

# Visual Classification of Images by Learning Geometric Appearances Through Boosting

Martin Antenreiter, Christian Savu-Krohn, and Peter Auer

Chair of Information Technology (CiT)  
University of Leoben, Austria

**Abstract.** We present a multiclass classification system for gray value images through boosting. The feature selection is done using the LPBoost algorithm which selects suitable features of adequate type. In our experiments we use up to nine different kinds of feature types simultaneously. Furthermore, a greedy search strategy within the weak learner is used to find simple geometric relations between selected features from previous boosting rounds. The final hypothesis can also consist of more than one geometric model for an object class. Finally, we provide a weight optimization method for combining the learned one-vs-one classifiers for the multiclass classification. We tested our approach on a publicly available data set and compared our results to other state-of-the-art approaches, such as the "bag of keypoints" method.

## 1 Introduction

Image recognition and categorization are interesting vision problems. There exist many approaches for solving specific problems (e.g. for face recognition). The task becomes more difficult if the goal is to develop an algorithm which is independent from the target object class. A state-of-the-art approach to overcome this problem is to use the "bag of keypoints" idea (see [5]). This method calculates a feature histogram for every image in the data set. Its main advantage is, that standard learning algorithms like SVMs [12,19], which need a fixed dimensional input vector, can be used to construct a classifier. On the other hand, feature histograms cannot exploit geometric relationships between the features contained in an image, although this might be discriminative information.

There exist various methods for incorporating such relationships between parts using statistical models. Early work in this direction was done by Burl et al. [2] for the recognition of planar object classes. There, important parts are selected by previously learned detectors, and afterwards a shape model is learned from the detector locations. This approach was later improved by using a soft-detection strategy in [3]. The two problems; detecting features, and building a shape model from the detection, are solved simultaneously. Furthermore, unsupervised scale-invariant learning of parts and shape models has been done in [7], where an entropy-based feature detector from Kadir [13] has been used to select the important parts from an image.

Recently, graph-based models called " $k$ -fans" were introduced [4]. The structure of the graph, and therefore the representational power of the shape model is controlled by the parameter  $k$ . There exist well defined algorithms to solve the learning and detection problems for models with  $k$ -fan graphs. In general, methods like [2,3,4,7] force the user to predefine a fixed number of parts considered for learning. This quantity is usually determined in a second run by trial and error. In contrast, we show how the correct number of parts as well as the geometric complexity for the model can be estimated during learning with a boosting algorithm.

Previous work from Opelt et al. [17,16] and Fussenegger et al. [9] have shown that image categorization using AdaBoost [8] is a powerful method. Particularly, they have used AdaBoost to select discriminative features to learn a classifier against a background class. This work extends their methods in several directions. First, it is not always clear beforehand which feature types are advisable for learning a certain class. Therefore, we use nine different feature types simultaneously, and leave it up to the learning algorithm to determine the useful types. To reduce computational efforts we cluster each feature type using  $k$ -means. Secondly, we use LPBoost [1,6] as the learning algorithm which is advantageous compared to AdaBoost, since LPBoost can handle noisy data well. Our third contribution is a procedure for incorporating geometric relations between features into the weak learner of the boosting algorithm. Finally, we address the multi-class classification problem and provide a weight optimization method for one-vs-one classifiers using Support Vector Machines (SVMs) [12,19]. We conclude with the evaluation and the results obtained on the Xerox image data set [5], which is publicly available at <ftp://ftp.xrce.xerox.com/pub/ftp-ipc/>. There, we also compare our results with those reported in the literature.

## 2 Classification of Images Through Boosting

In this Section we will present our method for learning a one-vs-one classifier. We will describe our feature extraction method as well as our preprocessing steps. Afterwards, we will give a short overview of the learning algorithm, and introduce an extension of the weak learner in order to manage geometric relations.

