Weakly supervised learning of visual models and its application to content-based retrieval

Cordelia Schmid

Weakly supervised learning of visual models and its application to content-based retrieval

Cordelia Schmid

INRIA Rhône-Alpes, 655 av. de l'Europe, 38330 Montbonnot, France

Cordelia.Schmid@inrialpes.fr

Abstract

This paper presents a method for weakly supervised learning of visual models. The visual model is based on a two-layer image description: a set of "generic" descriptors and their distribution over neighbourhoods. "Generic" descriptors represent sets of similar rotational invariant feature vectors. Statistical spatial constraints describe the neighborhood structure and make our description more discriminant. The joint probability of the frequencies of "generic" descriptors over a neighbourhood is multi-modal and is represented by a set of "neighbourhood-frequency" clusters. Our image description is rotationally invariant, robust to model deformations and characterizes efficiently "appearance-based" visual structure. The selection of distinctive clusters determines model features (common to the positive and rare in the negative examples). Visual models are retrieved and localized using a probabilistic score. Experimental results for "textured" animals and faces show a very good performance for retrieval as well as localization.

Keywords: visual model, two-layer image description, weakly supervised learning

1 Introduction

The growing number of images has increased the need for tools which automatically determine image content. While tools based on keywords exist, they have two major drawbacks. Firstly, each image has to be described by keywords which is extremely time consuming. Secondly, the expressive power of keywords is limited and cannot be exhaustive. Consequently, a significant need for image content based tools exists, for example in stock photo agencies.

The first image retrieval systems were based on the comparison of global signatures, such as colour or texture histograms [16]. Results of these systems have shown to be unsatisfactory, as they do not represent the "semantic" image content; they are unable to find images containing instances of a model, as for example faces or zebras. More recent approaches learn visual models and localize them in the image. This issue has for example been addressed in the context of face detection [23, 28].

A visual model has to capture the variability of a set of training images. Note that training images include in general negative examples which improve the description of the distribution. Visual models differ in the image description and in the learning algorithm. Images are for example described by global greyvalue patches [24], geometric relations of parts [1, 28] or statistical models [10, 20]. Note that models based on geometric shape are limited to rigid, spatially similar objects, as for example faces and cars.

To learn a visual model, most approaches either determine a discriminant function or a generative model of the distribution. Discriminant functions can for example be learnt with a support vector machine [25] which has been successfully applied to the detection of pedestrians [17]. A simple way to describe distributions are Gaussian mixture models [5] which can for example be used to describe facial features [27]. The training process can be either supervised or weakly supervised. Supervised algorithms require the manual extraction of regions or features [20, 24]. In the weakly supervised case [19, 28] images are labelled as positive or negative which avoids time consuming manual intervention. In this case significant parts have to be determined auto-

matically which presents an additional difficulty. The benefit of weakly supervised learning of visual structure represented by "generic" descriptors and the joint probability of their frequencies over neighbourhoods has been described by Schmid [22]. This approach learns a flexible statistical image description and selects the significant structure of objects without manual intervention. It is described in detail in this paper.

The steps of the approach are the following:

1. Computation of "generic" descriptors (cf. section 2). Rotationally invariant "Gabor-like" feature vectors are extracted for all pixel locations. A clustering algorithm then groups similar vectors together, that is determines the "generic" descriptors.

2. Computation of the joint probability of frequencies of "generic" descriptors over neighbourhoods (cf. section 3). These probabilities are multi-modal and are represented by a set of "neighbourhood-frequency" clusters.

3. Selection of distinctive "neighbourhood-frequency" clusters (cf. section 4). This determines the visual model, that is background patterns are eliminated and distinctive model patterns are kept.

Our two-layer representation is able to represent textures, for example the stripes of a zebra, as well as highly structured patterns such as parts of a face. Note that both layers are invariant to rotation and that for example horizontal and vertical stripes of a zebra are grouped together. This makes the method robust to model deformations, as for example in the case of a zebra sitting instead of standing upright. To retrieve and localize instances of the visual model, we introduce a probabilistic score in section 5. Results are shown in section 6.

2 Generic descriptors

We represent local greyvalue structure by rotationally invariant feature vectors which are computed at each pixel location. These multi-dimensional feature vectors are

in the following referred to as greyvalue descriptors. Greyvalue structures can be repeated in the image (in the case of texture) or between images (similar visual structure); they can also be similar over a region. To summarize the information, it is therefore appropriate to form groups of similar descriptors and describe them by their mean and variance. These groups are obtained by clustering multi-dimensional feature vectors and are in the following referred to as "generic" descriptors. Similar descriptors have been proposed previously. Rikert et al. [20] use a simple clustering algorithm to extract clusters of similar descriptors from a large set of sample images and then select significant clusters. Malik et al. [14] use the k-means algorithm to cluster descriptors of one image. They call the centers "textons" and use histograms of "textons" for a compact texture representation.

