

# Feature Selection for Automatic Image Annotation

Lokesh Setia and Hans Burkhardt

Albert-Ludwigs-University Freiburg  
79110 Freiburg im Breisgau, Germany  
{setia, burkhardt}@informatik.uni-freiburg.de

**Abstract.** Automatic image annotation empowers the user to search an image database using keywords, which is often a more practical option than a query-by-example approach. In this work, we present a novel image annotation scheme which is fast and effective and scales well to a large number of keywords. We first provide a feature weighting scheme suitable for image annotation, and then an annotation model based on the one-class support vector machine. We show that the system works well even with a small number of visual features. We perform experiments using the Corel Image Collection and compare the results with a well-established image annotation system.

## 1 Introduction

The amount of available multimedia data is continuously on the rise. With this arises the need to be able to locate existing data effectively. Data which cannot easily be found is as good as lost. Multimedia search differs from text search in that the results are much more subjective, and exact matches are normally not possible. For digital images, a lot of research has been done in the field of “Content-Based Image Retrieval” (CBIR) in the past decade. A user typically searches a CBIR database using the **query-by-example** paradigm, and the CBIR system bases its search on visual features extracted from the image. A big obstacle for CBIR to gain mainstream acceptance has been the so-called *semantic-gap* problem [1,5], though it can be somewhat reduced using *relevance-feedback* techniques [2,3]. Another practical problem in CBIR is that the user may not have a query image available.

A metadata search system on the other hand bases its search on image metadata, such as date and place of creation, image size, other image acquisition parameters, and on image keyword-annotation. Here, the database images are typically manually annotated with keywords, a task which is very time-consuming and also subjective. For large databases, this is simply prohibitively expensive. Automatic Image Annotation tries to bridge these two approaches, in that it works on the content of the images, but gives the user a possibility to perform a metadata search. Of course, semantic-gap remains a problem here too.

We describe briefly some prior work in the field of automatic annotation. Barnard et al. [12] presented a scheme to link segmented image regions with

words, the results depending heavily on the quality of the segmentation. Julia Vogel [6] assigned semantically meaningful labels to local image regions formed by dividing the image into a rectangular grid. Li and Wang [11] gave a statistical modeling approach using a 2-D Multiresolution Hidden Markov Model for each keyword and choosing the keywords with higher likelihood values. Cusano et al. [13] use a multi-class SVM for annotation though their scheme can hardly be judged due to their very small vocabulary consisting of seven keywords.

In this paper we describe our annotation methodology which consists of a feature extraction, feature weighting, model evaluation and a keyword assignment routine. Note that we sometimes use the terms feature weighting and feature selection interchangeably, as once the system has given a weight to each feature, they can always be ranked to select the ones with the higher weights.

We describe briefly the outline of this paper. We first give a description of the visual features used, then present our feature weighting algorithm. Later we give a description of our model based on the one-class SVM, and present the results of the experiments. We conclude with a discussion and an outlook for possible improvements and future work.

## 2 Features

To demonstrate the effectiveness of the feature weighting and model evaluation modules, we use a small set of simple visual features comprising of the following:

**Colour Features:** Colour features are widely used for image representation because of their simplicity and effectiveness. We use color moments calculated on HSV images. For each of the three layers we compute the layer mean, layer variance and layer skewness respectively. This yields a 9-dimensional vector. Since this does not incorporate any interlayer information, we calculate three new layers SV, VH and HS non-linearly by point-wise multiplication of pairs from original layers and calculate the same 3 moments also for the new layers. The final 18-dimensional vector outperformed a 512-bin 3D Joint Colour Histogram in CBIR tests that we performed.

**Texture Features:** Texture features can describe many visual properties that are perceived by human beings, such as coarseness, contrast etc. [4]. We use the Discrete Wavelet Transformation (DWT) for calculating texture features. The original image is recursively subjected to the DWT using the Haar (db1) wavelet. Each decomposition yields 4 subimages which are the low-pass filtered image and the wavelets in three orientations: horizontal, vertical and diagonal. We perform 4 level of decompositions and for the orientation subimages we use the entropy  $(-\sum_{i=1}^L H(i) \cdot \log(H(i)))$ , with  $\mathbf{H} \in \mathbb{R}^L$  being the normalized intensity histogram of the subimage) as the feature, thus resulting in a 12-dimensional vector.