### 2.1 Feature Extraction

We use the scale invariant Harris-Laplace detector [15] to obtain regions of interest. From every region we extract four different feature types: scale invariant feature transforms (SIFTs) [14], sub-sampled grayvalues (see [17]), basic moments and moment invariants [11]. In addition to these descriptors, we use the segmentation method and the features of Fussenegger et al. [9]. For some feature types, we also normalize illumination by homomorphic filtering (see e.g. [10], Chap. 4.4.3). Furthermore, all features are normalized by whitening. Additionally, we obtain another feature type by reducing the SIFT-features to their 40 largest components using PCA, which accounts for their sparseness. Altogether, we use

**Table 1.** Feature types with preprocessing steps

$\phi$	feature type	intensity normal.	whitening	$m_\phi$	$k_\phi$
1	subsampled grayvalues		x	854 376	1 848
2		x	x	854 376	1 848
3	basic moments		x	852 755	1 846
4		x	x	854 376	1 848
5	moment invariants [11]		x	854 360	1 848
6		x	x	854 313	1 848
7	SIFTS [14]		x	809 063	1 798
8			PCA 40	809 063	1 798
9	segments [9]		x	690 070	1 661

nine different types of features  $\phi$ . In a second preprocessing step, we cluster the different features by k-means using  $k_\phi = \lfloor 2\sqrt{m_\phi} \rfloor$  centers with a random initialization from the data, where  $m_\phi$  denotes the number of features per type extracted from the database. Table 1 shows an overview of the calculated features.

## 2.2 LPBoost

We use a boosting approach since those algorithms are able to select important features from a large feature set. Instead of the common AdaBoost, we use LPBoost as the learning algorithm. One reason is that LPBoost has a well defined stopping criterion; learning is stopped if no further weak hypothesis will improve the value of the objective function for the current combination of weak hypotheses. Furthermore, AdaBoost is a hard margin classifier and therefore might overfit noisy data, whereas LPBoost is a soft margin classifier and handles noisy data well. The linear optimization problem in its primal formulation is:

$$\begin{aligned}
 \max_{\rho, a, \xi} \quad & \rho - D \sum_{n=1}^m \xi_n \\
 \text{s.t.} \quad & y_i \sum_{t=1}^T \alpha_t h_t(x_n) + \xi_i \geq \rho \quad i = 1, \dots, m \\
 & \sum_{t=1}^T \alpha_t = 1 \quad \alpha_t \geq 0 \\
 & \xi_i \geq 0 \quad i = 1, \dots, m
 \end{aligned} \tag{1}$$

and its dual is given by:

$$\begin{aligned}
 \min_{\beta, w} \quad & \beta \\
 \text{s.t.} \quad & \sum_{i=1}^m y_i w_i h_t(x_i) \leq \beta \quad t = 1, \dots, T \\
 & \sum_{i=1}^m w_i = 1 \quad 0 \leq w_i \leq D
 \end{aligned} \tag{2}$$

Thus, the final decision function is simply:

$$f(x_i) = \text{sign} \left( \sum_{t=1}^T \alpha_t h_t(x_i) \right) \in \{+1, -1\} \tag{3}$$

Note that the parameter  $D$  must be chosen carefully depending on the data set. An interpretation of the parameter and additional information can be found in Bennett et al. [1], Demiriz et al. [6] and Rätsch et al. [18].

### 2.3 Weak Learner

The weak learner is called in every boosting round and selects a hypothesis  $h^*$  from the hypothesis space  $\mathcal{H}$  which fulfills equation

$$\max_{h \in \mathcal{H}} \left( \sum_{i=1}^m h(x_i) y_i w_i \right) = \sum_{i=1}^m h^*(x_i) y_i w_i. \quad (4)$$

We implemented three different weak learners. The first and simplest one selects a reference feature of type  $\phi$  with an optimal threshold according to the current boosting weights  $\mathbf{w}_t$ . The second and third weak learners search for geometric relations between distinctive features. Note computational complexity is twofold when building geometric relations based on relative position of the features and their number.

