one-class-SVM classifier, we artificially increase the training set size by applying small affine transformations of the training images. In order to control the false alarms, we look for the patch corresponding to a particular word only at scales and positions similar to the one observed in the training set.
exclusive classes. We assume that each data ultimate ground truth is such a multilabel. In the application tested, we have 20 leaves.

We state that the description complexity is equal to the number of labels used to describe the data, and define:
- node complexity: $C(i) = |\text{arc}(i)| + 1$
- multilabel complexity: $C(y) = |y| = \sum_{j=1}^{N_y} y_j$

Node and multilabel complexities are equivalent since we have $C(i) = C(y(i))$.

### 3.3 Multilabel probability computation

The computation of a probability for each consistent multilabel is done in two steps: computation of a probability for each node of the graph $G$; propagation of those probabilities to all the other consistent multilabels, ensuring global coherence.

The computation of a probability for node $i$, or equivalently for multilabel $y(i)$, is achieved using a binary SVM with gaussian kernel applied on the data signature. For each node, the database is divided into positive and negative samples in a one-versus-rest approach. A sample will be considered positive if its ground truth contains all the labels of the multilabel associated with the node.

This scheme leads to highly unbalanced problems when going down the graph. We handled this by using different SVM $C$ parameters for positive and negative data and using an adapted error. After training, we followed Platt’s method [26] for converting SVM outputs into probabilities by fitting a sigmoid on the classifier output.

Let $\{p_1, \ldots, p_{N_y}\}$ be the set of the SVM probabilistic outputs for all nodes in the graph given an input signature. We need to estimate probabilities for the multilabels that are not associated with a node due to multiple inheritance.

The idea is to build the global distribution of multilabel probabilities on the probabilities assigned to the leaves, i.e. the most complex descriptions. Indeed, the leaves make a partition of the data — they are exhaustive and mutually exclusive — so that each multilabel should verify:

$$ p(y) = \sum_{j \in G} y_j p_j, \quad (1) $$

where $p(y)$ is the probability assigned to multilabel $y$ and $G$ is the set of leaves. This equality must hold also for multilabels associated with each node $y(i)$:

$$ p_i = p(y(i)) = \sum_{j \in G} y(i, j) p_j, \quad (2) $$

where $y(i, j)$ is the $j$-th coordinate of multilabel $y(i)$.

The probabilities obtained using the direct computation of the SVMs may not satisfy the constraint (2) for every node. We seek a regularized approximation $\hat{p}_i$ of the probabilities assigned to each leaf, optimizing the criterion:

$$ \min_{\hat{p}} \sum_{i \in G} w_i \cdot (p_i - \sum_{j \in G} y(i, j) \hat{p}_j)^2 + \sum_{j \in G} w_j \cdot (p_j - \hat{p}_j)^2, \quad (3) $$

s.t. $\sum_{j \in G} \hat{p}_j = 1$, and $\forall j, \hat{p}_j > 0, \quad (4)$

where $\hat{p}$ is the estimated probabilities on the leaves, and $w_i$ is a weight on node $i$ such that probabilities on low complexity nodes are favored. We choose to take $w_i = \frac{1}{C(i)^2}$, and after experiments, we set $\gamma$ to 0.01. This is a simple convex quadratic programming problem of dimension the number of leaves of the semantic graph. The two members in eq. (3) correspond respectively to (a) trying to reach the constraint (2) for nodes and (b) keeping probabilities for leaves the nearest possible to the probabilistic outputs.

The probability of any multilabel in $Y_y$ is then computed using equation 1 where $p_i$ is replaced by $\hat{p}_i$. Using this global computation scheme — SVM on every node + regularization — we have assigned a probability to each consistent multilabel. The next section describes how to exploit this probability distribution, and how to evaluate its descriptive capacity.

### 4. EVALUATION

An algorithm solving our multiple complexity level image description task should output confidence levels for any possible description. To be consistent, these probabilities need to be decreasing along any chain of multilabels $y_1, \ldots, y_k$ such that $y_1 \subseteq \cdots \subseteq y_k$.