2.1 Greyvalue descriptors

Our greyvalue descriptors $\mathbf{d}_l$ are computed for each image location $\mathbf{p}_l$. They are rotationally invariant and are obtained by convolution with isotropic "Gabor-like" filters. These filters combine frequency and scale:

$$F(x, y, \tau, \sigma) = F_0(\tau, \sigma) + \cos\left(\frac{\sqrt{x^2 + y^2}\,\pi\,\tau}{\sigma}\right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

where τ is the number of cycles of the harmonic function within the Gaussian envelope of the filter, commonly used in the context of Gabor filters [7]. $F_0(\tau, \sigma)$ is added to obtain a zero DC component. This makes the filters robust to illumination changes [4], as we obtain invariance to intensity translations. For our experiments we use 13 filters with scales σ between 2 and 10 and τ between 1 and 4. For smaller scales only small τ are used to avoid high frequency responses. A comparison of different descriptors [26] has shown that our descriptors perform equivalently well compared to [8, 20, 21] who use 24 filters or more. Furthermore, we have observed that our "Gabor-like filters" outperform rotational invariant combinations of derivatives [9].

Our experimental results show robustness to limited scale changes. However, invariance to scale changes requires the use of scale invariant descriptors [13, 15]. These descriptors use scale selection to determine the appropriate scale for computation.

Initial results for texture representation with scale invariant descriptors has shown very promising results in the presence of significant scale changes [12].

2.2 Extraction of "generic" descriptors

"Generic" descriptors are groups of similar greyvalue descriptors. These groups are obtained by clustering with a k-means algorithm [3]. We extract "generic" descriptors separately for the set of positive and negative sample images. Negative images are included to obtain a more descriptive set of "generic" descriptors which permit to eliminate non-model descriptors.

The k-means algorithm finds k centers such that after assigning each data vector to the nearest center, the sum of the squared distance from the centers is minimized. Note that the k-means algorithm will only achieve a local minimum of this criterion. Our algorithm first normalizes the descriptors $\mathbf{d}_l$ using their mean and variance to avoid scaling effects. The Euclidean distance is then used to compare descriptors. We iteratively choose k centers such that after assigning each data vector to the nearest center, the sum of the squared distance from the centers decreases. Once the algorithm has converged to k clusters with centers $\boldsymbol{\mu}_i$, the covariance matrix $\boldsymbol{\Sigma}_i$ of each cluster is computed using the descriptors assigned to it. Our k clusters are described by $C_i = (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$.

The number of clusters k depends on the context and is difficult to be chosen automatically. In the context of region segmentation, a small number of clusters is required, for example in [2] the number varies between 2 and 5. In this case clusters have a significant variance. This is not appropriate in our context, as such clusters are not sufficiently distinctive. A more significant number of clusters is therefore required [14, 20]. The cluster number k was set to 50 for all our experiments. This number has been determined experimentally.

Figure 1 shows three "generic" descriptors (clusters) for the cheetah image on the left. These clusters have been selected manually to illustrate object and background clusters. A cluster is represented by the image locations which have been assign to it, that is the descriptors computed at these locations are part of the cluster.

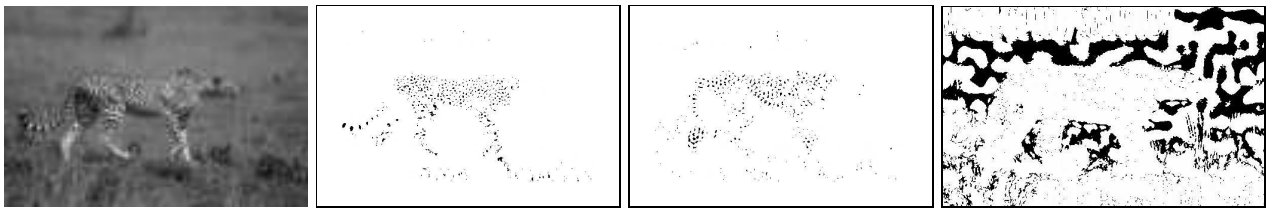


Figure 1: "Generic" descriptors for the cheetah image (on the left). The two images in the middle display "generic" descriptors which characterize the cheetah. The image on the right represents a "generic" descriptor of the background. See text for details.

2.3 Probability of a "generic" descriptor

We now define the probability of a "generic" descriptor C_i. For a pixel location $\mathbf{p}_l$ or equivalently for its greyvalue descriptor $\mathbf{d}_l$, the probability $P(C_i|\mathbf{d}_l)$ is defined by:

$$P(C_i|\mathbf{d}_l) = \frac{P(\mathbf{d}_l|C_i)P(C_i)}{P(\mathbf{d}_l)} = \frac{P(\mathbf{d}_l|C_i)P(C_i)}{\sum_{j=1}^{k} P(\mathbf{d}_l|C_j)P(C_j)} \tag{1}$$

The probability $P(\mathbf{d}_l|C_i)$ is computed by approximating the distribution of a "generic" descriptor $C_i = (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ with a Gaussian. Furthermore, we assume that the clusters C_i are equally probable. Both assumptions are of course only approximations. However, they have shown to give very good results and are therefore appropriate.