Since a full search over all possible geometric directions is a computationally time consuming process, we use rather simple geometric relations. More precisely, our geometric primitives use four geometric directions (up, down, left, right) relating up to three reference features. If an object category requires a geometric relation consisting of more than three features, our search algorithms build hierarchies of such geometric primitives modeled as trees. These relations are denoted as 'relations A' throughout this paper. Furthermore, we build more complex geometric relations to distinguish between more directions, i.e. we divide our primitives into eight sections and denote those as 'relations B'. Note that our geometric relations are invariant to translation and scale but not to rotation.

To speed up computation, our weak learners use a greedy search strategy to find geometric relations [Fig. 1]. In particular, we combine the previous hypotheses only with the selected hypothesis  $h^*$  that has just one reference feature (see Fig. 1, Step 2a). This is reasonable due to (4). There might exist a better feature for a combined hypothesis  $h_{and}$ , but it would require a search through all features for every previous hypothesis to determine it. Nevertheless, we tested this search

1. Select a hypothesis  $h^*$  using equation (4) and current boosting weights  $\mathbf{w}_t$ .
2. For all previously generated hypotheses  $h_p$ ,  $p = 1, \dots, t - 1$  do:
  - (a) Create a hypothesis with a logical AND using the current simple weak hypothesis  $\rightarrow h_{and} = h^* \text{ AND } h_p$ .
  - (b) The hypothesis  $h_{and}$  is used for the geometric relations search. The two sub hypotheses from  $h_{and}$  are applied on every image yielding two point sets. We seek a common geometric relation between these sets, yielding a geometric hypothesis  $h_{geom}$ .
3. The weak hypothesis finder compares the performance of the simple weak hypothesis  $h^*$  and the geometric hypothesis  $h_{geom}$  and outputs the hypothesis with the best performance.

**Fig. 1.** Greedy search strategy for the weak learner

strategy on a subset of the data. We create an optimal hypothesis  $h_{and}$  for every previous hypothesis  $h_p$  by selecting an additional hypothesis  $h_{opt}$  with a reference feature, such that the hypothesis  $h_{and}$  achieves the least possible weighted error. Since this approach gives comparable results at higher computational cost, we use the faster greedy strategy proposed [Fig. 1].

Within every boosting iteration, the weak learner either builds a simple or geometric hypothesis. During the incremental construction of the geometric hypotheses, various geometric sub hypotheses are generated. If such a sub hypothesis is useful with respect to the training set, LPBoost incorporates it into the final decision function by assigning a positive weight  $\alpha_t$  to it; otherwise  $\alpha_t$  will be set to zero. Hence, the final classifier can contain more than one geometric hypothesis per object. In consequence we do not have to flip input images to guarantee that the objects always face the same way (e.g. motorbikes, airplanes), but rather to ensure that there are sufficient examples for all the important orientations in the data set.

### 3 Multiclass Image Classification

Within our experiments for multiclass classification, we noticed low performance using one-vs-all and hierarchic classifiers. Considering the object categories of this database, it is likely that the extracted features are shared within different classes. Actually, Csurka et al. [5] do achieve good results learning feature histograms with a one-vs-all strategy. Nevertheless, feature histograms cannot exploit geometric relationships between the features contained in an image, although this might be discriminative information. Hence, we chose a one-vs-one strategy and combine our individual classifiers by a voting scheme.

Simple voting methods like majority voting using hard labels, not only ignore available information about the different degrees of confidence in the different classifiers, but also the classifier's confidence in its own prediction. Hence, a weighted voting scheme incorporating such information seems more reasonable.

An appropriate way to measure a classifier's confidence in its prediction is the signed distance

$$\delta(x_i) = \sum_{t=1}^T \alpha_t h_t(x_i), \quad (5)$$

with  $\delta(x_i) \in [-1, 1]$ , of a data point  $x_i$  to the decision boundary. In this case, a great magnitude of  $\delta(x_i)$  reflects high confidence in a prediction. Thus, for an  $r$ -class problem upon  $m$  images  $x_i$  ( $i = 1, \dots, m$ ), we denote the predictions of the  $r \cdot (r - 1)$  different classifiers by

$$\begin{aligned} \mathbf{c}_i &= (\delta_{1,2}(x_i), \delta_{2,1}(x_i), \dots, \delta_{r-1,r}(x_i), \delta_{r,r-1}(x_i))^T \\ &\in [-1, 1]^{r \cdot (r-1)} \end{aligned} \quad (6)$$