**Edge Features:** Shape features are particularly effective when image background is uncluttered and the object contour dominates. We use the edge-orientation histogram [8] which we compute directly on gray-scale images by first calculating the gradient at each point. For all points where the gradient magnitude

exceeds a certain threshold, the gradient direction is correspondingly binned in the histogram. We use an 18-bin histogram which yields bins of size 20 degrees each.

The final feature vector is a concatenation of the above three vectors and has a dimensionality of 48.

### 3 Feature Weighting

A large number of feature selection or feature weighting methods have been proposed in the machine learning literature. The interested reader can refer to [7] for an overview of some of the popular alternatives. The main distinction is between the so called *Filter* methods, which compute a ranking for the features without taking the inducer (classifier) into account, and the *Wrapper* methods, which search in the set of subsets of features for the optimum subset for the specific inducer.

We propose a feature weighting method suitable for the image annotation problem. Image annotation with keywords can be interpreted as a classification problem but with two distinct characteristics: a) The number of classes (keywords) can be very large, and b) An image object can belong to multiple classes simultaneously (in other words, an image is usually annotated with multiple keywords). Thus, traditional feature weighting methods for multi-class classification are not only overloaded with the high number of classes, but would also give incorrect weights because of the overlap between the classes.

Our final aim is to learn a model for each class (keyword) based on a few training images. If we consider the training data for all the classes collectively, the properties of the ensemble become evident: the classes overlap, data belonging to the positive class (the class in question) is limited, but the data belonging to the negative classes is huge and spread around the feature space. Thus a multi-class classifier or a feature selection method based on it would not easily find decision boundaries or relevant features. We show that it is indeed possible to weight the features effectively for each class, taking into account the general distribution of the features. Let us start with a short data terminology. Let the training samples belonging to the positive class be given through

$$\mathbf{x}_1, \dots, \mathbf{x}_l \in \mathbb{R}^n$$

and the training examples in all the negative classes through

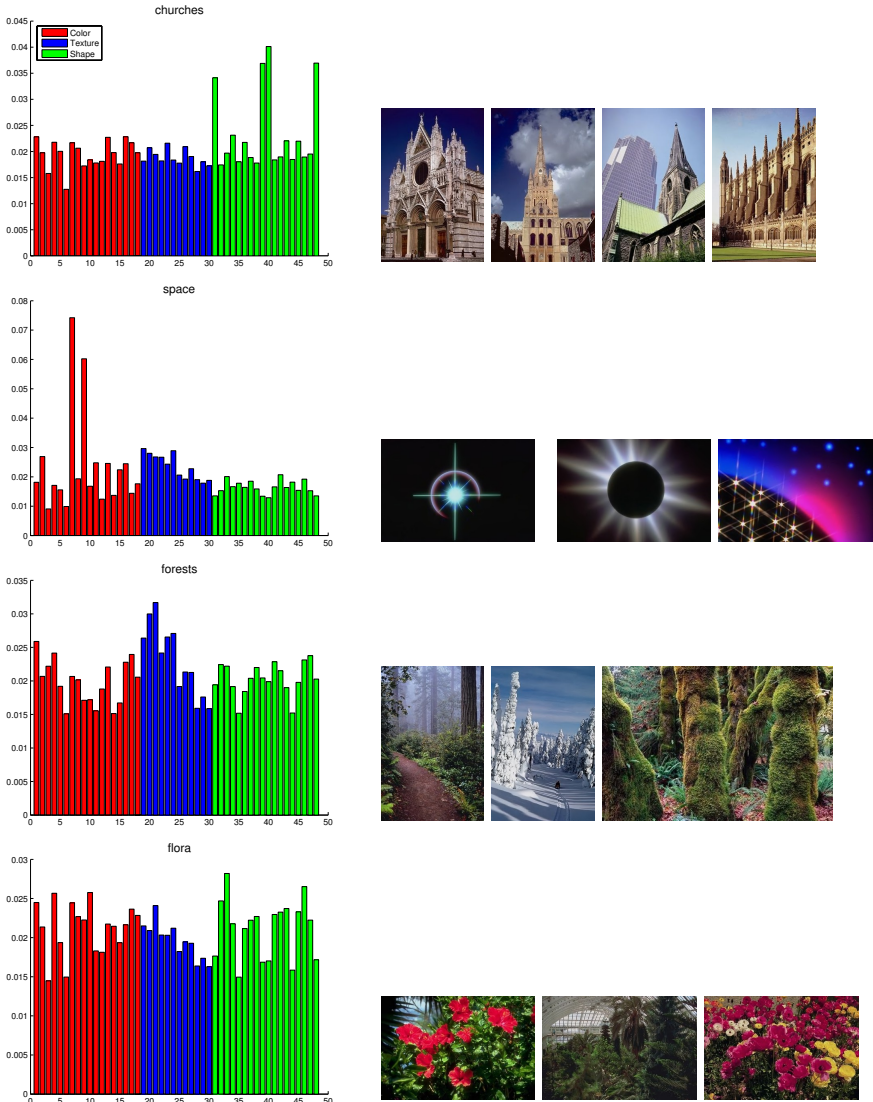
$$\mathbf{x}_{l+1}, \dots, \mathbf{x}_{l+m} \in \mathbb{R}^n$$