We can then select for each image location $\mathbf{p}_l$ and its descriptor $\mathbf{d}_l$ the most probable "generic" descriptor C^*, that is the one with the maximum probability $P(C_i|\mathbf{d}_l)$:

$$C^*(\mathbf{p}_l) = C^*(\mathbf{d}_l) = \operatorname*{argmax}_{C_i} p(C_i|\mathbf{d}_l) \tag{2}$$

The most probable cluster is stored in a label image at the corresponding pixel location. Labels vary from 1 to k with k the number of clusters.

3 "Neighbourhood-frequency" descriptors

We use a second layer of information to increase the distinctiveness of our representation. It is based on the "neighbourhood-frequency" descriptors which are more

distinctive than simple "generic" descriptors. A "neighbourhood-frequency" descriptor represents the frequencies of the "generic" descriptors over a neighbourhood (cf. equation 3). Note that such a descriptor ignores the geometric relationship of the "generic" descriptors and that it is rotationally invariant.

The joint probability of these frequency descriptors is multi-modal ; our experiments have shown that it is clearly not sufficient to describe the distribution by its mean and variance. We therefore represent the distribution by a set of clusters. Furthermore, we do not estimate the global joint probability, but the conditional joint probabilities with respect to the descriptor of the center location. This verifies the coherence of the neighbours with respect to the center and represents an additional constraint.

Most of the spatial constraints proposed previously are based on geometric shape information [1, 28]. Geometric shape constraints are valid for object classes which share features that are visually similar and occur in similar spatial configurations. Examples for such classes are faces or cars. Such constraints are not adapted for "textured" deformable objects such as animals, as they do not have similar spatial structure. The geometric structure of a cheetah for example is very different, if it is sitting or standing upright.

Distributions of descriptors over neighbourhoods have been previously used by Schneiderman and Kanade [23]. They use attribute histograms over neighbourhoods. Neighbourhoods are fixed with respect to a reference frame, that is the local distributions have to occur in similar spatial positions. Their model is therefore not adapted to deformable objects.

In the context of image segmentation, Malik et al. [14] compare windowed "texton" histograms, where the windows are centred around the two pixels being compared. This comparison decides on the presence of a region boundary. They do not attempt to learn a model.

3.1 Extraction of "neighbourhood-frequency" descriptors

In section 2.3 we have introduced a label image. Each label represents the most probable "generic" descriptor for the greyvalue descriptor computed at the image

location. The label image is used to compute for each image location the frequencies (probabilities) of the "generic" descriptors C_i over a neighbourhood:

$$\mathbf{v}_l = \begin{pmatrix} P(C_1|\mathbf{w}_l) \\ P(C_2|\mathbf{w}_l) \\ \cdots \\ P(C_k|\mathbf{w}_l) \end{pmatrix} = \begin{pmatrix} \frac{|\{C^*(\mathbf{p})=C_1|\mathbf{p}\in\mathbf{w}_l\}|}{|\mathbf{w}_l|} \\ \frac{|\{C^*(\mathbf{p})=C_2|\mathbf{p}\in\mathbf{w}_l\}|}{|\mathbf{w}_l|} \\ \cdots \\ \frac{|\{C^*(\mathbf{p})=C_k|\mathbf{p}\in\mathbf{w}_l\}|}{|\mathbf{w}_l|} \end{pmatrix} \tag{3}$$

where $\mathbf{w}_l$ is a window centred on the pixel location $\mathbf{p}_l$ and $|\mathbf{w}_l|$ is the number of pixels in the window. Note that the "generic" descriptor of the center location is not included, as it is used to compute the conditional joint probability. For our experiments we have used a circular window of radius 10.

The set of frequency vectors $\mathbf{v}_l$ with the same center label C_i (same most probable cluster at the center) represents the conditional joint probability of frequencies with respect to the center $P(\mathbf{v}_l|C^*(\mathbf{p}_l) = C_i)$. This set of frequency vectors is in the following denoted by V_i. The distribution of V_i is multi-modal and the different modes of the distribution are described by a set of clusters $\{V_{ij}\}$. These clusters are obtained with the k-means algorithm; frequency vectors $\mathbf{v}_l$ are compared with the Euclidean distance. Each cluster represents statistically similar neighbourhoods. These clusters are in the following referred to as "neighbourhood-frequency" descriptors. For our experimental results 10 clusters are determined for each "generic" descriptor.