Addressing the overall confidence in each classifier w.r.t. a certain class  $l$ , we try to find optimal weights  $\mathbf{w}_l \in \mathbb{R}^{1 \times r \cdot (r-1)}$  with  $l = 1, \dots, r$  and some  $\mathbf{b} \in \mathbb{R}^r$  such that the overall vote

$$l = \arg \max_{l'} \mathbf{w}_{l'} \cdot \mathbf{c}_i + b_{l'} \quad (7)$$

corresponds to the true class.

Hence, we formulate the following quadratic problem which gives a linear SVM:

$$\begin{aligned} \min \quad & \|(\mathbf{w}_1, \dots, \mathbf{w}_r)\|^2 + C \cdot \sum_i \xi_i \\ \text{s.t.} \quad & \mathbf{w}_l \cdot \mathbf{c}_i + b_l \geq 1 - \xi_i, & l = \text{class}(x_i) \\ & -\mathbf{w}_l \cdot \mathbf{c}_i - b_l \geq 1 - \xi_i, & \forall l : l \neq \text{class}(x_i) \\ & \xi_i \geq 0 & i = 1, \dots, m, \\ & & l = 1, \dots, r \end{aligned} \quad (8)$$

where, similar to (1), the amount of slackness over all predictions  $\mathbf{c}_i$  is controlled by the parameter  $C$ .

## 4 Evaluation and Results

For our experiments we used the Xerox database consisting of 1774 real-world images from seven different categories. The categories are faces (790), buildings (150), trees (150), cars (201), phones (216), bikes (125) and books (142). The numbers in brackets indicate the number of images per category.

**Table 2.** Accuracy upon 10-fold cross-validation

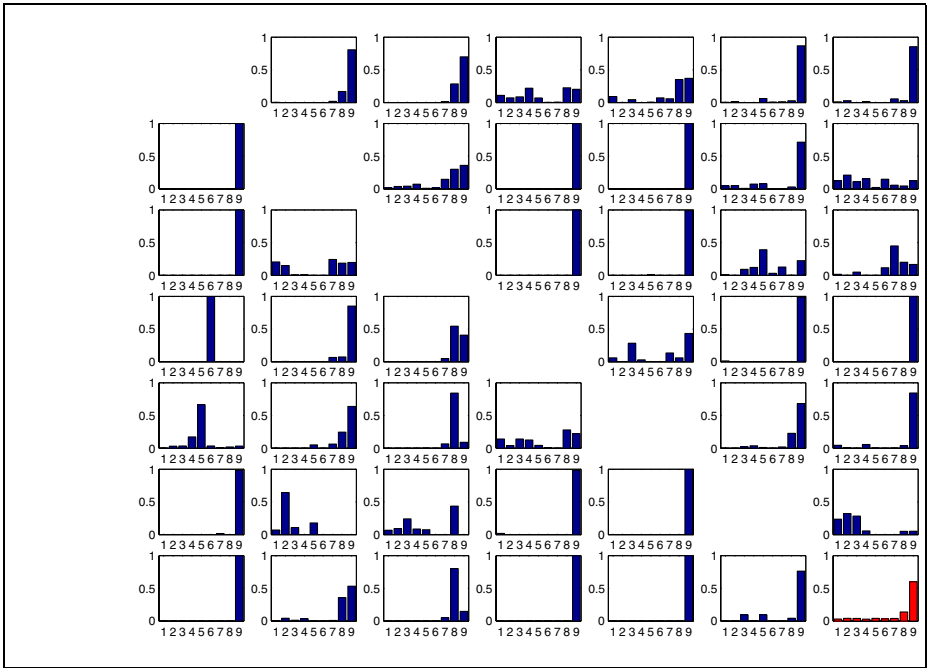
voting	geometry	parameter	mean	(std)
majority voting	none	—	64.25	(3.21)
majority voting	relations A	—	74.78	(2.92)
majority voting	relations B	—	75.08	(2.51)
[5]	—	—	85	n/a
SVM	none	$C = 0.2583$	90.60	(2.06)
SVM	relations A	$C = 0.7622$	90.90	(2.16)
SVM	relations B	$C = 0.1666$	<b>91.28</b>	(2.28)