with  $m \gg l$ . Furthermore, we represent the  $i$ -th feature vector through the notation

$$\mathbf{x}_i = [x_i^{(1)} x_i^{(2)} \dots x_i^{(n)}]$$

All features are first normalised to zero mean and unit variance. Then, we estimate the distribution for each feature independently using the complete training data. We use a gaussian mixture model with three components to estimate the density,

$$p(x^{(k)}) = \sum_{j=1}^J \pi_j N(x^{(k)} | \Theta_j)$$



**Fig. 1.** Feature Weights for four sample Corel categories. Each row plots the feature weights and a few sample images for the category. The order of features in each graph is as follows 1) 18 Colour features: Mean, Variance and Skewness of Hue layer, followed by that of S, V, HS, SV and VH layers. 2) 12 Texture features: Entropy of H, V and D first level decomposition, followed by  $2^{nd}$  and  $3^{rd}$  levels 3) 18 Edge Features: Bins starting from  $0^\circ$  degrees in anti-clockwise direction with each bin having a span of  $20^\circ$ .

where  $N$  is the normal distribution with parameters  $\Theta_j = (\mu_j, \sigma_j)$ , and  $\pi_j$  is the weight of the  $j$ -th component, with  $\sum_{j=1}^J \pi_j = 1$ . The density is estimated using the expectation-maximization method.

We define the average likelihood for feature  $k$ , averaged only over the images of the positive class as

$$\text{avg}_k = \frac{\sum_{i=1}^l p(x^{(k)} = x_i^{(k)})}{l}$$

The higher the average likelihood is, the more similar this feature is between the positive and the negative classes and therefore less discriminative. Thus, we define the weight for the  $k$ -th feature as

$$w_k = 1/\text{avg}_k$$

The weights are normalized so that  $\sum_{k=1}^n w_k = 1$ . This has the effect that all models deliver optimum performance (tested through crossvalidation) for about the same model parameters. The features are then weighted with  $w_k$  before fed to the model computation routine (each model gets its own sets of weights). We show now that the weighting scheme is effective and the weights can in fact even be directly interpreted for our features. To do this, we plot in Fig. 1 the calculated weights for the 48 features for 4 corel categories: **churches**, **space1**, **forests** and **flora**. The training data consisted of 40 images each in the positive class and the complete Corel collection of 60,000 images as the negative (Note that it is immaterial here if the positive images are considered for determining the gaussian mixture distribution or not, as we have a very large number of samples available from the stochastic process). The sequence of the 48 features is explained in the figure caption.