3.2 Probability of a "neighbourhood-frequency" descriptor

Each image location $\mathbf{p}_l$ is assigned the most probable "generic" descriptor and a neighbourhood descriptor $\mathbf{v}_l$ is computed from these assignments (cf. equation 3). In the following we determine the most similar "neighbourhood-frequency" descriptor V_{ij} for an image location $\mathbf{p}_l$. The probability $P(V_{ij}|\mathbf{p}_l)$ of a descriptor V_{ij} at an image location $\mathbf{p}_l$ is given by $P(V_{ij}|\mathbf{v}_l \wedge \mathbf{d}_l)$:

$$\begin{aligned} P(V_{ij}|\mathbf{v}_l \wedge \mathbf{d}_l) &= \frac{P(\mathbf{v}_l \wedge \mathbf{d}_l|V_{ij})P(V_{ij})}{P(\mathbf{v}_l \wedge \mathbf{d}_l)} \\ &= \frac{P(\mathbf{v}_l|\mathbf{d}_l \wedge V_{ij})P(\mathbf{d}_l|V_{ij})P(V_{ij})}{\sum_s \sum_t P(\mathbf{v}_l|\mathbf{d}_l \wedge V_{st})P(\mathbf{d}_l|V_{st})P(V_{st})} \end{aligned} \tag{4}$$

We assume in the following that the $P(V_{ij})$ are equal and that the distribution of a "neighbourhood-frequency" cluster is approximated with a Gaussian $(\boldsymbol{\mu}_{ij}, \boldsymbol{\Sigma}_{ij})$. We have $P(\mathbf{d}_l|V_{ij}) = P(\mathbf{d}_l|C_i)$ and $P(\mathbf{v}_l|\mathbf{d}_l \wedge V_{ij})$ is defined by :

$$P(\mathbf{v}_l|\mathbf{d}_l \wedge V_{ij}) =$$

$$\begin{cases} G(\mathbf{v}_l; \boldsymbol{\mu}_{\mathbf{ij}}, \boldsymbol{\Sigma}_{\mathbf{ij}}) & \text{if } C^*(\mathbf{d}_l) = C_i \\ 0 & \text{otherwise} \end{cases}$$

Note that we need to evaluate $P(V_{ij}|\mathbf{v}_l \wedge \mathbf{d}_l)$ only if $C^*(\mathbf{d}_l) = C_{\mathbf{i}}$, otherwise its value is zero. Equation 4 then simplifies to:

$$P(V_{\mathbf{ij}}|\mathbf{v}_l \wedge \mathbf{d}_l) = \frac{P(\mathbf{v}_l|V_{\mathbf{i}j})}{\sum_s P(\mathbf{v}_l|V_{\mathbf{i}s})}$$

To compute the significance as well as the retrieval score we select for each image location $\mathbf{p}_l$ the most probable "neighbourhood-frequency" descriptor V^*, that is the one with the maximum probability $P(V_{ij}|\mathbf{p}_l)$:

$$V^*(\mathbf{p}_l) = V^*(\mathbf{v}_l \wedge \mathbf{d}_l) = \operatorname*{argmax}_{V_{ij}} p(V_{ij}|\mathbf{v}_l \wedge \mathbf{d}_l) \tag{5}$$

4 Significance

The distinctiveness of each "neighbourhood-frequency" descriptor determines its significance for the visual model. We can categorize "neighbourhood-frequency" descriptors as positive and distinctive, positive and not distinctive, background (non relevant parts of the positive sample images) and negative.

We want to identify descriptors which are positive and distinctive. Intuitively, "neighbourhood-frequency" descriptors which appear often in the positive examples and rarely in the negative samples fall into this category. This is captured by our significance measure defined in the following. Note that it is fundamental that non-significant descriptors are part of the model. These descriptors are matched to background or negative images and the significance measure eliminates them without using an arbitrary distance threshold. This avoids false positive responses for test images

which do not contain the model. The importance of negative clusters has been confirmed by Sung and Poggio [24] in the context of learning the distribution of global face patches. Note that the negative distribution can not be characterized in general, but only in a specific context, i.e. for a given set of examples. One-class classifiers [11] avoid this problem and present a possible solution for the general case.

In the following we determine which of the "neighbourhood-frequency" descriptors are significant for the model. For each cluster V_{ij} we compute its probability for the positive and negative sample images separately. Given a set of m sample images, for which the probabilities are assumed independent and equal ($P(I_j) = 1/m$), we obtain:

$$P(V_{ij}|\{I_1, I_2 ... I_m\}) = \frac{1}{m} \sum_{q=1}^{m} P(V_{ij}|I_q) \tag{6}$$