Due to time restrictions we used a 50-50-split of the data in order to optimize the parameters of the learning algorithms, i.e.  $D$  for LPBoost, and  $C$  for the SVM. In every case, we apply a simple iterative search using nested intervals to obtain reliable estimates. Thus we are able to select the value yielding the lowest test error on the corresponding 50-50 split of the data. Finally, we fix those parameters, and conduct a stratified 10-fold cross-validation on the database [Tab. 2 - 3]. Note each one-vs-one classifier is learned over a reduced training and test set, including only the instances of the class combination. Fixing those hypotheses, we calculate their predictions over the instances from all classes and perform the weighted voting scheme proposed. For the SVM, we use SVMlight [12]<sup>1</sup>, where we also tried nonlinear kernels but omit their use on since those kernels performed poorer than the linear one.

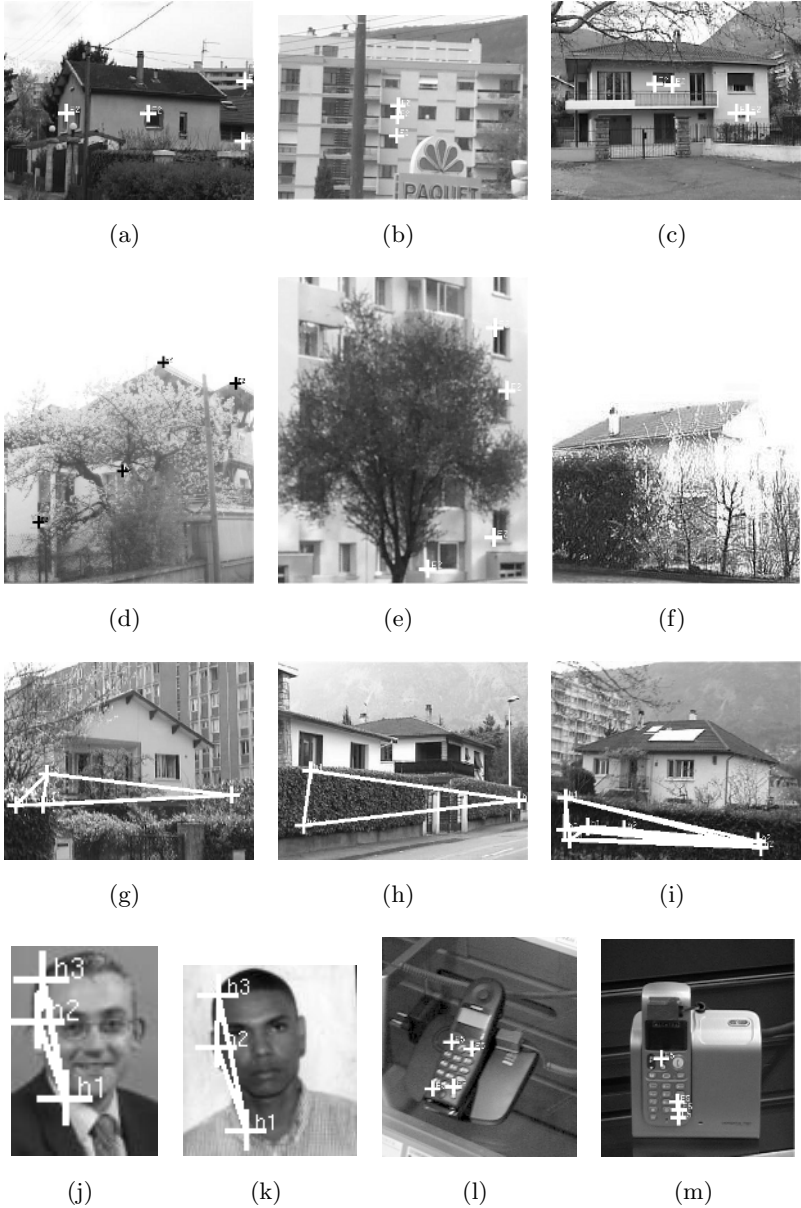
<sup>1</sup> Available at <http://svmlight.joachims.org/>