For the **churches** category, the maximum weight went to the edge features corresponding to the directions  $0^\circ$  and  $180^\circ$ , i.e., the discriminative vertical edges present in churches and other buildings (most images in the category were taken upright). For the **space1** category, the most discriminative feature the system found was the 7<sup>th</sup> feature, which is the mean of the brightness ( $V$ ) component of the image (the images in the category are mostly dark). For the **forests** category, texture features get more weight, as does the hue component of the colour features. We however did find some categories where the weights were somewhat counter-intuitive or difficult to interpret manually. An example is the category **flora** in part d).

## 4 Model Computation

We assume that the presence or absence of a keyword in an image can be tested independently of other keywords. Though it is not necessarily true, it is a reasonable assumption to keep the complexity of the overall system in check. Otherwise, the system would need access to the conditional probabilities of keywords given the presence of other keywords.

We propose a slightly modified one-class Support Vector Machine (SVM) as our model. One-Class SVM were introduced by Schölkopf et al. [9]. One-Class

SVMs are binary functions which capture regions in the input space where the probability density lies (i.e. its support). We train a one-class SVM for every keyword with the aim to determine subspaces in the feature space where most of the data for that keyword is present.

One-Class SVMs are the solution to the following optimization problem: Find a hypersphere in  $\mathbb{R}^n$  which contains most of the training data and is at the same time as small as possible. This can be written in primal form as:

$$\min_{R \in \mathcal{R}, \zeta \in \mathcal{R}^l, \mathbf{c} \in \mathcal{F}} R^2 + \frac{1}{\nu l} \sum_i \zeta_i$$

subject to

$$\|\phi(\mathbf{x}_i) - \mathbf{c}\|^2 \leq R^2 + \zeta_i, \quad \zeta_i \geq 0, \quad i = 1, \dots, l$$

$\phi(\mathbf{x}_i)$  is the  $i$ -th vector transformed to another (possibly higher-dimensional) space using the mapping  $\phi$ .  $\mathbf{c}$  is the center and  $R$  the radius of the hypersphere in the transformed space. With the kernel trick [10] it is possible to work in the transformed space without ever calculating the map  $\phi(\mathbf{x}_i)$  explicitly. This can be achieved by defining a kernel function  $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$  as the algorithm needs access only to scalar products between vectors, and not to the actual vectors themselves.

The tradeoff between the radius of the hypersphere and the number of outliers can be controlled by the single parameter  $\nu \in (0, 1)$ . Using Lagrange multipliers, the above can be written in the dual form as:

$$\min_{\alpha} \sum_{i,j} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) - \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}_i)$$

subject to

$$0 \leq \alpha_i \leq \frac{1}{\nu l}, \quad \sum_i \alpha_i = 1$$

The optimal  $\alpha$ 's can be computed with the help of QP optimization algorithms. The decision function then is of the form

$$f(\mathbf{x}) = \text{sign}(R^2 - \sum_{i,j} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) + 2 \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) - k(\mathbf{x}, \mathbf{x}))$$

This function returns positive for points inside this hypersphere and negative outside (note that although we use the term hypersphere the actual decision boundary in the original space can be varied by choosing different kernel functions. We use a gaussian kernel  $k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$ , with  $\gamma$  and  $\nu$  determined empirically through cross-validation). Since we need a rank for each keyword in order to annotate the image, we leave out the `sign` function, so that the results can be sorted on the basis of their “positiveness”. Furthermore, it was found that the results are biased towards keywords whose training images are very dissimilar to each other, i.e., the models for which  $R^2$  term is high.

Compact models are penalised, and therefore we use the following function instead for model evaluation:

$$g(\mathbf{x}) = \frac{R^2 - \sum_{i,j} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) + 2 \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) - k(\mathbf{x}, \mathbf{x})}{R^2}$$

which can be interpreted as the normalized distance from the model boundary in the transformed space.