To compute the probability of a "neighbourhood-frequency" descriptor for an image, we assume the n pixel locations $\mathbf{p}_l$ to be independent and equally probable ($P(\mathbf{p}_l) = 1/n$). Independence is not valid in the case of adjacent pixels. However, modeling inter-pixel dependence is complex and the independence assumption has shown to give very good results and is therefore appropriate. The pixel locations $\mathbf{p}_l$ are described by the descriptors $\mathbf{d}_l$ and the "neighbourhood-frequencies" $\mathbf{v}_l$:

$$P(V_{ij}|I) = P(V_{ij}|\{\mathbf{p}_1, \mathbf{p}_2, \ldots \mathbf{p}_n\}) = \frac{1}{n} \sum_{l=1}^{n} P(V_{ij}|\mathbf{p}_l) = \tag{7}$$

$$\frac{1}{n} \sum_{l=1}^{n} \begin{cases} P(V_{ij}|\mathbf{p}_l) & \text{if } V^*(\mathbf{p}_l) = V_{ij} \\ 0 & \text{otherwise} \end{cases}$$

Note that we only include the probability of the most probable "neighbourhood-frequency" descriptor. This avoids the accumulation of insignificant probabilities and corresponds to the retrieval algorithm which takes into account only the most probable descriptor. The above equations compute the probability of a descriptor V_{ij} for a set of positive sample images $P(V_{ij}|\{I_{pos}\})$ as well as for a set of negative sample images $P(V_{ij}|\{I_{neg}\})$. The significance of descriptor V_{ij} for a model M is then defined

as follows:

$$\mathbf{Sig}(V_{ij}|M) = \frac{P(V_{ij}|\{I_{pos}\})}{P(V_{ij}|\{I_{pos}\}) + P(V_{ij}|\{I_{neg}\})}$$

The values of this significance measure vary between 0 and 1. If the value is close to one, the "neighbourhood-frequency" descriptor is significant, that is relevant for the model. For example a "spatial" descriptor which has close to zero probability in the negative images and high probability in all or most of the positive examples is significant.

5 Retrieving images

In the previous sections we have constructed a visual model M from a set of positive and negative images. This model is described by a set of "generic" descriptors, a set of "neighbourhood-frequency" descriptors and the significance of each "neighbourhood-frequency" descriptor. In the following we want to retrieve images which contain instances of this visual model as well as localize instances of the model in these images.

We retrieve and localize instances of a model using a probabilistic score. The first step is to compute the model probability for an individual pixel $P(M|\mathbf{p}_l)$. This probability is based on the most probable "generic" descriptor and the most probable "neighbourhood-frequency" descriptor. $P(M|\mathbf{p}_l)$ is determined as follows:

1. For pixel location $\mathbf{p}_l$ we compute its descriptor $\mathbf{d}_l$.

2. For descriptor $\mathbf{d}_l$ we obtain the probabilities $P(C_i|\mathbf{d}_l)$ using equation (1). We then determine the most probable cluster $C^*(\mathbf{d}_l)$ as described by equation (2).

3. The "neighbourhood-frequency" descriptor $\mathbf{v}_l$ is computed for the neighbourhood of pixel $\mathbf{p}_l$. Note that for each pixel in the neighbourhood, the most probable "generic" descriptor has to be determined.

4. The probabilities $P(V_{ij}|\mathbf{v}_l \wedge \mathbf{d}_l)$ are computed using equation (4) and the most probable "neighbourhood-frequency" descriptor $V^*(\mathbf{v}_l \wedge \mathbf{d}_l)$ is determined as described by equation (5).

5. If $\mathbf{Sig}(V^*(\mathbf{p}_l)|M)$ is below a threshold t, the probability $P(M|\mathbf{p}_l)$ is set to zero. t

equals 0.5 in our experiments, that is $\mathbf{p}_l$ is rejected if it is more likely to belong to a negative sample.

6. The score of a pixel is computed by

$$P(M|\mathbf{p}_l) = P(C^*(\mathbf{p}_l)|\mathbf{d}_l)P(V^*(\mathbf{p}_l)|\mathbf{v}_l \wedge \mathbf{d}_l)\mathbf{Sig}(V^*(\mathbf{p}_l)|M)$$

For retrieval we determine the probability of a model given an image. If the n pixel locations $\mathbf{p}_l$ are assumed independent and equivalently probable, this probability can be computed by :

$$P(M|I) = P(M|\{\mathbf{p}_1, \mathbf{p}_2, \ldots \mathbf{p}_n\}) = \frac{1}{n}\sum_{l=1}^{n} P(M|\mathbf{p}_l)$$

Note that the above equation summarizes pixel-based probability scores. It assumes independence of pixel locations which is in general not valid. We should obtain different scores if significant pixels are spread out over the image or localized in a region. This should be taken into account when computing our score and is currently under investigation.

To localize instance of models in images, we select pixels with high probabilities (see for example figure 4). Selecting such pixels is only a crude method which can easily be improved, for example by including region segmentation [18]. Our results are however already more than satisfactory.