For a test sample with true multilabel $t$, any multilabel $y \subseteq t$ is a correct answer. The semantic precision of the final answer of the algorithm is controlled by an input confidence parameter, denoted hereafter $p$, that the user can tune. The idea is to make the algorithm output the most complex (or precise) explanation of the image which has a confidence level greater than $p$. The choice of the multilabel of maximum complexity with probability larger than $p$ is done through the following steps:

1. Threshold the set of multilabels to keep only multilabels having confidence coefficients larger than $p$:

$$ Y_y^p = \{ y \in Y_y | p(y) \geq p \}, \quad (5) $$

2. Build the set of maximal (in complexity) multilabels among the high confidence multilabel set $Y_y^p$:

$$ \partial Y_y^p = \{ y \in Y_y^p | \forall y' \in Y_y, \text{ s.t. } y \subseteq y', y' \notin Y_y^p \}, \quad (6) $$

3. Choose the multilabel $\hat{y}$ maximizing the probability $p(y)$ in $\partial Y_y^p$.

To evaluate the classification efficiency of algorithms solving our multiple complexity level image description task, we plot an error/complexity curve. This curve is parameterized by the confidence factor $p \in [0, 1]$, a point $(c(p), \varepsilon(p))$ being the mean complexity and mean error of answers for $p$ on the test set:

$$ c(p) = \frac{1}{\sum_{i=1}^{N} C(y_i)}, \quad (7) $$

$$ \varepsilon(p) = \frac{1}{N} \sum_{i=1}^{N} \ell(y_i, t_i), \quad (8) $$

where $t_i$ is the ground truth for sample $i$ and $\ell$ is the 0/1-loss function:

$$ \ell(y_1, y_2) = \begin{cases} 0 & \text{if } y_1 \subseteq y_2, \\ 1 & \text{otherwise}. \end{cases} \quad (9) $$

This error/complexity curve is the fundamental tool used to evaluate and compare multiple description algorithms.

### 5. EXPERIMENTS

#### 5.1 Dataset

Our dataset is composed of 644 images of 20 classes of cars with varied inter-class visual and semantical differences. The dataset is divided into two separate sets namely “Lc.”
where all images were richly annotated (326 images) and “Lₐ” where only the class was given (318 images). The distribution of examples per class is highly variable depending on their real global statistical distribution (the photos were taken in the streets) varying from 3 to 29 photos for a class in Lₐ or Lₜ. Illumination changes, reflections and car colors create intra-class variation. The viewpoint angle was limited to a relatively small range included in “3/4 left-front view”. Yet the viewpoint change is high enough to have important differences on the front of the car, such as the right headlight not always being visible.

The set Lₐ was used to design the individual detectors. We divided it into 5 folds to cross-validate the one-class SVM parameter C. The best parameter was used to train the classifier on the whole database. The signatures were then computed for all images in Lₜ using the detectors thus trained.

5.2 Probabilities

We used SVMs with RBF kernels and a two-level cross-validation on Lₜ. The first level is used to find the optimal SVM parameters C and σ. The second level is used to generate the probabilistic outputs for error estimation. All probabilities are regularized using (3), and a probability is computed for each consistent multilabel. These probabilities are shown in the graphs Fig. 6. Each node in this graph corresponds to a multilabel, and each link from yᵢ to yⱼ denotes the fact that multilabel yᵢ is included in multilabel yⱼ with only one more label in yⱼ. A green node means probability near to 1, whereas a red node means probability 0. The figure shows different scenarios : in graph 6(a), there is strong confidence on the output; in graph 6(b), the result is more ambiguous, and even low confidence thresholds are likely to give a low complexity output.