## 5 Experiments and Discussion

We perform our experiments similar to the ALIP system [11] to facilitate an objective comparison. The Corel Database with 600 categories is used. Each category is manually labelled<sup>1</sup> with few descriptive keywords (typically 3 to 5). Each category consists of 100 colour images of size  $384 \times 256$ , out of which we select 40 images randomly as training images. Normally for image annotation we would be training a model for every annotation keyword, and would annotate a query image with the keywords whose models evaluate the query image most favourably. For this experiment however, we learn a model for every Corel *category* instead of each *annotation keyword*. Then, for the best  $k$  category matches (we experiment with  $k = \{5, 8, 11, 14\}$ ), the category keywords are combined and the keywords least likely to have appeared by chance are taken for annotation, as in [11]. This scheme favours infrequent words like `waterfall` and `asian` over common ones like `landscape` and `people`.

To have an estimate of the discriminative performance of the system, we perform a classification task with the 600 categories. The system attains an accuracy of 11.3 % as compared to 11.88 % that of ALIP. However, as also pointed out in [11], many of the categories overlap (e.g. `Africa` and `Kenya`) and it is not clear how much can be read from this classification performance. Furthermore, we found that although the best category match was incorrect in the sense of the category ground truth, it was often meaningful with regard to the query image. We provide some annotation examples in Table 1.




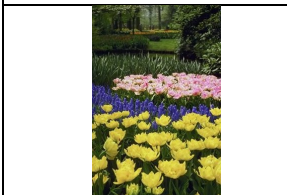


For a more controlled test, we take 10 distinct Corel Categories, namely `Africa`, `beach`, `buildings`, `buses`, `dinosaurs`, `elephants`, `flowers`, `horses`, `mountains` and `food`. The confusion matrix for this task is shown in Table 2. Overall, the system attains classification accuracy of 67.8% as compared to 63.6% attained in ALIP.

**Computation Time:** All experiments were performed on an Intel Pentium IV 2.80 GHz single-CPU machine running Debian Linux. Calculation of image features takes about 1.2 seconds per image. Model computation with 40 training vectors per model takes only about 20 msec per model. A new query image needs about 4 seconds to be fully annotated (this includes computation time for feature extraction, evaluation of 600 models, and decision on unlikely keywords), as compared to 20 minutes for the HMM-based approach in ALIP. This makes

---

<sup>1</sup> We thank James Wang for making the category annotation available for comparison.

**Table 1.** Sample annotation results from the system. The query images are taken from the Corel Collection but did not belong to the training set. It can be observed that while the original category was sometimes not found, it was due to the fact that the categories often overlapped, as the top matches do indeed contain very similar categories, leading to robust annotation results.

Query Image	Original Category	Top 8 Matches	Final Annotation
	africa	wildlife_rare, architect, shells, dogs, mammals, newzealand, 197000, pastoral	grass, animal, dog, rareanimal, shell, mammal, NewZealand, pastoral
	wl_ocean	plants, green, foliage, can_park, US_garden, flora, texture13, flower2	plant, flower, green, foliage, leaf, flora
	189000	tribal, 239000, thailand, 189000, groups, perenial, indonesia, work	people, cloth, guard, face, life, tribal
	holland	rural_UK, forest, zionpark, flowerbeds, plants, forests, perenial, flower2	tree, forest, flower, ruralEngland, Zion, flowerbed, perenial
	lizard1	microimg, design1, textures, texture1, skins, texture7, texture9, food2	texture, natural, microimage
	yosemite	canyon_park, isles2, US_parks, alaska, 126000, rural_UK, gardens, cal_sea	Alaska, mountain, park, landscape, garden, house, California