6 Experimental results

For our experimental results we construct models from 15 sample images (5 positive and 10 negative). This corresponds to a realistic setting where negative examples are more easily available. The number of "generic" descriptors is 50 and the number of spatial clusters for one "generic" descriptor is 10. The radius of the neighbourhood window is 10.

Our database contains 600 images of the corel dataset and 60 face images. We learn and test 4 different models : a zebra model (more precisely a zebra texture

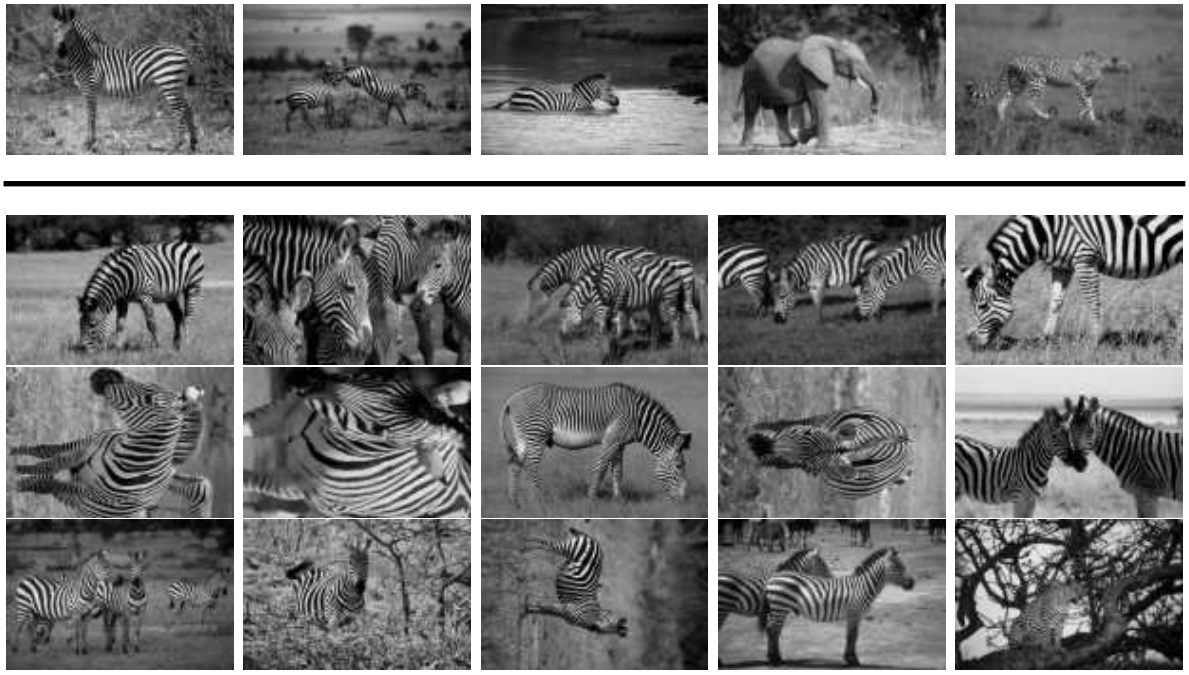


Figure 2: Retrieval results. The top row shows a subset of the training images (3 positive and 2 negative examples). The remaining rows show the first 15 retrieved images ordered by their score (from left to right and from top to bottom).

model), a cheetah model, a giraffe model and a face model. Our database contains approximatively 60 images of each category, 5 of which are part of the training set and excluded from the test set. Equivalently, negative examples of the training set are not included in the test set. Retrieval results are evaluated by computing precision as a function of recall. Precision is the number of relevant images retrieved relative to the total number of retrieved images. Recall is the number of relevant images retrieved relative to the total number of relevant images in the database.

The top row of figure 2 shows a subset of the training images (3 positive and 2 negative examples) used to learn the zebra model. The remaining rows display the first 15 retrieved images ordered by their probability score (from left to right and from top to bottom). The 14 most similar images are zebras; the 15th image is incorrectly retrieved. This incorrect retrieval is due to high probabilities for the branches which are visually similar to zebra stripes. The precision/recall graph is shown in figure 3. Results are comparable to those of other systems which manually extract objects.

Moreover, our method can localize the model in a retrieved test image by selecting locations with a high score. Results of localizing the zebra model are presented in figure 4. The locations with high scores are displayed in black. The body of the animal and three of its legs are correctly detected.

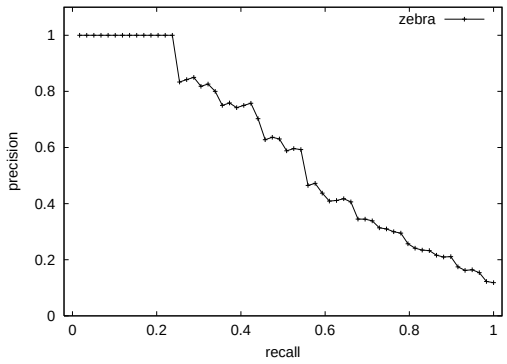


Figure 3: Precision as a function of recall for the zebra model.