5.3 Classification Results

As a first step, we test the algorithm performance in a classification framework. The error/complexity curve obtained with our method is shown Fig. 7 along with our adaptation of Marszalek and Schmid’s algorithm [23]. The algorithm proposed by [23] is adapted to trees or taxonomies. At node A they train a binary SVM for each of its child nodes Bᵢ using positive and negative sets P and N:

\[ P = \text{supp}(Bᵢ), \quad N = \text{supp}(A) - \text{supp}(Bᵢ), \]

where supp(X) is the set of samples belonging to category X. The structure we are working on is the combination of different trees sharing some of their nodes (see Fig. 2). Thus their algorithm can be applied in each underlying tree to get a set of multilabels. The confidence threshold p is used with SVM probabilistic outputs as a stopping criteria: starting at each base node r, we descend the hierarchy while the classifier associated with the link returns a probability bigger than p, giving eventually an output yᵢ(\(p\)). For confidence p, taking as multilabel the union of the outputs from the different hierarchies might not be consistent. In this case, we impose a consistent multilabel output by taking the one of greatest complexity in the union of the multilabels yᵢ(\(p\)):

\[ \hat{y}(p) = \arg\max_{y} \{ C(y) | y \subseteq \bigvee_{r \in \text{root}} yᵢ(\(p\)) \}. \]

We compute the mean complexity and mean loss (c(\(p\)), c(\(p\))) on the test set for several values of \(p\) to draw the curve in figure 7. The results show that our algorithm performs better, especially for higher complexities. For a mean complexity of 6 on the database, our algorithm gives a mean 0/1-loss rate of 49%, compared to 54% for [23].

5.4 Retrieval results

The principle of multiple annotation is tested on an image retrieval problem. The protocol conforms to a standard Google-like session. Queries are conjunctions of keywords and results consist of ranked lists of data. Since we are interested in a domain specific context with moderate size database, evaluation can rely on the knowledge of the entire database and on the computation of precision/recall curves.
The retrieval algorithm is a simple string matching comparing the query and the annotations. The possible queries are equal to the set of consistent annotations as explained in section 3. The global retrieval performance evaluation is based on examining the returned lists of data for various thresholds on the confidence coefficients.

Figure 8 shows the first 12 images found for queries of increasing semantic precision. The computations being done offline, these results are obtained instantaneously for the whole database. The corresponding Precision-Recall curves for the same multilabels are shown in Fig. 9, along with other related queries. The thick green curve is the average over all possible queries where each point is obtained by thresholding the returned confidence coefficients.

Those curves show large variations in retrieval behavior. The equal precision-recall points vary from 50% to 90% on Fig. 9, with an average at 68%. The performance decreases with the complexity of the query, although not strictly monotonically. It is also related to the number of items in each class, as was noticed also in [16], though some classes with few examples give also good results. This is not surprising, since the quality of annotations depends on learning, and therefore on the amount of available data in each class.

6. CONCLUSION AND FUTURE WORK

This article settles a framework for dealing with multiple level semantic annotation of images, allowing the setting of a trade-off between confidence and semantic precision. A complete processing chain to describe images with confidence-rated multilabels was presented. We defined a criterion for the evaluation of such multilabel classifiers. Both the comparison with an algorithm adapted from a hierarchical classification task and the tests in image retrieval showed promising results.

The work presented in this paper can be developed in several directions. The image signature can be improved using other types of features or using geometrical relations between local descriptors. A more interesting approach would be to use the domain knowledge to control the type of discriminant features in order to build a more “intelligent” signature.

7. ACKNOWLEDGEMENTS

This work was supported in part by the Agence Nationale de la Recherche project “Modèles Graphiques et Applications”.

Figure 8: First 12 images retrieved (from left to right and top to bottom) respectively for multilabels (a) Renault, (b) Supermini,Hatchback,Renault,Clio,ClioI and (c) Supermini,Hatchback,Renault,Clio,ClioII,ClioII-2. The system is able to retrieve objects from classes with relatively large intra-class variation as in (a) as well as distinguishing small variations. The only falsely retrieved image is marked with a red border.

Figure 9: Precision-Recall for multilabels at different semantic level. Retrieval is done on the entire database.
8. REFERENCES