**Table 2.** Confusion Matrix for the 10-category classification task

%	Africa	beach	buildings	buses	dnsrs	elephants	flowers	horses	mnts	food
Africa	<b>66</b>	6	12	0	0	2	2	2	6	4
beach	16	<b>32</b>	28	0	0	8	2	2	6	6
buildings	6	6	<b>76</b>	2	0	0	2	0	6	2
buses	0	0	30	<b>64</b>	0	0	0	6	0	0
dinosaurs	0	0	2	0	<b>94</b>	0	0	0	0	4
elephants	28	0	0	0	0	<b>50</b>	0	8	12	2
flowers	10	0	4	0	0	0	<b>78</b>	0	4	4
horses	6	2	6	0	0	2	2	<b>72</b>	10	0
mountains	4	4	10	0	0	0	6	0	<b>70</b>	6
food	0	2	14	0	2	0	2	0	4	<b>76</b>

our system faster by a factor of 300 (or 100 taking the clock speed of the ALIP system into account). The system scales linearly with the number of models.

## 6 Conclusion and Future Outlook

A feature weighting method and a modelling scheme based on the one-class SVM for automatic image annotation was presented in this paper. It is clear that the power of the overall system is heavily dependant on the discriminative power of the used features. Thus, complex features should in general be expected to lead to a performance improvement. Local features extracted around interest points, e.g. [14], have recently given excellent results in the field of object recognition and could be directly plugged into the system (at least the methods which can return a single consolidated feature vector per image, instead of a bag of vectors).

It was shown that the modelling scheme scales well to larger number of keywords, both in terms of annotation results quality as well as the speed of execution. The system ran orders of magnitude faster than a MHMM-based scheme while giving comparable or better results. The effectiveness of the feature weighting was also demonstrated as the small number of visual features used lent themselves to direct interpretation.

A simplified view of the linguistic component of the annotation system was taken, as it lies outside the scope of this work. Also, currently the system does not check for mutually exclusive keywords or other inconsistencies, and ends up annotating the same image with combinations like *sunrise* and *sunset*, or with *England* and *Finland*. This can however be taken care of automatically to an extent by extracting conditional probabilities of keywords given the presence or absence of other keywords, given sufficient training data.

**Acknowledgements.** This work was supported by the German Ministry for Education and Research (BMBF) through grant FKZ 01IRB02B and by the Muscle Network of Excellence, through contract no. 507752.

## References

1. Smeulders et. al.. Content-based image retrieval at the end of the early years, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, no. 12, pp. 1349-1380. Dec. 2000.
2. Y Rui, TS Huang, M Ortega, S Mehrotra. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval, *IEEE transactions on circuits and systems for video technology*, vol. 8, no. 5, september 1998.
3. C Meilhac, C Nastar. Relevance Feedback and Category Search in Image Databases, *ICMCS*, Vol. 1, 1999.
4. H. Tamura, S. Mori, T. Yamawaki. Texture features corresponding to visual perception, *IEEE Trans. Systems Man Cybernet SMC8 (6)* (1978) 00.
5. T Westerveld. Image retrieval: Content versus context, *Content-Based Multimedia Information Access RIAO* 2000.
6. Julia Vogel. Semantic Scene Modeling and Retrieval, PhD Thesis. Zurich, October 2004.
7. Ron Kohavi and George H. John. Wrappers for Feature Subset Selection, *Artificial Intelligence* 97, 1-2, Pages 273-324, 1997.
8. A. Vailaya, A. K. Jain and H.-J. Zhang. On Image Classification: City Images vs. Landscapes , *Pattern Recognition*, vol. 31, pp 1921-1936, December, 1998.
9. Schölkopf, B., J.C. Platt, J. Shawe-Taylor, A.J. Smola and R.C. Williamson. Estimating the support of a high-dimensional distribution. Technical report No.(87) Microsoft Research (1999)
10. V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, N.Y., 1995
11. Jia Li and James Z. Wang. Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1075-1088, 2003.
12. Kobus Barnard et al., Matching Words and Pictures, *Journal of Machine Learning Research*, Vol 3, pp 1107-1135. 2003
13. C. Cusano, G. Ciocca, R. Schettini. Image annotation using SVM, *Proceedings of Internet imaging V*, Vol. SPIE 5304, pp. 330-338, 2004.
14. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *IJCV* 60 (2004) 91-110