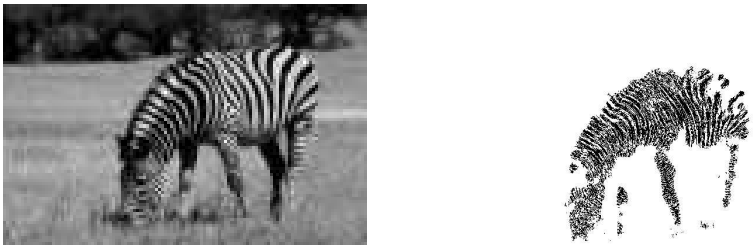


Figure 4: Localization of the zebra model for one of the test images (left). Locations with the high probability scores are displayed in black (right).

Figure 5 compares retrieval results for three different cheetah models; each model is learnt for a different set of training images. We can observe that the retrieval results for the different models are very similar. The performance does not depend on the initially chosen model images. Figure 6 displays a result for localization. We can notice that the three cheetahs are correctly localized. Equivalent results were obtained for the giraffe model.

Results for faces are displayed in figure 7. The graph for precision/recall is equivalent to those obtained for "textured" animals. The performance is in fact slightly better. This can be explained by the test images used: the faces have all the same

size and are in front of a simple background. Figure 8 shows a result for localization as well as a few of the "neighbourhood-frequency" clusters of the face (selected manually). Note the quality of these clusters. Eyes and mouth correspond to separate clusters and are well localized. The eyes are represented by two clusters, one for the inner part and one for the outer part of the eye.

Compared to an existing face detector [23], our results are not as good. However, this approach has been explicitly designed for faces, while our approach is general and uses only a weak neighbourhood structure. Our goal was to show that our representation is appropriate for textured objects as well as for highly structured ones. Note that we can represent any kind of texture which can be characterized by statistical neighbourhood distributions, that is regular or stochastic textures.

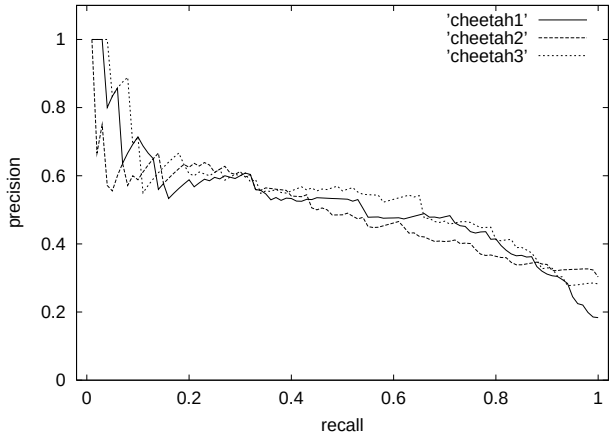


Figure 5: Precision as a function of recall for three different cheetah models. Each model is learnt from a different set of training images.

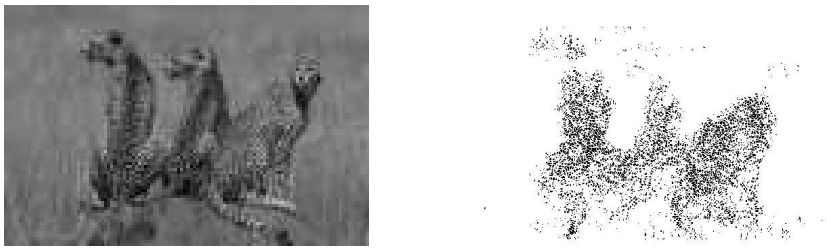


Figure 6: Localization of the cheetah model for one of the test images (left). Locations with high probability scores are displayed in black (right).

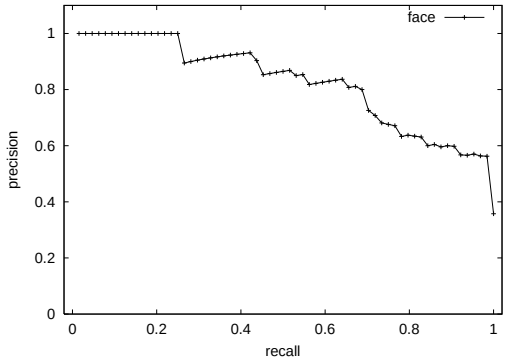
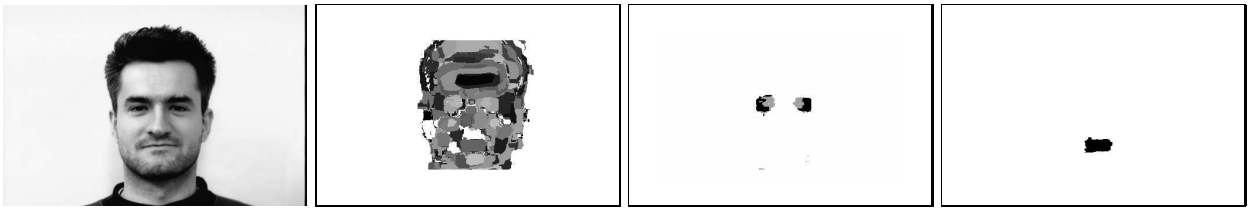


Figure 7: Precision as a function of recall for the face model.