**Table 3. (top)** Confusion matrix upon 10-fold cross validation using SVM and the more complex geometry ‘relations B’. The true classes are denoted in the top row. **(bottom)** Histogram of the feature types  $\phi$  selected by each one-vs-one classifier upon the 50-50-split for the non-geometric case. Thus, a column denotes different background classes. - lower-right: overall selection upon the 50-50-split for the non-geometric case.

→	faces	buildings	trees	cars	phones	bikes	books
faces	98.9873	0.6667	1.3333	8.4762	2.6455	0	0.7143
bldgs	0	70.6667	8.0000	0	0	2.8431	8.9286
trees	0	10.0000	87.3333	0	0	0.8333	1.4286
cars	0.5063	0	0.6667	84.0952	9.4180	0	0
phones	0.5063	0	0	7.4286	87.9365	0	0
bikes	0	2.6667	2.6667	0	0	94.6569	2.1429
books	0	16.0000	0	0	0	1.6667	86.7857



Furthermore, we analyzed the actual feature selection by the total weight assigned by LPBoost to the weak hypotheses of a certain feature type  $\phi$  (3). Using the optimal parameters for the 50-50-split, it turned out that most one-vs-one classifiers select segments along with SIFTs (PCA40) for the non-geometric (basic) case [Tab. 3]. As shown in Table 3, we observed a strong correlation between the test error of a one-vs-one classifier and the number of different feature types within it’s final hypothesis. If two categories are hard to classify, the



**Fig. 2.** Example images with voting location of selected weak hypotheses are taken from various one-vs-one classifiers. [Fig. 2(a) - 2(f)] are used for learning buildings against trees. Only the most important hypothesis and its matching feature locations are drawn. [Fig. 2(a) - 2(c)] show correctly classified examples, [Fig. 2(d) - 2(f)] show misclassified examples. [Fig. 2(g) - 2(i)] show three correct classified images using a geometric hypothesis learned from buildings vs books. [Fig. 2(j) - 2(m)] show examples for the geometric hypothesis for the class of faces and a simple hypothesis for phones.



learning algorithm will use more different feature types. This demonstrates the intrinsic flexibility of our method when dealing with difficult class combinations.

Figure 2 shows exemplary images along with weak hypotheses, used by the corresponding one-vs-one classifiers. All of them were taken from a single fold during cross-validation. In case of buildings vs. trees, our method selects only one simple SIFT (PCA40) feature for classification. The weak hypothesis is triggered particularly at window corners [Fig. 2(a) - 2(c)]. Figures 2(d) - 2(e) show false detections of that classifier. Both images belong to the class of trees since those are in the foreground. Although the buildings are in the background, our classifier detects the window corners, visible through and around the tree, and is still able to predict the building. On the other way around, Figure 2(f) gets misclassified as tree because there are no such corners visible. These examples show the difficulties in building an unambiguous database, and confirm the quality of our classifiers.

In that line of argument, one would expect that such a simple feature would be insufficient to distinguish buildings from books, since window corners are similar to those of books. Indeed, the weak hypothesis of highest weight is a geometric relation between two features. One feature represents a window corner and the other triggers on green fields. The second best weak hypothesis uses three features, and votes in the case where there is a hedgerow in front of a building [Fig. 2(g) - 2(i)]. This is reasonable considering that the class book only contains books on bookshelves or desktops, but no plants. Figures 2(j) and 2(k) belong to the class faces. The geometric hypothesis selected votes on triangle configurations of an ear, the hair line and the collar. Figures 2(l) and 2(m) show a weak hypothesis for the class of phones.