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

Figure 8: (b) Localization of the face model for one of the test images (a). Each "neighbourhood-frequency" cluster is represented by a different greyvalue. (c) and (d) show three of these clusters: (c) Two clusters which correspond to the eyes. (d) A cluster which corresponds to the mouth.

7 Conclusion and discussion

We have presented a novel approach for learning visual models which significantly improves on the state of the art. It presents the following three advantages. The first is our two-layer image description which captures efficiently "texture-like" visual structure. The second is the learning algorithm which is weakly supervised and therefore does not require manual extraction of objects or features. It automatically learns an appropriate representation of the model. The third is the independence of region segmentation and feature extraction which are never perfect.

Finally, we mention four extensions which we are currently investigating. The first is to learn which components of our multi-valued generic descriptors are significant, that is most appropriate to describe the object. The second is to improve the clustering algorithm and to automatically select the number of clusters. The third is to include global constraints, for example by modelling relations between parts [6] or by segmenting regions [18]. The fourth extension is to add relevance feedback, that is to improve the model over time by user interaction or by tracking in video sequences.

References

[1] Y. Amit and D. Geman. A computational model for visual selection. *Neural Computation*, 11(7):1691–1715, 1999.

[2] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color-and texture-based image segmentation using EM and its application to content-based image retrieval. In *Proceedings of the 6th International Conference on Computer Vision, Bombay, India*, pages 675–682, January 1998.

[3] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[4] A. Cozzi, B. Crespi, F. Valentinotti, and F. Worgotter. Performance of phase-based algorithms for disparity estimation. *Machine Vision and Applications*, 9(5/6):334–340, 1997.

[5] R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* Wiley-Interscience, 1973.

[6] D.A. Forsyth and M.M. Fleck. Body plans. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Puerto Rico, USA*, pages 678–683, 1997.

[7] D. Gabor. Theory of communication. *Journal I.E.E.*, 3(93):429 – 457, 1946.

[8] A.K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, December 1991.

[9] J.J. Koenderink and A.J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[10] S. Konishi and A.L. Yuille. Statistical cues for domain specific image segmentation with performance analysis. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA*, pages 125–132, 2000.

[11] C. Lai, D. Tax, R. Duin, E. Pekalska, and P. Paclik. One-class classifiers for image database retrieval. In *Multiple Classifier Systems*, pages 212–221, 2002.

[12] S. Lazebnik, C. Schmid, and J. Ponce. Sparse texture representation using affine-invariant neighborhoods. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, 2003.

[13] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.

[14] J. Malik, S. Belongie, J. Shi, and T. Leung. Textons, contours and regions : Cue integration in image segmentation. In *Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece*, pages 918–925, 1999.

[15] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, pages 525–531, 2001.

[16] W. Niblack, R. Barber, W. Equitz, M. Fickner, E. Glasman, D. Petkovic, and P. Yanker. The QBIC project: Querying images by content using color, texture and shape. In *Proceedings of the* SPIE *Conference on Geometric Methods in Computer Vision II, San Diego, California, USA*, February 1993.

[17] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.

[18] N. Paragios and R. Deriche. Geodesic active regions for supervised texture segmentation. In *Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece*, pages 926–932, 1999.

[19] A.L. Ratan, O. Maron, W.E.L. Grimson, and T. Lozano-Pérez. A framework for learning query concepts in image classification. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, USA*, pages 423–429, 1999.

[20] T.D. Rikert, M.J. Jones, and P. Viola. A cluster-based statistical model for object detection. In *Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece*, pages 1046–1053, 1999.

[21] Y. Rubner and C. Tomasi. Texture-based image retrieval without segmentation. In *Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece*, volume 2, pages 1018–1024, 1999.

[22] C. Schmid. Constructing models for content-based image retrieval. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, 2001.

[23] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA*, volume I, pages 746–751, 2000.

[24] K.K. Sung and T. Poggio. Example-based learning for view-based human face detection. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51, 1998.

[25] V. Vapnik. *The nature of statistical learning theory.* Springer-Verlag, 1995.

[26] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination indepence. In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, volume III, pages 255–271, 2002.

[27] V. Vogelhuber and C. Schmid. Face detection based on generic local descriptors and spatial constraints. In *Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, Spain*, volume vol. 1, pages 1084–1087, 2000.

[28] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland*, pages 18–32, 2000.