## 5 Conclusions and Outlook

In this paper we use a new method for learning geometric relations between features for image categorization through boosting. Our algorithm selects the important feature types, estimates the need of geometric models and learns such models if necessary. A final hypothesis can consist of several geometric hypotheses, that solves the multi-modal appearance problem of objects. We do not have to flip images, such that the target object always faces the same direction. We address the multiclass classification problem with a method for combining one-vs-one classifiers.

We found that learning without geometry already gives good performance, and that slight improvements are achieved by moving from simple to more complex geometric relations. An evaluation of the geometric hypotheses reveals that it is hard to find a relation with more than three features. Simple hypotheses using a single feature and pairwise relations dominate the final solution, which might be due to the rather small cardinality of some classes.

In the future, the framework may be extended with a detector stage. Also other types of geometric primitives within the weak learner are possible and should be tried out.

**Acknowledgments.** This work was supported by the European project LAVA (IST-2001-34405) and by the FSP/JRP Cognitive Vision of the Austrian Science Funds (FWF-JRP S9104-N04 SP4). This work was also supported in part by the IST program of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

## References

1. Kristin P. Bennett, Ayhan Demiriz, and John Shawe-Taylor. A column generation algorithm for boosting. In *Proc. 17th International Conf. on Machine Learning*, pages 65–72. Morgan Kaufmann, San Francisco, CA, 2000.
2. M.C. Burl, T.K. Leung, and P. Perona. Recognition of planar object classes. In *Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, pages 223–230. IEEE Computer Society, 1996.
3. Michael C. Burl, Markus Weber, and Pietro Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision-Volume II*, volume 1407, pages 628–641. Springer-Verlag, 1998.
4. David Crandall, Pedro F. Felzenszwalb, and Daniel P. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *CVPR (1)*, pages 10–17. IEEE Computer Society, 2005.
5. Gabriela Csurka, Cedric Bray, Christopher Dance, and Lixin Fan. Visual categorization with bags of keypoints. In *European Conference on Computer Vision, ECCV'04*, Prague, Czech Republic, May 2004.
6. Ayhan Demiriz, Kristin P. Bennett, and John Shawe-Taylor. Linear programming boosting via column generation. *Machine Learning*, 46(1-3):225–254, 2002.
7. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003.
8. Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995.
9. Michael Fussenegger, Andreas Opelt, Axel Pinz, and Peter Auer. Object recognition using segmentation for feature detection. In *ICPR (3)*, pages 41–44, 2004.
10. R. Gonzalez and R. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, Inc., 1992.
11. Luc J. Van Gool, Theo Moons, and Dorin Ungureanu. Affine/photometric invariants for planar intensity patterns. In *ECCV '96: Proceedings of the 4th European Conference on Computer Vision-Volume I*, pages 642–651. Springer-Verlag, 1996.
12. Thorsten Joachims. Making large-scale support vector machine learning practical. In *Advances in kernel methods: support vector learning*, pages 169–184. MIT Press, Cambridge, MA, USA, 1999.
13. Timor Kadir and Michael Brady. Saliency, scale and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.
14. D.G. Lowe. Object recognition from local scale-invariant features. In *Seventh International Conference on Computer Vision*, pages 1150–1157, 1999.
15. K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *International Conference on Computer Vision*, pages 525–531, 2001.

16. Andreas Opelt, Michael Fussenegger, Axel Pinz, and Peter Auer. Weak hypotheses and boosting for generic object detection and recognition. In *Proc. of the 8th European Conference on Computer Vision (ECCV)*, volume 2, pages 71–84, 2004.
17. Andreas Opelt, Michael Fussenegger, Axel Pinz, and Peter Auer. Generic object recognition with boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):416–431, March 2006.
18. G. Rätsch, B. Schölkopf, A. Smola, K.-R. Müller, T. Onoda, and S. Mika.  $\nu$ -Arc: Ensemble learning in the presence of outliers. In D.A. Cohn M.S. Kearns, S.A. Solla, editor, *Advances in Neural Information Processing Systems 12*. MIT Press, 2000.
19. Vladimir Vapnik. *The nature of statistical learning theory*. Springer, New York, 1995.